

**UNIVERSIDAD NACIONAL DE CHIMBORAZO**



**FACULTAD DE INGENIERIA  
CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES**

Proyecto de Investigación previo a la obtención del título de Ingeniero en Electrónica y  
Telecomunicaciones

**TRABAJO DE TITULACIÓN**

“DESARROLLO DE UN SISTEMA TELEOPERADO PARA LA MANIPULACIÓN DE UN  
PROTOTIPO DE BRAZO ROBÓTICO BASADO EN EL APRENDIZAJE POR  
DEMOSTRACIÓN, PARA LA AUTOMATIZACIÓN DE PROCESOS DE APILAMIENTO  
EN LA INDUSTRIA DE HORMIGÓN”

**Autores:**

Alexis Fernando Ilbay Vicuña

Dania Katherine Llanos Satama

**Tutor:**

PhD. Leonardo Rentería

**Riobamba - Ecuador**

**Año 2020**

Los miembros del tribunal de graduación del proyecto de investigación de título: **DESARROLLO DE UN SISTEMA TELEOPERADO PARA LA MANIPULACIÓN DE UN PROTOTIPO DE BRAZO ROBÓTICO BASADO EN EL APRENDIZAJE POR DEMOSTRACIÓN, PARA LA AUTOMATIZACIÓN DE PROCESOS DE APILAMIENTO EN LA INDUSTRIA DE HORMIGÓN**, presentado por: Alexis Fernando Ilbay Vicuña y Dania Katherine Llanos Satama, dirigida por PhD. Leonardo Fabian Rentería Bustamante.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual consta el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la Biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firma

Mgs. Carlos Peñafiel  
**Presidente del Tribunal**



Firma

Mgs. Giovanni Cuzco  
**Miembro del Tribunal**

**EDGAR  
GIOVANNY  
CUZCO  
SILVA**



Firmado digitalmente por EDGAR GIOVANNY CUZCO SILVA  
Nombre de reconocimiento (DN): c=EC, o=BANCO CENTRAL DEL ECUADOR, ou=ENTIDAD DE CERTIFICACION DE INFORMACION-ECIBCE, l=QUITO, serialNumber=0000431517, cn=EDGAR GIOVANNY CUZCO SILVA  
Fecha: 2020.06.03 20:42:11 -05'00'

Firma

Mgs. Daniel Haro  
**Miembro del Tribunal**



Firma

## DECLARACIÓN EXPRESA DE TUTORÍA

En calidad de tutor del tema de investigación: **DESARROLLO DE UN SISTEMA TELEOPERADO PARA LA MANIPULACIÓN DE UN PROTOTIPO DE BRAZO ROBÓTICO BASADO EN EL APRENDIZAJE POR DEMOSTRACIÓN, PARA LA AUTOMATIZACIÓN DE PROCESOS DE APILAMIENTO EN LA INDUSTRIA DE HORMIGÓN**, Realizado por el Sr. Alexis Fernando Ilbay Vicuña y la Srta. Dania Katherine Llanos Satama, para optar por el título de Ingeniero en Electrónica y Telecomunicaciones, considero que reúnen los requisitos y méritos suficientes para ser sustentada públicamente y evaluada por el jurado examinador que se designe.

Riobamba, 21 de mayo de 2020

PhD. Leonardo Rentería

C.I. 1104064132

**TUTOR**



**Firma**

## AUTORÍA DE LA INVESTIGACIÓN

La responsabilidad del contenido de este proyecto de graduación corresponde exclusivamente a **Alexis Fernando Ilbay Vicuña, Dania Katherine Llanos Satama, PhD. Leonardo Rentería**; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.



---

Alexis Fernando Ilbay Vicuña

C.I. 060394760-7



---

Dania Katherine Llanos Satama

C.I. 060405230-8

## **DEDICATORIA.**

*“Un ave parada en un árbol nunca tiene miedo de si la rama se quiebra, porque su confianza no está en la rama, sino en sus propias alas. SIEMPRE CREE EN TI MISMO”.*

### ***Confucio***

El presente trabajo va dedicado a DIOS como eje principal en toda mi vida, por su amor y esfuerzo a mi madre, padre y hermano que día con día con sus risas y ocurrencias, me han sabido alentar para seguir luchando por todos mis objetivos, metas y sueños, así como también a todas las personas que han estado conmigo apoyándome en todo momento, dándome ánimos para seguir en adelante y no caer, pero sobre todo por haberme dedicado algo que es muy valioso y no regresa, su tiempo.

*Katherine Llanos S.*

Dedico este trabajo a Dios quien ha sido el pilar fundamental en el día a día de mi vida, por ser quien me dio padres y hermanos quienes inspiraron e inspirarán motivación, esfuerzo y aliento en mí.

*Alexis Ilbay V.*

## AGRADECIMIENTOS.

Todo el conocimiento que he adquirido, la estricta educación que tengo y he tenido a lo largo de mi vida, han venido por parte de mi padre **VICTOR LEONEL LLANOS VEGA**, con lo cual me ha hecho madurar como persona y con su ejemplo me ha enseñado a perseverar cada día con esfuerzo y sacrificio para alcanzar las metas planteadas y el mejor relato de un amor verdadero e incondicional, de una lucha constante, de una paciencia inagotable y una sonrisa inquebrantable en los momentos más difíciles han sido impartidos por mi madre **CONSUELO DEL CARMEN SATAMA LLANES**, juntos me han enseñado a valorar su esfuerzo y su entrega diaria, acompañándome cada día en este, mi proyecto de vida, por aconsejarme en bien, por llenarme de humildad, por secar mis lágrimas y por siempre depositar toda su confianza en mí, por todo esto y mucho más les estoy inmensamente agradecida, hare lo mejor posible por recompensar todo lo bonito que he recibido.

A mi hermano *Johel Llanos* por todos los momentos que hemos vivido, por siempre estar conmigo sacándome una sonrisa, por depositar su confianza en mí, por ser mi ejemplo a seguir ya que me llena de mucho orgullo el ver cómo ha alcanzado sus sueños con mucho esfuerzo y perseverancia.

A mi novio *Edgar Martínez Fuentes* por llenar mi vida de mucho amor, por tener las palabras adecuadas para motivarme e impulsarme a perseguir con firmeza mis sueños y a su vez construir los nuestros, por ser mi apoyo fundamental en todo momento y aún más en los momentos difíciles, por siempre confiar en mí y ayudarme a superar mis inseguridades, gracias por darme esa paz y ese amor que tanto necesitaba *E711K*.

A mis hermanas *Mayra* y *Carla Satama* por tenerme paciencia, por saberme aconsejar y por haberme dado su valioso tiempo para compartir momentos de risas, de llantos, de experiencias únicas, por la confianza y por muchas más locuras juntas *C.M.K*, todo lo que hemos vivido está guardado en mi corazón.

A mis docentes a mi tutor *Leonardo Rentería* y *Giovanny Cuzco* les agradezco por todo el apoyo brindado a lo largo de la carrera, por su tiempo, amistad y por todos los conocimientos que me transmitieron.

*Katherine Llanos S.*

## **AGRADECIMIENTOS.**

Agradezco a mi madre por enseñarme con su ejemplo el significado de las palabras amor, coraje, disciplina, esfuerzo, humildad, lealtad, honor y gratitud.

Por enseñarme amor al trabajar más de 1 millón de horas, para que así pueda tener que comer, que vestir y tratar de que nada me falte, por quitarse el pan de la boca y ponerlo en la mía y la de mis hermanos, por prepararme más de 27 393 alimentos con alegría, que más que llenar mi cuerpo, llenaban mi corazón, enseñándome que el principal ingrediente en la receta de la vida es el coraje y la valentía, mostrándome el camino para no rendirme ni dar treguas, con abrazos eliminar la tristeza, con sonrisas el dolor y con un “yo sé que puedes” levantarme del suelo y darme fortaleza más allá de lo humano, además de darme el verdadero significado de conquistar y colonizar, Inculcándome moral y valores, instruyéndome a ser rebelde con un propósito, a no vender mi paz por puñados de monedas, enseñándome que lo más importante es Dios.

*Alexis Ilbay V.*

## ÍNDICE GENERAL

DEDICATORIA.....	V
AGRADECIMIENTOS.....	VII
ÍNDICE GENERAL.....	VIII
ÍNDICE DE ILUSTRACIONES.....	XII
ÍNDICE DE TABLAS.....	XVI
RESUMEN.....	XVII
ABSTRACT.....	XVIII
CAPITULO I.....	1
1.1 INTRODUCCIÓN.....	1
1.2 PLANTEAMIENTO DEL PROBLEMA.....	1
1.3 OBJETIVOS.....	4
1.3.1 OBJETIVO GENERAL.....	4
1.3.2 OBJETIVOS ESPECÍFICOS.....	4
CAPITULO II.....	5
2.1 ESTADO DEL ARTE.....	5
2.2 ROBÓTICA Y TELEOPERACIÓN.....	6
2.3 COMPONENTES DEL SISTEMA TELEOPERADO DE UN ROBOT.....	6
2.3.1 ZONA LOCAL O ESTACIÓN DE TELEOPERACIÓN.....	7
2.3.2 ZONA REMOTA O ESCLAVO (BRAZO ROBÓTICO).....	8
2.3.3 CANAL DE COMUNICACIÓN.....	8
2.3.3.1 PROTOCOLOS DE COMUNICACIÓN.....	9
2.4 METODOLOGÍA DE ENSEÑANZA PARA ROBOTS.....	9
2.4.1 MÉTODOS DE PROGRAMACIÓN DE UN ROBOT.....	9
2.4.1.1 ROBOTS DE SECUENCIA LIMITADA (BAJA TECNOLOGÍA).....	9
2.4.1.2 ROBOTS DE REPRODUCCIÓN (MEDIADA TECNOLOGÍA).....	10
2.4.1.3 ROBOTS INTELIGENTES (ALTA TECNOLOGÍA).....	10
2.4.2 ENTORNO DE PROGRAMACIÓN.....	10
2.4.2.1 PROGRAMACIÓN TEXTUAL.....	10
2.4.2.2 PROGRAMACIÓN GUIADA.....	11
2.4.2.2.1 UTILIZACIÓN DE DISPOSITIVOS DE ENSEÑANZA.....	11
2.4.2.2.2 APRENDIZAJE DIRECTO.....	11
CAPITULO III.....	12

<b>3.1 TIPO DE INVESTIGACIÓN</b> .....	12
<b>3.1.1 INVESTIGACIÓN EXPERIMENTAL</b> .....	12
<b>3.1.2 INVESTIGACIÓN CUALITATIVO</b> .....	12
<b>3.2 UNIDAD DE ANÁLISIS</b> .....	12
<b>3.2.1 POBLACIÓN DE ESTUDIO</b> .....	12
<b>3.3 TÉCNICAS DE RECOLECCIÓN</b> .....	12
<b>3.3.1 OBSERVACIÓN</b> .....	12
<b>3.3.2 FUENTE DE RECOPIACIÓN DE INFORMACIÓN</b> .....	12
<b>3.4 INSTRUMENTOS DE LA INVESTIGACIÓN</b> .....	13
<b>3.5 HIPÓTESIS</b> .....	13
<b>3.6 METODOLOGÍA</b> .....	13
<b>3.8 PROCEDIMIENTOS Y ANÁLISIS</b> .....	14
<b>3.8.1 BRÓKER MQTT</b> .....	16
<b>3.8.1.1 LECTURA Y ESCRITURA EN EL SERVIDOR MQTT</b> .....	16
<b>3.8.1.1.1 CREACIÓN DEL TÓPICO Y SUBTÓPICOS SOBRE EL SERVIDOR MQTT</b> .....	17
<b>3.8.1.1.2 ESCRITURA EN LOS SUBTÓPICOS DEL SERVIDOR MQTT (PUBLICACIÓN)</b> .....	18
<b>3.8.1.1.3 LECTURA EN LOS SUBTÓPICOS DEL SERVIDOR MQTT (SUBSCRIPCIÓN)</b> .....	19
<b>3.8.2 ESTACION LOCAL</b> .....	21
<b>3.8.2.1 ALGORÌTMO DE ADQUISICIÓN E INTERPRETACIÓN DE DATOS</b> .....	21
<b>3.8.2.1.1 ACCESO AL SERVIDOR MQTT</b> .....	22
<b>3.8.2.1.2 ADQUISICIÓN DE DATOS</b> .....	23
<b>3.8.2.1.3 ACONDICIONAMIENTO DE DATOS</b> .....	23
<b>3.8.2.2 UNIDAD ROBÓTICA (BRAZO ROBÓTICO DE CONTROL)</b> .....	25
<b>3.8.2.2.1 DISEÑO DE LOS COMPONENTES DE LA UNIDAD ROBÓTICA</b> .....	25
<b>3.8.2.2.2 MONTAJE DE LA UNIDAD ROBÓTICA</b> .....	26
<b>3.8.3 ESTACIÓN REMOTA</b> .....	28
<b>3.8.3.1 ALGORÌTMO DE BAJADA Y PROCESAMIENTO DE DATOS</b> .....	28
<b>3.8.3.1.1 SUBSCRIPCIÓN O BAJADA DE DATOS DE BRÓKER MQTT</b> .....	28
<b>3.8.3.1.2 PROCESAMIENTO DE DATOS DE BRÓKER MQTT</b> .....	29
<b>3.8.3.2 ALGORÌTMO DE APRENDIZAJE POR DEMOSTRACIÓN</b> .....	30
<b>3.8.3.2.1 RECOLECCIÓN DE DATOS DEL APRENDIZAJE</b> .....	31

3.8.3.2.2 REPRODUCCIÓN DE DATOS DEL APRENDIZAJE .....	32
3.8.3.3 UNIDAD ROBÓTICA (BRAZO ROBÓTICO MASTER) .....	34
3.8.3.3.1. UNIDAD DE ALIMENTACIÓN .....	34
3.8.3.3.2. UNIDAD DE SENSORES .....	35
3.8.3.3.3. UNIDAD DE CONTROL DE MOVIMIENTO .....	36
3.8.3.3.4. BRAZO ROBÓTICO .....	36
3.8.3.3.4.1. DISEÑO DE LOS COMPONENTES DEL BRAZO ROBÓTICO .....	36
3.8.3.3.4.2. DESCRIPCIÓN CINEMÁTICA DEL BRAZO ROBÓTICO .....	38
3.8.3.3.4.2. MONTAJE DE LA UNIDAD ROBÓTICA .....	41
<b>CAPITULO IV .....</b>	<b>42</b>
<b>4.1 RESULTADOS EN EL SISTEMA DE TELEOPERACIÓN .....</b>	<b>42</b>
4.1.1 PRUEBA DEL SISTEMA DE TELEOPERACIÓN AL VACÍO .....	42
4.1.2 PRUEBA DE EVALUACIÓN DE MOVIMIENTO DEL BRAZO ROBÓTICO .....	44
4.1.3 PRUEBAS DEL SISTEMA DE SUPERVISIÓN Y CONTROL .....	47
<b>4.2 RESULTADOS OBTENIDOS EN EL APRENDIZAJE DEL BRAZO ROBÓTICO .....</b>	<b>49</b>
4.2.1 PRUEBAS DE LA ALIMENTACIÓN DEL SISTEMA .....	49
4.2.2 PRUEBAS DE LA RECOLECCIÓN DE DATOS .....	50
4.2.3 PRUEBAS DE LECTURA O REPRODUCCIÓN DE ENSEÑANZA .....	67
4.2.4 PRUEBAS PARA LA DETECCIÓN DE OBSTÁCULOS .....	77
4.2.5 PRUEBA DE TODA LA ETAPA DE APRENDIZAJE .....	79
<b>4.3 RESULTADOS DE REALIZACIÓN DE TAREAS SOBRE AMBIENTE DE TRABAJO EMULADO .....</b>	<b>81</b>
4.3.1 PRUEBAS DE TRAYECTORIA CON CARGA Y SIN CARGA .....	81
4.3.2 REPETIVIDAD DE TAREAS .....	87
<b>CAPITULO V .....</b>	<b>90</b>
<b>5.1 CONCLUSIONES.....</b>	<b>90</b>
<b>5.2 RECOMENDACIONES.....</b>	<b>91</b>
<b>BIBLIOGRAFÍA .....</b>	<b>93</b>
<b>ANEXOS .....</b>	<b>100</b>
ANEXO 1 – MANUAL DE MANEJO PARA EL BRAZO ROBÓTICO .....	101
ANEXO 2 – INSTRUCCIONES DEL BRAZO ROBÓTICO .....	102
ANEXO 3 – INSTALACIÓN EN LA ESTACIÓN REMOTA .....	104
ANEXO 4 – INSTALACIÓN EN LA ESTACIÓN LOCAL .....	105

ANEXO 5 – INSTALACIÓN DE RASPBIAN ..... 107  
ANEXO 6 – INSTALACIÓN DEL PROTOCOLO I2C ..... 109  
ANEXO 7 – HERRAMIENTAS Y COMPONENTES..... 110

## ÍNDICE DE ILUSTRACIONES

<i>Ilustración 1. Elementos de un sistema Teleoperado.....</i>	<i>7</i>
<i>Ilustración 2. Esquema General de un Sistema de Teleoperación .....</i>	<i>7</i>
<i>Ilustración 3. Diagrama de Bloques de un Sistema de Teleoperación Típico.....</i>	<i>8</i>
<i>Ilustración 4. Prototipo experimental del prototipo de brazo robótico Teleoperado con aprendizaje por demostración (Esquema General). .....</i>	<i>14</i>
<i>Ilustración 5. Diagrama general del sistema de Teleoperación y Aprendizaje por Demostración. .....</i>	<i>15</i>
<i>Ilustración 6. Estructura del servidor MQTT .....</i>	<i>16</i>
<i>Ilustración 7. Diagrama general para la creación de tópicos y subtópicos.....</i>	<i>17</i>
<i>Ilustración 8. Diagrama de fases del proceso de tareas del servidor MQTT (Publicación, Suscripción, Ping).....</i>	<i>18</i>
<i>Ilustración 9. Diagrama del proceso de Publicación en el bróker MQTT. ....</i>	<i>19</i>
<i>Ilustración 10. Diagrama del proceso de Suscripción en el bróker MQTT.....</i>	<i>20</i>
<i>Ilustración 11. Algoritmo de adquisición y acoplamiento de datos. ....</i>	<i>21</i>
<i>Ilustración 12. Parámetros generales para la adquisición, acondicionamiento y envío de datos. .....</i>	<i>22</i>
<i>Ilustración 13. Diagrama de acceso al servidor.....</i>	<i>22</i>
<i>Ilustración 14. Diagrama del algoritmo de adquisición, acondicionamiento y envío de datos. .</i>	<i>25</i>
<i>Ilustración 15. Componentes del Brazo Robótico de Control. ....</i>	<i>26</i>
<i>Ilustración 16. Unidad Robótica o Brazo Robótico de Control.....</i>	<i>26</i>
<i>Ilustración 17. Diagrama esquemático de la Estación Local.....</i>	<i>27</i>
<i>Ilustración 18. Diagrama general del algoritmo de baja y procesamiento de datos. ....</i>	<i>28</i>
<i>Ilustración 19. Diagrama general del procesamiento de los datos obtenidos del bróker MQTT. .....</i>	<i>29</i>
<i>Ilustración 20. Diagrama general de funcionamiento del algoritmo de suscripción y procesamiento de datos.....</i>	<i>30</i>
<i>Ilustración 21. Diagrama general del algoritmo de Aprendizaje Por Demostración. ....</i>	<i>30</i>
<i>Ilustración 22. Diagrama de la recolección y almacenamiento de datos.....</i>	<i>31</i>
<i>Ilustración 23. Diagrama de conexiones de la etapa de recolección de datos de aprendizaje. ..</i>	<i>32</i>
<i>Ilustración 24. Diagrama de general de la reproducción de datos de aprendizaje. ....</i>	<i>33</i>
<i>Ilustración 25. Diagrama de conexiones de la etapa de reproducción de movimientos del aprendizaje.....</i>	<i>33</i>
<i>Ilustración 26. Unidad robótica.....</i>	<i>34</i>
<i>Ilustración 27. Diagrama de conexiones de la unidad de alimentación.....</i>	<i>35</i>
<i>Ilustración 28. Diagrama general de la unidad de sensores (Detección de obstáculos). ....</i>	<i>35</i>
<i>Ilustración 29. Diagrama de conexión de la unidad de control de movimiento y actuadores(servomotores).....</i>	<i>36</i>
<i>Ilustración 30. Engrane realizado en SOLIDWORK.....</i>	<i>37</i>
<i>Ilustración 31. Piñón realizado en SOLIDWORK .....</i>	<i>37</i>
<i>Ilustración 32. Componentes del Brazo Robótico de Master.....</i>	<i>37</i>
<i>Ilustración 33. Diagrama de relación entre cinemática inversa y directa .....</i>	<i>38</i>
<i>Ilustración 34. Diagrama esquemático de los primeros 3 GDL.....</i>	<i>40</i>
<i>Ilustración 35. Brazo Robótico Master.....</i>	<i>41</i>

<b>Ilustración 36.</b> Escenario para pruebas de Teleoperación.....	43
<b>Ilustración 37.</b> Métodos de verificación de resultados .....	44
<b>Ilustración 38.</b> Trayectoria realizada por el usuario con el dispositivo de control en la estación loca.....	45
<b>Ilustración 39.</b> Trayectoria realizada por el brazo robótico en la estación remota.....	46
<b>Ilustración 40.</b> Comparación de trayectoria deseada y trayectoria realizada. ....	47
<b>Ilustración 41.</b> Pruebas de Supervisión y control (a) usuario control 1º GDL (b) usuario control 2ª GDL (c) usuario control brazo completo .....	48
<b>Ilustración 42.</b> Conexiones del módulo relé para alimentación de servomotores y encoders.....	49
<b>Ilustración 43.</b> Trayectoria de los datos recolectados en la primera articulación BASE. ....	52
<b>Ilustración 44.</b> Trayectoria expresada en grados de los datos recolectados en la primera articulación BASE.....	52
<b>Ilustración 45.</b> Trayectoria de los datos recolectados en la segunda articulación HOMBRO. ..	53
<b>Ilustración 46.</b> Trayectoria expresada en grados de los datos recolectados en la segunda articulación HOMBRO .....	54
<b>Ilustración 47.</b> Trayectoria de los datos recolectados en la tercera articulación CODO.....	55
<b>Ilustración 48.</b> Trayectoria expresada en grados de los datos recolectados en la tercera articulación CODO.....	55
<b>Ilustración 49.</b> Trayectoria completa de los datos recolectados en cada grado de libertad.....	56
<b>Ilustración 50.</b> Trayectoria completa expresada en grados de los datos recolectados en cada grado de libertad.....	57
<b>Ilustración 51.</b> Conexiones de encoder a pines de interrupción.....	58
<b>Ilustración 52.</b> Trayectoria de la recolección de los datos de la primera articulación Base, con una velocidad 1 .....	59
<b>Ilustración 53.</b> Trayectoria de la recolección de los datos de la primera articulación Base, con una velocidad 2 .....	60
<b>Ilustración 54.</b> Trayectoria de la recolección de los datos de la segunda articulación Hombro, con una velocidad 1 .....	61
<b>Ilustración 55.</b> Trayectoria de la recolección de los datos de la segunda articulación Hombro, con una velocidad 2 .....	62
<b>Ilustración 56.</b> Trayectoria de la recolección de los datos de la tercera articulación Codo, con una velocidad 1 .....	63
<b>Ilustración 57.</b> Trayectoria de la recolección de los datos de la tercera articulación Codo, con una velocidad 2 .....	64
<b>Ilustración 58.</b> Trayectoria completa de los datos recolectados en cada grado de libertad con una velocidad 1 .....	65
<b>Ilustración 59.</b> Trayectoria completa de los datos recolectados en cada grado de libertad con una velocidad 2 .....	66
<b>Ilustración 60.</b> Diagrama de conexión para la reproducción .....	68
<b>Ilustración 61.</b> Trayectoria realizada de la reproducción de los datos de la primera articulación Base.....	69
<b>Ilustración 62.</b> Trayectoria realizada de la reproducción de los datos de la primera articulación Base con respecto a la trayectoria deseada. ....	70
<b>Ilustración 63.</b> Trayectoria realizada de la reproducción de los datos de la segunda articulación Hombro .....	71

<b>Ilustración 64.</b> Trayectoria realizada de la reproducción de los datos de la segunda articulación Hombro con respecto a la trayectoria deseada. ....	72
<b>Ilustración 65.</b> Trayectoria realizada de la reproducción de los datos de la tercera articulación Codo.....	73
<b>Ilustración 66.</b> Trayectoria realizada de la reproducción de los datos de la segunda articulación Hombro con respecto a la trayectoria deseada. ....	74
<b>Ilustración 67.</b> Trayectoria realizada de la reproducción de los datos de cada grado de libertad del Brazo robótico.....	75
<b>Ilustración 68.</b> Trayectoria realizada de la reproducción de los datos de cada grado de libertad del Brazo robótico con respecto a la trayectoria deseada. ....	76
<b>Ilustración 69.</b> Conexiones del sensor ultrasónico .....	77
<b>Ilustración 70.</b> Conexiones del sensor PIR.....	78
<b>Ilustración 71.</b> Conexiones de todas las etapas que conforman el aprendizaje directo.....	79
<b>Ilustración 72.</b> Conexión Final de todas las etapas que conforman el aprendizaje directo.....	80
<b>Ilustración 73.</b> Trayectoria con carga y sin carga con un peso de 0,5 kg.....	82
<b>Ilustración 74.</b> Trayectoria con carga y sin carga representado en cada eje de plano con un peso de 0,5kg.....	83
<b>Ilustración 75.</b> Trayectoria con carga y sin carga con un peso de 0,7kg.....	84
<b>Ilustración 76.</b> Trayectoria con carga y si carga con un peso de 0,7kg representada en cada eje del plano.....	84
<b>Ilustración 77.</b> Diagrama en bloques del Control de error.....	85
<b>Ilustración 78.</b> Señal con Desfase corregida por el control de error.....	85
<b>Ilustración 79.</b> Trayectoria con carga sin desfase.....	86
<b>Ilustración 80.</b> Trayectoria con carga sin desfase presentada en cada eje del plano.....	86
<b>Ilustración 81.</b> Función “Update”.....	104
<b>Ilustración 82.</b> Instalación del bróker Mosquitto.....	104
<b>Ilustración 83.</b> Instalación del cliente Mosquitto.....	105
<b>Ilustración 84.</b> Instalación de la librería “PubSubClient”.....	106
<b>Ilustración 85.</b> Software de supervisión del Bróker MQTT.....	106
<b>Ilustración 86.</b> Software de supervisión de video.....	107
<b>Ilustración 87.</b> Iniciación del sistema operativo sobre Raspberry Pi.....	108
<b>Ilustración 88.</b> Proceso de instalación del Raspbian.....	108
<b>Ilustración 89.</b> Finalización de la instalación de Raspbian.....	109
<b>Ilustración 90.</b> Comprobación del protocolo I2C sobre Raspberry.....	110
<b>Ilustración 91.</b> IDE Arduino.....	111
<b>Ilustración 92.</b> Arduino UNO.....	112
<b>Ilustración 93.</b> Arduino MEGA 2650.....	112
<b>Ilustración 94.</b> Arduino NANO.....	112
<b>Ilustración 95.</b> RASPBERRY PI 3.....	113
<b>Ilustración 96.</b> Sistemas Operativos de terceros.....	114
<b>Ilustración 97.</b> IDE Thonny de Python.....	114
<b>Ilustración 98.</b> Potenciómetro.....	115
<b>Ilustración 99.</b> Arduino Shield Ethernet.....	115
<b>Ilustración 100.</b> Servomotores a) MG996R y b) TD-8320MG.....	116
<b>Ilustración 101.</b> Módulo KY -040.....	117

<i><b>Ilustración 102. Módulo PCA 9685 .....</b></i>	<b>117</b>
<i><b>Ilustración 103. Sensor de movimiento PIR.....</b></i>	<b>118</b>
<i><b>Ilustración 104. Módulo Relé.....</b></i>	<b>118</b>
<i><b>Ilustración 105. Sensor Infrarrojo de detección de obstáculos .....</b></i>	<b>119</b>

## ÍNDICE DE TABLAS

<i>Tabla 1. Tópicos y subtópicos del servidor MQTT.....</i>	<i>17</i>
<i>Tabla 2. Asignación de sensores y pines para la adquisición de datos .....</i>	<i>23</i>
<i>Tabla 3. Equivalencias máximas y mínimas de valores de lectura de movimiento expresada en grados. ....</i>	<i>24</i>
<i>Tabla 4. Resolución de GDL del brazo robótico.....</i>	<i>38</i>
<i>Tabla 5. Parámetros DH para el brazo robótico Master.....</i>	<i>39</i>
<i>Tabla 6. Comprobación del éxito de la comunicación con datos publicados y suscritos .....</i>	<i>43</i>
<i>Tabla 7. Posicionamiento del encoder con el servomotor .....</i>	<i>51</i>
<i>Tabla 8. Interrupciones que posee ARDUINO.....</i>	<i>58</i>
<i>Tabla 9. Posiciones recolectadas y posiciones reproducidas. ....</i>	<i>67</i>
<i>Tabla 10. Tabla comparativa de los sensores de detección .....</i>	<i>79</i>
<i>Tabla 11. Cargas de prueba con la pinza.....</i>	<i>87</i>
<i>Tabla 12. Pruebas realizadas con el brazo robótico.....</i>	<i>88</i>
<i>Tabla 13. Especificaciones técnicas del servomotor 996R .....</i>	<i>116</i>
<i>Tabla 14. Especificaciones técnicas del servomotor TD-8320 .....</i>	<i>116</i>

## **RESUMEN**

Un sistema de teleoperación y aprendizaje por demostración, tiene como propósito ayudar en el sector industrial en zonas de difícil acceso o en zonas de producción que representen un riesgo latente para la integridad y salud de los trabajadores además dándole un sistema que permite la simplicidad de la programación, evitando costos innecesarios en capacitaciones complejas para manipulación de robots, por estos motivos la presente investigación busca desarrollar un sistema teleoperado para la manipulación de un prototipo de brazo robótico basado en el aprendizaje por demostración, para la automatización de procesos de apilamiento en la industria de hormigón.

Para lograr el cometido, se implementó el uso de los elementos de un sistema teleoperado es decir una estación local, el canal de comunicación y una estación remota. El canal de comunicación se basa en la utilización de una red lógica única de alcance mundial (Internet) y un servidor MQTT para establecer una comunicación entre la estación local y la estación remota. Mientras que la estación remota presenta la integración del sistema de aprendizaje por demostración que basa su funcionamiento en el uso de componentes, como principal se tiene el encoder rotativo KY-040 que en conjunto con el servomotor dan una relación de posicionamiento de cada grado de libertad, además se utilizaron sistemas embebidos que son los encargados de recolectar la información necesaria en la enseñanza del movimiento para posteriormente poder replicar dicha acción.

Las pruebas obtenidas en el aprendizaje directo fueron óptimas, se realizó pruebas de repetitividad con carga y sin carga, para visualizar el funcionamiento correcto del brazo robótico, las mismas que corroboraban su buen funcionamiento, los datos obtenidos de esto, pueden ser visualizados desde un monitor serial, y dentro de lo que son, las pruebas en la teleoperación, se puede afirmar que estas cumplen con efectividad todos los movimientos realizados por el usuario en la estación local, viendo así una mayor fluidez en los movimientos del brazo master que se encuentra en la estación remota tiendo un buen tiempo de respuesta, esto pudo ser comprobado por pruebas con software MQTTLens y comandos ping, además de pruebas de trayectoria que se realizaron a través de simulación que ayudaron a determinar el margen de error del brazo robótico obteniendo resultados favorables para este proyecto.

Palabras Clave: MQTT, Teleoperación, Aprendizaje directo, Brazo robótico, Thonny Python

## ABSTRACT

A teleoperation and demonstration learning system proposal helps in the industrial sector in hard-to-reach areas or production areas. Which represents a latent risk to the workers' integrity and health-giving it a system that simplifies programming. Avoiding unnecessary costs in sophisticated training for robot manipulation. For these reasons, this research seeks to develop a teleoperated system for manipulating a prototype robotic arm based on demonstration learning, for the automation of stacking processes in the industry of concrete.

The use of elements of a teleoperated system was implemented, such as a local station, the communication channel, and a remote station. The communication channel is based on the use of a single world-wide logical network (Internet) and an MQTT server to establish communication between the local station and the remote station. While the remote station presents the integration of the demonstration learning system that bases its operation on the use of components, the KY-040 rotary encoder is the main one.

Which, together with the servo motor, give a positioning ratio of each degree of freedom. Also, embedded systems were used that are responsible for collecting the necessary information in the teaching of movement to be later able to replicate said action.

To visualize the correct operation of the robotic arm. The tests obtained indirect learning were optimal, and repeatability tests were performed with the load. Or, without charge, the same ones that corroborated its functional operation, the data collected from this can be viewed from a monitor serial. The tests in teleoperation can be affirmed that these effectively fulfill all the movements made by the user at the local station. Thus seeing greater fluidity in the actions of the master arm in the remote station had a good response time. It could be verified by tests with MQTTLens software and ping commands, in addition to trajectory tests that were carried out through simulation that helped to determine the margin of error of the robotic arm obtaining favorable results for this project.

**Keywords:** MQTT, Teleoperation, Direct learning, Robotic arm, Thonny Python.



SIGNATURE

Reviewed by: Maldonado, Ana  
Language Center English Professor

## **CAPITULO I**

### **1.1 INTRODUCCIÓN**

El desarrollo y la investigación ha permitido darles a los robots la capacidad de ser multifuncionales y adaptables a diferentes entornos de trabajo y tareas a realizar en donde se requiera precisión y repetición. Modelos de programación y sistemas Teleoperados – Telecontrolados, permiten la automatización de procesos continuos, situándose por encima de las capacidades humanas, además de la posibilidad de que estos procesos sean supervisados. (Tirado Ruiz & Higinio Rubio, 2012)

Entre los modelos de programación que han sido desarrollados hasta la actualidad, se encuentran la programación guiada y la programación directa o también llamada de aprendizaje por demostración. En donde el primer modelo señalado usa controles como joysticks, dispositivos hápticos, etc para poder guiar la movilidad y funcionalidad del robot; el segundo modelo mencionado tiene como principio de funcionamiento la manipulación directa del robot para ser programado, es decir, el operador le enseña de forma manual y demostrativa las acciones que este debe realizar, para que luego el robot ejecute las mismas acciones de forma automática. (Kyrarini, M., Haseeb, M. A., Ristić-Durrant, D., & Gräser, A, 2018).

Uno de los parámetros que ayudan a los robots a ser multifuncionales y adaptables, son los sistemas Teleoperados, los cuales permiten sustituir al personal humano en entornos de trabajo en los que no es conveniente que un humano realice labores debido a que se expone la integridad del mismo. (Hewitt, A., Yang, C., Li, Y., & Cui, R, 2017).

Basado en las conceptualizaciones antes mencionadas, se pretende implementar, un sistema de teleoperación que sea capaz de controlar un brazo robótico, el mismo que cuenta con un funcionamiento orientado en la programación de aprendizaje por demostración, para procesos de producción de apilamiento en la industria de hormigón.

### **1.2 PLANTEAMIENTO DEL PROBLEMA**

El uso de la teleoperación robótica o Telerobótica abarca distintos campos aplicativos que ofrecen facilidades de dirigir tareas específicas desde lugares remotos, cabe mencionar que,

el retraso de transmisión de información es un punto muy importante que debe ser tomado en cuenta para el desarrollo del sistema de control.

Una problemática existente, es el desarrollo de operaciones en zonas de difícil acceso, en entornos no estructurados o en zonas de producción, en las cuales, los empleados exponen su salud al trabajar de forma directa con químicos perjudiciales; también el manejo de residuos biológicos, que se presentan en sectores medicinales y la manipulación de explosivos, que se realizan en sectores mineros. (Breñosa, J., 2016)

Si bien es cierto estos trabajos implican el contacto directo con el ser humano, por tales motivos, la utilización de un sistema teleoperado nos permite controlar el movimiento y la fuerza ejercida de un robot o brazo robótico (slave), el mismo que se encuentra ubicado en una zona remota, es decir que puede estar ya sea a distancias muy lejanas o a su vez en entornos hostiles perjudiciales para el hombre. Por tal razón se debe proteger al operador, a través del manejo de un periférico de entrada, siendo este un joystick o el diseño a escala del mismo brazo robótico(master) que se encuentra localizado en el punto de trabajo del operador. (J.M. Sabater, 2006).

Con esto, el movimiento del brazo robótico debe ser predecible desde el inicio de los movimientos que el operador provoca en la zona local, esto quiere decir que, el operador debe comprender como realizar el movimiento del master para que el slave realice dicha acción de una manera correcta, no obstante, algunas interacciones robot-humano restringen el movimiento que el operario desea realizar, para esto el diseñar un entorno de trabajo es de vital importancia ya que se puede captar visualmente los movimientos que ejecuta el slave en tiempo real (P.K. Pook, 2014) (Julio Astete Rupay, 2016).

En este tipo de sistemas, un aspecto muy importante a mejorar, es el tiempo de respuesta que se da en función al ancho de banda de la red que posibilita la comunicación entre unidades remotas, dicho de otra manera, se debe considerar el tiempo que toma el envío de información, la actuación del brazo robótico y la actualización de la nueva posición del mismo. La trama de información que se envían a dichas unidades se las realiza mediante un canal de comunicación que son las redes de Internet, las mismas que demuestran la viabilidad para controlar robots en tiempo real, con retardos que son menos perceptibles al ser humano (P.K. Pook, 2014).

Cuando hablamos de la temática de diseño de maquinarias automatizadas en las industrias, estas representaban un trabajo muy complejo, debido, a la inexistencia de la programación o algún método de comunicación entre hombre y máquina, por lo que se usaba recursos humanos para la manipulación de las mismas. Con la aparición de dispositivos electrónicos, se incrementó la autonomía de las maquinarias y se redujo parte de los recursos humanos que las manejaban. En la actualidad se lleva a cabo el uso de microcontroladores en las maquinarias industriales, incorporando de forma directa el uso de códigos y lenguajes de programación, provocando así, la combinación de diferentes maquinarias para incrementar aún más la autonomía y disminuir de forma proporcional el recurso humano que las manipulaba (López & De Pro Bueno, 2017).

El uso de la programación mediante plataformas que utilizan códigos y distintos tipos de lenguajes de traducción hombre máquina, son empleados en su totalidad por personas capacitadas en este tema, ya que la realización de esto es lo bastante complejo, siendo incomprensible para personas que no cuentan con los conocimientos adecuados o suficientes. Además, el hecho de que las empresas realicen productos de mejor calidad para satisfacer a los clientes, demanda que los obreros tengan una gran capacidad para realizar trabajos complejos y también para llegar a lugares de difícil acceso, ya sea por el mantenimiento de los equipos o la programación de los mismos (López & De Pro Bueno, 2017).

Los procesos que son muy repetitivos, ocasionan un cansancio laboral en el personal, así mismo, la optimización de recursos es muy bajo debido a que el personal debe estar capacitado para un mejor desempeño, pero la mayoría de las empresas optan por contratar a nuevo personal lo que ocasiona un incremento de costos debido a capacitaciones y salarios de los mismos, todo esto debilita la eficiencia y productividad de una empresa (P.K. Pook, 2014).

Por este motivo el uso de programación directa, reduce esta inaccesibilidad de las personas comunes y corrientes para la programación de distintos tipos de robot que usan actuadores en su funcionamiento, como lo son los brazos robóticos, que son usados para distintos procesos de producción en las industrias, consiguiendo así, la reducción de costes por capacitación o por contratación de personal y el aumento de la eficiencia de producción de la empresa, además de usar este tipo de tecnología para generar sistemas de respaldo que ayuden a mejorar la eficacia de las empresas de producción industrial.

Los beneficios de los sistemas teleoperados con entornos de visualización para el manejo de robots industriales en zonas remotas, además del desarrollo de un prototipo de brazo robótico el cual permite la programación por demostración para su funcionamiento, justifican el desarrollo de un sistema teleoperado para la manipulación de un prototipo de brazo robótico basado en el aprendizaje por demostración, generando así un sistema de teleoperación en donde se elimina la necesidad de que un operador humano se encuentre en la ubicación del robot. Pretendiendo lograr que, el mismo pueda realizar las acciones de control del robot de forma remota y de cierta manera proporcionar la posibilidad de compartir el control entre un grupo limitado de operadores localizados en el sitio o lugares distintos. Conservando así, la programación del brazo robótico basado en aprendizaje por demostración y del sistema teleoperado.

### **1.3 OBJETIVOS**

#### **1.3.1 OBJETIVO GENERAL**

1. Desarrollar un sistema teleoperado para la manipulación de un prototipo de brazo robótico basado en el aprendizaje por demostración, para la automatización de procesos de apilamiento en la industria de hormigón.

#### **1.3.2 OBJETIVOS ESPECÍFICOS**

2. Diseñar un sistema de teleoperación con tiempos de retardo mínimos y que permita la visualización de los movimientos al realizar la manipulación del brazo robótico.
3. Fundamentar el proceso teórico acerca de la programación para procesos de enseñanza de aprendizaje de robots.
4. Crear un prototipo de brazo robótico con 4 grados de libertad que utilice el aprendizaje por demostración basado en la programación directa, para procesos de apilamiento de adoquines en industrias de hormigón.

## CAPITULO II

### 2.1 ESTADO DEL ARTE

El uso de la programación en robot o sistemas autónomos han experimentado un aumento importante en las últimas décadas, en donde diferentes empresas y universidades han definido diferentes campos aplicativos del aprendizaje por demostración, sea en vehículos autónomos de transporte, exploración (Kuderer, Gulati, & Burgard, 2016), drones autónomos desarrollados por la NASA (T. H. Cox, 2004), robot humanoides; y otros tipos de robots usados en medicina (Lauretti, Cordella, Guglielmelli, & Zollo, 2017), residencia (WANG, CHEN, Y K LAU, & REN, 2016) e industria (Kramberger, 2014).

Kyrrarini, Haseeb, Ristić-Durrant y Gräser manifiestan que el aprendizaje por demostración genera un creciente interés por la colaboración humano-robot en la fabricación de procesos industriales, estos procesos requieren que el robot aprenda la secuencia de acciones con aprendizaje de alto nivel o bajo nivel. El aprendizaje de alto nivel es necesario para realizar una tarea de ensamblaje que incluye el movimiento de objetos y las acciones de agarre / liberación sin pre-programación; a diferencia del aprendizaje de bajo nivel, en donde la tarea se indica en demostraciones humanas múltiples a través de la enseñanza kinestésica, en ambos casos el robot aprende la secuencia de acciones y las rutas del efector final necesarias para completar la tarea de ensamblaje. El aprendizaje de alto nivel o también llamado aprendizaje simbólico, consiste en que la tarea se codifique de acuerdo con las secuencias de elementos de movimientos predefinidos, esto permite al robot aprender la secuencia de acciones, mientras que el aprendizaje de bajo nivel se basa en la deformación dinámica del tiempo (DTW), ya que es uno de los algoritmos para medir la similitud entre dos secuencias temporales que pueden variar en velocidad, el robot realiza la secuencia aprendida de acciones en diferente orden de aparición después de la sugerencia del colaborador humano, si el colaborador humano está satisfecho con la nueva secuencia de acciones, el robot lo aprende (Kyrrarini, Haseeb, Ristić-Durrant, & Gräser, 2018).

Las demostraciones de movimientos de bajo nivel son difíciles en sistemas de control de un motor complejo, como lo son, los humanoides de alto grado de libertad, como resultado de esto, se han aplicado técnicas de aprendizaje automático, Argall, Chernova, Veloso y Browning desarrollaron una encuesta sobre el aprendizaje de robots a partir de demostraciones donde enuncian que: la teleoperación proporciona el método más directo para la transferencia

de información dentro del aprendizaje de demostración. Sin embargo, se dice que la teleoperación requiere que el robot sea manejable y, como resultado, no todos los sistemas son adecuados para esta técnica. A pesar de ello, existen demostraciones registradas mediante la teleoperación humana a través de un joystick, que se utiliza en una variedad de aplicaciones, como volar un helicóptero robótico (Ng, A. Coates, M. Diel, V. Ganapathi & J. Schulte), movimientos de patada de robot (Browning, L. Xu & M. Veloso), agarrar objetos (P.K. Pook & D.H. Ballard) (J.D. Sweeney & R.A. Grupen), tareas de ensamblaje de brazos robóticos (J. Chen & A. Zelinsky) y evitar obstáculos y la navegación (T. Inamura, M. Inaba & H. Inoue)(W.D. Smart). La teleoperación también se aplica a una amplia variedad de dominios simulados, que van desde laberintos estáticos (J.A. Clouse) (R.P.N. Rao, A.P. Shon & A.N. Meltzoff) a la conducción dinámica (P. Abbeel & A.Y. Ng) (Chernova & M. Veloso) y dominios de fútbol (R. Aler, O. Garcia & J.M. Valls), y muchas otras aplicaciones.

## **2.2 ROBÓTICA Y TELEOPERACIÓN**

Robótica es un campo multidisciplinario de la ingeniería encargado de la investigación y desarrollo de robots. En el mismo convergen distintas áreas como la mecánica, automatización, informática, electrónica. En la actualidad, existen distintos tipos de robots que varían desde su arquitectura y aplicación que van desde los robots industriales hasta robots humanoides.

Dependiendo de la autonomía del robot, estos pueden ser divididos en teleoperados de funcionamiento repetitivo y, autónomos o inteligentes. Se puede definir como teleoperación, a la manipulación del robot ubicado en una zona remota desde una zona local en donde se encuentra un operador humano.

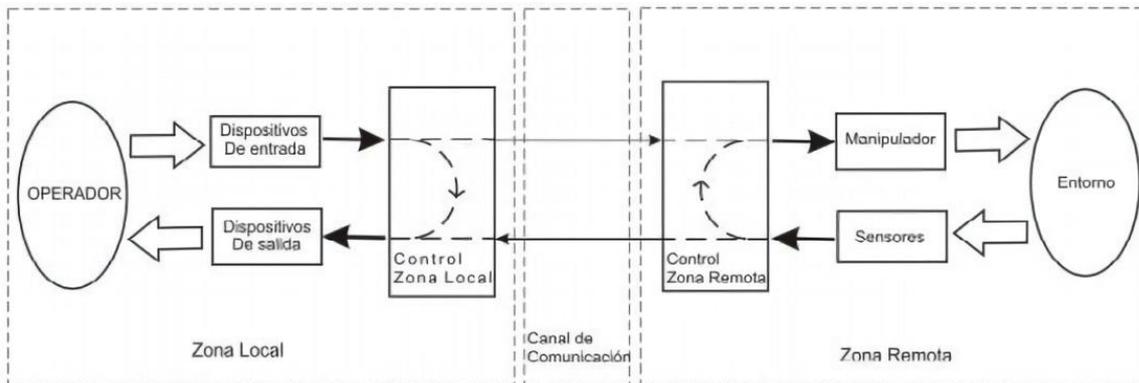
## **2.3 COMPONENTES DEL SISTEMA TELEOPERADO DE UN ROBOT**

El funcionamiento de un sistema teleoperado de robots se puede describir cuando un operador o master envía comandos a un robot que se encuentra en un lugar remoto, el cual ejecuta las instrucciones correspondientes, este a su vez realimenta al operador con datos que permiten saber el estado actual de robot.

Los elementos que conforman un sistema teleoperado de robots son:

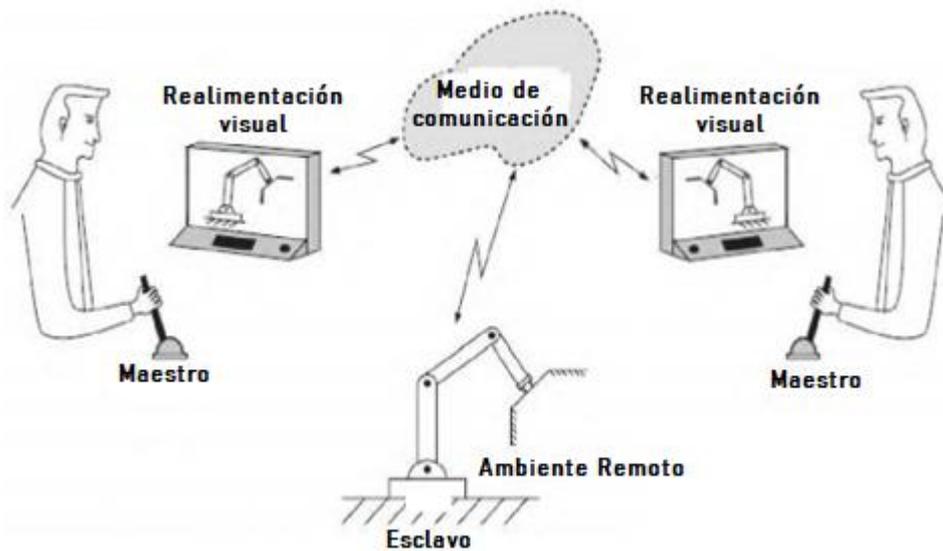
1. Zona Local o Estación de Teleoperación

2. Zona Remota o Esclavo (Brazo Robótico)
3. Canal de Comunicación



*Ilustración 1. Elementos de un sistema Teleoperado*

*Fuente: (P.K. Pook, 2014)*



*Ilustración 2. Esquema General de un Sistema de Teleoperación*

*Fuente: Autores*

### 2.3.1 ZONA LOCAL O ESTACIÓN DE TELEOPERACIÓN

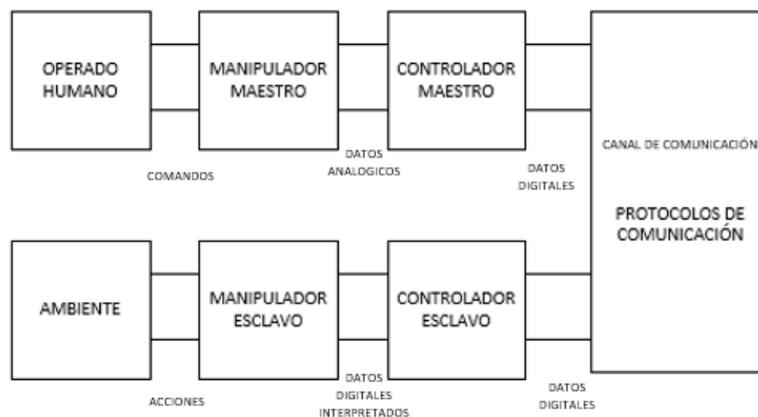
En la estación de teleoperación es donde se encuentran los operadores humanos quienes realizan la operación del robot de forma remota. La estación de teleoperación debe contar además con dispositivos de actuación los cuales permiten generar los comandos del operador que serán enviados al robot que se encuentra en la zona remota. Para que el operador tenga una idea del trabajo que se está realizando en la estación remota, esta deberá contar con una interface de retroalimentación la cual envíe información del estado de las tareas y su ejecución.

### 2.3.2 ZONA REMOTA O ESCLAVO (BRAZO ROBÓTICO)

Este es el entorno en el cual se encuentra ubicado el brazo robótico basado en el aprendizaje directo y el lugar donde se encuentra localizado e interactúa en el entorno remoto de trabajo. Es importante resaltar que, a diferencia del robot industrial, el entorno de trabajo de un robot usado como manipulador teleoperado, por lo general es no estructurado, variable o desconocido. Debido a que el operador humano es el que cierra el bucle de control del manipulador esclavo, el mismo no requiere de gran precisión en sus movimientos ya que el operador realizara las correcciones necesarias.

### 2.3.3 CANAL DE COMUNICACIÓN

El canal de comunicación se define como el medio por el cual un transmisor y un receptor comparten e intercambian señales de información (mensajes).



*Ilustración 3. Diagrama de Bloques de un Sistema de Teleoperación Típico*

*Fuente. (P.K. Pook,2014)*

En la actualidad el canal de comunicación utilizado para la teleoperación más común es el internet el cual cuenta con diferentes tipos de arquitectura de red entre lo más usados son:

- Atmósfera (ondas electromagnéticas)
- Red de fibra óptica
- Red de cobre

Es necesario que estos cuenten con protocolos que ayuden a que el intercambio de la información entre transmisor y receptor sea exitoso.

### **2.3.3.1 PROTOCOLOS DE COMUNICACIÓN**

Un protocolo de comunicación describe las reglas de comunicación entre un transmisor y receptor, es decir que un protocolo de comunicación es un estándar que rige cada una de las normas con las que deben cumplir las señales de información, el transmisor y el receptor, además cuenta con parámetros que ayudan a que la comunicación se establezca con éxito.

Existen diferentes protocolos de comunicación entre los cuales se tiene:

- HTTP
- TCP/IP
- SERIAL
- RF
- SCP
- POP3

Los protocolos de comunicación actúan en la capa 3 del modelo OSI, el mismo que establece un conjunto de acuerdos para el intercambio de datos, regulando así, las condiciones para el transporte, direccionamiento, enrutamiento que es el camino por donde es dirigido el paquete y el control de fallos (Digital Guide IONOS, 2019).

## **2.4 METODOLOGÍA DE ENSEÑANZA PARA ROBOTS**

En esta sección se da a conocer los tipos de sistemas de control y los entornos de programación de un brazo robótico para la enseñanza por aprendizaje directo.

### **2.4.1 MÉTODOS DE PROGRAMACIÓN DE UN ROBOT**

Para que un brazo robótico funcione correctamente, es necesario tener un sistema que regule sus movimientos. Según su sistema de control o CIM (Manufactura Integrada por computadora), (s.f.), existen las 3 categorías siguientes:

- Robot de secuencia limitada (baja tecnología)
- Robots de reproducción (mediada tecnología)
- Robots inteligentes (alta tecnología)

#### **2.4.1.1 ROBOTS DE SECUENCIA LIMITADA (BAJA TECNOLOGÍA)**

Los robots de secuencia limitada representan el nivel de control más bajo, se controlan por el posicionamiento de interruptores de fin de carrera y/o topes mecánicos para establecer los puntos finales de desplazamiento para cada una de sus articulaciones. Con este método de

control, las articulaciones individuales solo pueden desplazarse a sus límites de desplazamiento extremos. Las aplicaciones de este robot suelen implicar movimientos simples, tales como operaciones de “coger y sujetar”.

#### **2.4.1.2 ROBOTS DE REPRODUCCIÓN (MEDIADA TECNOLOGÍA)**

Estos tipos de robots utilizan para su reproducción una unidad de control más sofisticada, en la que una serie de posiciones o movimientos los cuales son enseñados al robot, se registran en una memoria y luego son reproducidos por el robot bajo su propio control. El proceso de enseñar y registrar en una memoria se la determina como la programación del robot.

#### **2.4.1.3 ROBOTS INTELIGENTES (ALTA TECNOLOGÍA)**

Los robots inteligentes con los avances tecnológicos forman una cantidad cada vez más numerosa dentro de los robots industriales, ya que no solo reproducen un ciclo de movimiento programado, sino interactúa con su entorno de una manera que parece inteligente. Los robots inteligentes pueden acoplar su programación en respuesta a las condiciones del entorno de trabajo, incluyendo operadores humanos o sistemas computacionales.

### **2.4.2 ENTORNO DE PROGRAMACIÓN**

Se le llama método de programación o entorno de programación, al conjunto de procedimientos que son necesarios para la enseñanza del robot, el mismo que emplea un listado de acciones que debe llevar a cabo con mayor precisión. Existen dos métodos de programación que son: la programación guiada y la textual; en las cuales, el programador es el principal responsable de introducir de forma manual las distintas instrucciones que gobernarán al robot, mediante el uso del lenguaje textual o gráfico. Para la programación guiada, la persona que ejecuta los códigos tiene una participación casi nula, ya que el robot genera un aprendizaje empírico en base a un sistema guiado (CIM (Manufactura Integrada por computadora), s.f.).

#### **2.4.2.1 PROGRAMACIÓN TEXTUAL**

La programación textual, permite que la persona que opera un sistema robótico, pueda indicar el trabajo a realizar mediante un lenguaje de programación en específico, es decir, dar un listado de instrucciones al sistema y que sea ejecutado por el sistema robótico. Dicha programación se la realiza sin la necesidad de la presencia de un robot, en modo off – line. El

operador no realiza las acciones del brazo o del robot, sino que, define el cálculo matemático de una manera precisa y según las características del lenguaje pueden crearse programas que son lo bastante complejos, evitando así, el posicionamiento empírico del sistema.

#### **2.4.2.2 PROGRAMACIÓN GUIADA**

La programación guiada también es conocida como aprendizaje asistido, y radica principalmente en la manipulación directa de los diferentes actuadores del robot basados en un control o maqueta del sistema, que lo posicionan en distintos puntos, gracias a la ayuda del controlador embebido, lo cual permite registrar movimientos y trayectorias para la ejecución de las secuencias de repetición. Dicho proceso de programación requiere que el robot esté conectado, es decir de manera on – line.

##### **2.4.2.2.1 UTILIZACIÓN DE DISPOSITIVOS DE ENSEÑANZA**

El aprendizaje mediante dispositivos de enseñanza cuenta con controles de mando, que llevan al robot hacia puntos donde este pasará de forma automática en el instante en que se ejecute el programa. Al igual que el aprendizaje directo, los operadores no necesitan conocer de algún lenguaje de programación específico, con solo poder familiarizarse con el empleo de elementos que están inmersos en el dispositivo de enseñanza, pueden realizar ya, las diferentes secuencias que están dentro del programa.

##### **2.4.2.2.2 APRENDIZAJE DIRECTO**

Se le denomina aprendizaje directo a la manipulación del brazo maestro para llevarlo hasta puntos específicos de programación, durante un intervalo de ejecución automática. En este proceso la unidad de control del sistema, interpola los puntos establecidos, según la trayectoria determinada. Dichos puntos son recorridos de manera secuencial, es decir, en el mismo orden en el que son programadas.

## **CAPITULO III**

### **3. METODOLOGÍA DE LA INVESTIGACIÓN**

#### **3.1 TIPO DE INVESTIGACIÓN**

##### **3.1.1 INVESTIGACIÓN EXPERIMENTAL**

Esta tesis es de tipo experimental, ya que se implementó un prototipo de brazo robótico y un sistema teleoperado, con el fin de evaluar el resultado, tanto de la sustitución de robots de programación textual como la baja latencia de la red, mediante una comunicación en tiempo real del prototipo de brazo robótico esclavo con el master, para evaluar los retardos de una comunicación entre estaciones remotas.

##### **3.1.2 INVESTIGACIÓN CUALITATIVO**

Esta tesis es de tipo cualitativo, ya que se utiliza para examinar el comportamiento de los movimientos del brazo robótico, evaluando la precisión que tiene al momento de ejecutar las instrucciones generadas por el operador.

#### **3.2 UNIDAD DE ANÁLISIS**

##### **3.2.1 POBLACIÓN DE ESTUDIO**

Debido al tipo de proyecto no es posible aplicar una población de estudio.

#### **3.3 TÉCNICAS DE RECOLECCIÓN**

##### **3.3.1 OBSERVACIÓN**

En esta tesis se empleó la técnica de observación, que consiste en capturar y visualizar de manera más detallada los retardos en una red y la precisión de los movimientos de un prototipo de brazo robótico basado en el aprendizaje por demostración.

##### **3.3.2 FUENTE DE RECOPIACIÓN DE INFORMACIÓN**

Para el desarrollo de la parte teórica de esta tesis se utilizó principalmente la información de la red ya que es la primera herramienta mundial con mayor cantidad de avances diarios en segundos de tiempo, llevándolo así a la búsqueda en libros relacionados a la temática, revistas confiables, paper, blogs direccionados a la requerido en los objetivos, fichas técnicas que contengan las especificaciones de los dispositivos electrónicos, páginas web entre otras tesis

relacionadas. Toda esta información recolectada de las diferentes fuentes antes mencionadas, tiene su respectiva referencia.

### **3.4 INSTRUMENTOS DE LA INVESTIGACIÓN**

Para el desarrollo de esta tesis los instrumentos utilizados son los siguientes: procedimientos experimentales y análisis de documentos.

En los procedimientos experimentales: Arduino es una plataforma abierta que nos permite la programación de un microcontrolador, para el sistema de control con su respectivo cable de conexión, Autocad para la realización de las extremidades del brazo robótico, Raspberry es un ordenador de placa reducida que se utiliza para el sistema de control con su respectivo cable de conexión, Sistema Operativo Raspbian, controlador de servos, tarjeta ethernet es un componente de hardware que conecta una computadora a un red para el envío de datos con su respectivo cable de conexión, Solidwork para el diseño de los engranes y piñones.

En el análisis de documentos: artículos científicos publicados, revisión bibliográfica y datasheet.

### **3.5 HIPÓTESIS**

El aprendizaje por demostración de un sistema teleoperado para la manipulación de un prototipo de brazo robótico, permitirá la automatización de procesos de apilamiento en la industria de hormigón.

### **3.6 METODOLOGÍA**

La metodología para el desarrollo de un sistema teleoperado y la manipulación de un prototipo de brazo robótico basado en el aprendizaje por demostración, se presenta a continuación:

Por medio de la investigación en páginas web de software libre y en artículos relacionados con la construcción de brazos robóticos, se encontró un boceto ya elaborado por un autor en la web (fortzero, 2019), el cual es utilizado para llevar a cabo el diseño del brazo robótico según las necesidades que se requieran.

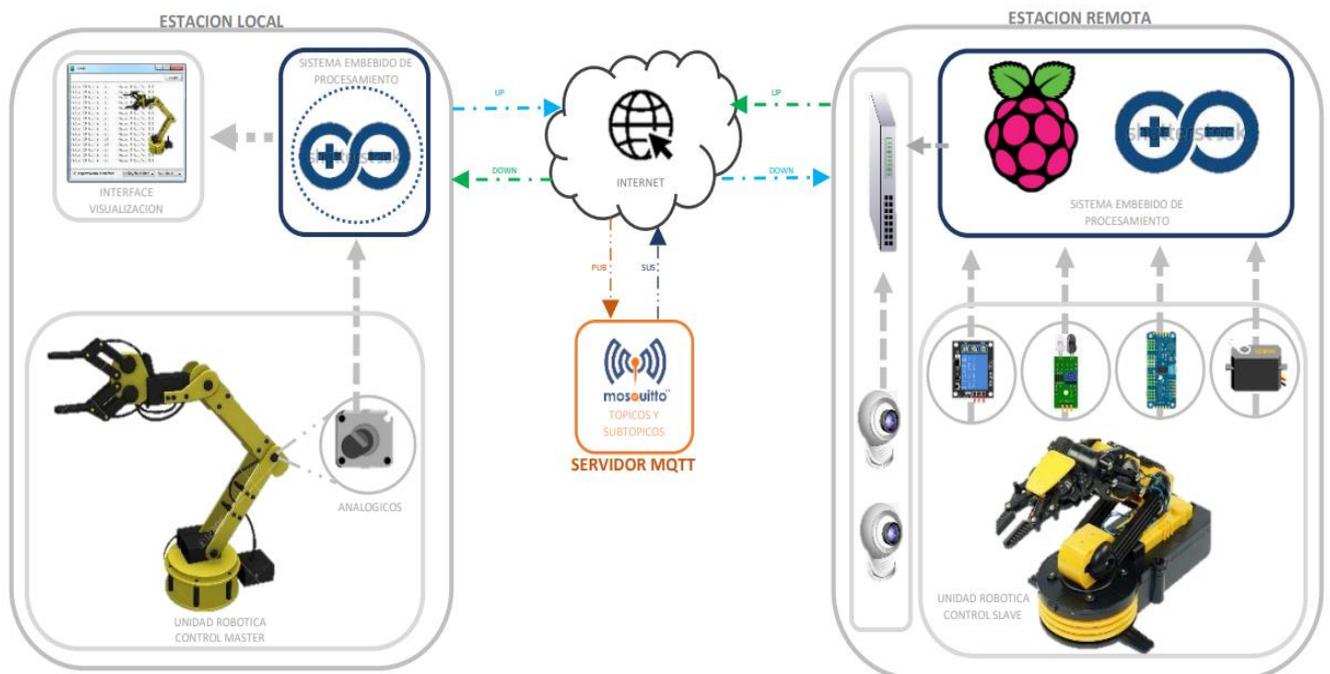
A través de la consulta en artículos científicos relacionados con los métodos de aprendizaje para un brazo robótico y conocimientos teóricos sobre electrónica, se encontró un artículo del

autor (CIM (Manufactura Integrada por computadora), s.f.), el cual nos indica la ubicación del brazo robótico dentro de una categoría de aprendizaje directo.

Mediante la investigación en la web y sitios científicos de temas relacionados a servidores libres para realizar la teleoperación, se encontró un artículo del autor (Berra, 2018), del cual se tomó en consideración el servidor MQTT como principal opción, para que, a través del mismo, el brazo de control suba los datos de los movimientos, y estos sean recibidos por el brazo master para que ejecute la tarea de teleoperación.

Y por último para comprobar el buen funcionamiento del proyecto a realizar, con respecto al aprendizaje por demostración se va analizar la recolección y reproducción en cada grado de libertad para analizar la posición en función del tiempo y la velocidad de enseñanza que viene dado por el usuario, y con respecto al sistema de teleoperación para comprobar el buen tiempo de respuesta, se lo realizara mediante el software MQTTLens y comandos ping.

### 3.8 PROCEDIMIENTOS Y ANÁLISIS



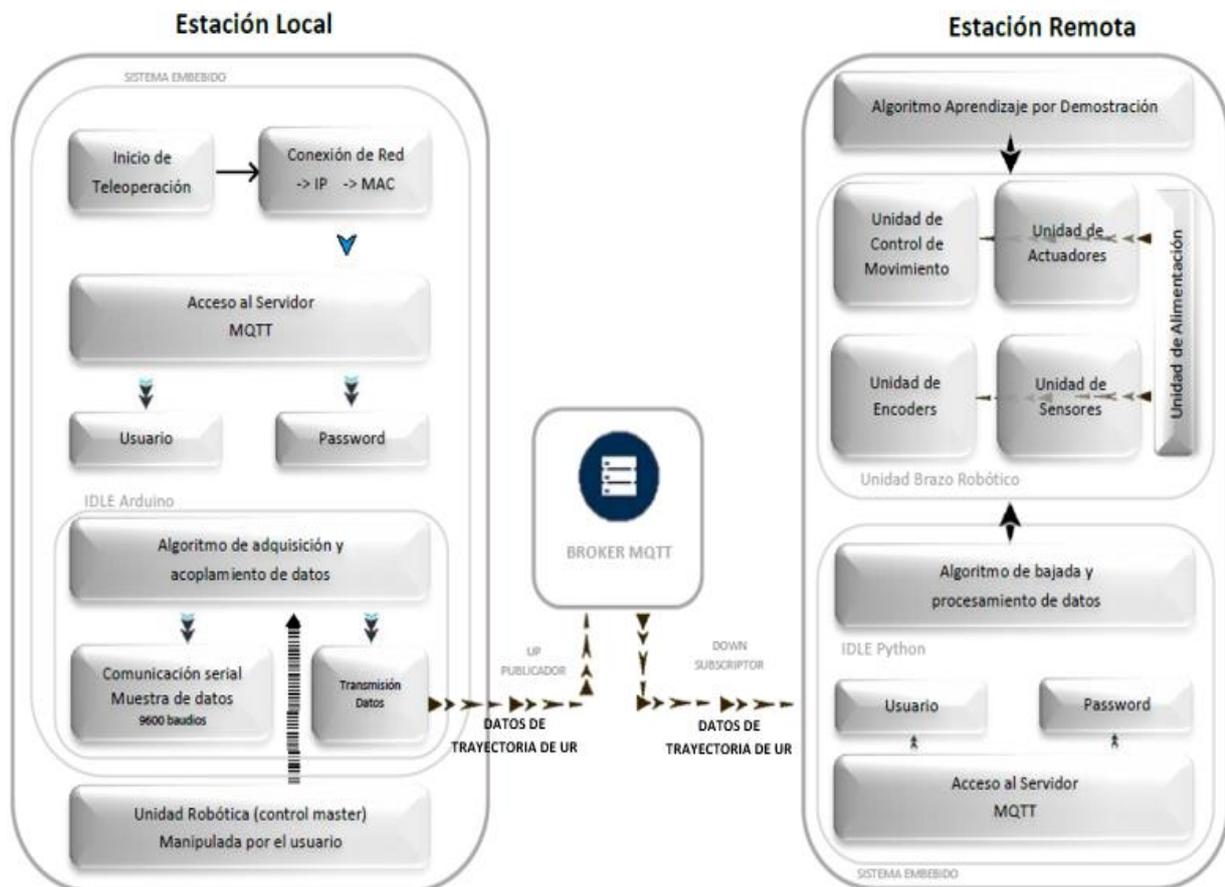
**Ilustración 4.** Prototipo experimental del prototipo de brazo robótico Teleoperado con aprendizaje por demostración

(Esquema General).

**Fuente:** Autores

Con la finalidad de poder cumplir los objetivos planteados en este proyecto se realizará los siguientes procedimientos:

- Bróker MQTT
  - ✓ Lectura y escritura sobre el servidor MQTT
- Estación Local
  - ✓ Algoritmo de adquisición y acoplamiento de datos.
  - ✓ Unidad robótica o brazo robótico de control.
- Estación Remota
  - ✓ Algoritmo de bajada y procesamiento de datos.
  - ✓ Algoritmo de aprendizaje por demostración.
  - ✓ Unidad robótica o brazo robótico master.



*Ilustración 5. Diagrama general del sistema de Teleoperación y Aprendizaje por Demostración.*

*Fuente: Autores*

### 3.8.1 BRÓKER MQTT

La utilización del bróker o servidor MQTT (Message Queue Telemetry Transport) en este proyecto, debido a las ventajas y beneficios que este representa, entre las cuales se pueden destacar son

- Al ser un protocolo de transporte Cliente/Servidor, permite publicaciones y suscripciones a tópicos.
- Establece una conexión TCP única, la cual es reutilizada de forma permanente, esto permite reducir el tráfico de red y optimizar el canal de comunicación.
- Cuenta con comunicación bi-direccional y sin pérdida de datos.
- Permite la creación de tópicos y subtópicos sin límite, por lo que son óptimos para aplicaciones de sensorización.

Para este proyecto se desarrolló tanto un servidor MQTT local y como un servidor MQTT por hosting los cuales realizan las mismas funciones y presentan los mismos beneficios.

#### 3.8.1.1 LECTURA Y ESCRITURA EN EL SERVIDOR MQTT

Para que los datos que son procesados en la estación local puedan ser almacenados en el servidor y posteriormente puedan ser transportados hacia la estación remota, es necesario el uso de una plataforma que permita tanto la lectura como la escritura, además de la actualización de datos de forma constante. Con este fin se ha escogido “Humble Hedgehog” ya que consta con una base de datos en tiempo real la cual permite la compatibilidad con diferentes interfaces.



*Ilustración 6. Estructura del servidor MQTT*

*Fuente: Autores*

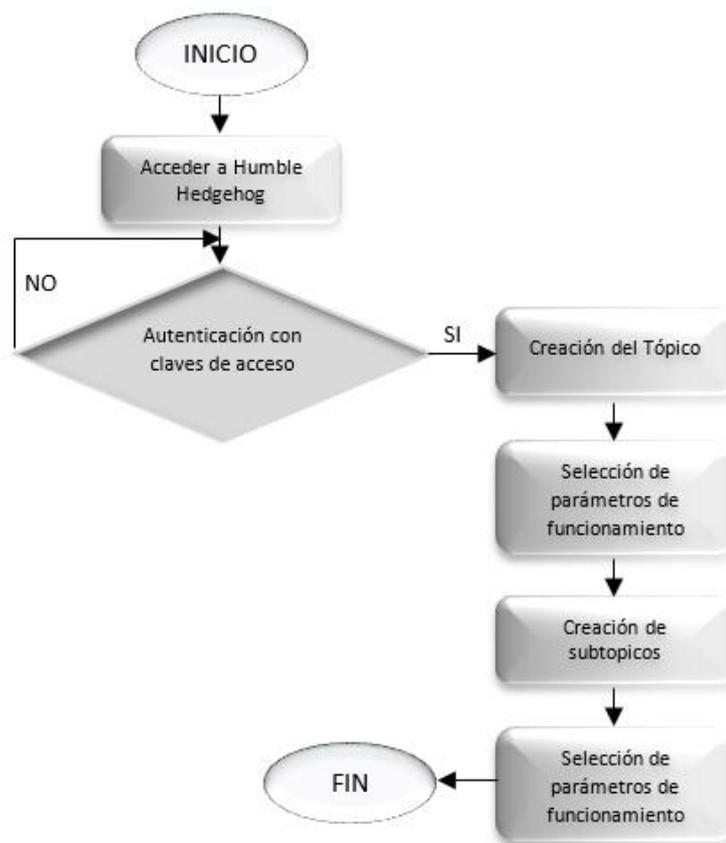
### 3.8.1.1.1 CREACIÓN DEL TÓPICO Y SUBTÓPICOS SOBRE EL SERVIDOR MQTT

*Tabla 1. Tópicos y subtópicos del servidor MQTT*

*Fuente: Autores*

Tópico			
/Brazo Robótico			
Subtópicos			
/Base	/Hombre	/Codo	/Actuador
Almacenamiento	Almacenamiento	Almacenamiento	Almacenamiento

La creación del tópico sobre el Bróker MQTT se realiza al acceder a la web de la plataforma Humble Hedgehog, la cual es de forma intuitiva permite acceder con claves de acceso (user y password) hacia el servidor. Una vez dentro del servidor se procede a crear el tópico principal sobre el cual se almacenan los datos provenientes de la estación local.

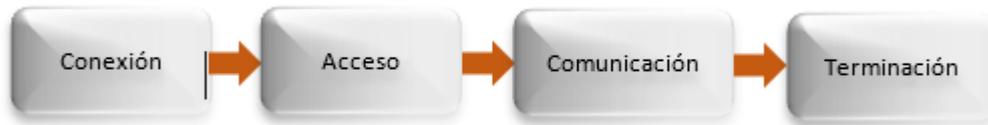


*Ilustración 7. Diagrama general para la creación de tópicos y subtópicos*

*Fuente: Autores*

### 3.8.1.1.2 ESCRITURA EN LOS SUBTÓPICOS DEL SERVIDOR MQTT (PUBLICACIÓN)

Una vez creados los tópicos y subtópicos sobre el servidor MQTT, se sigue una secuencia de fases los cuales permiten la escritura los mismos, los cuales se mencionan en la siguiente ilustración.

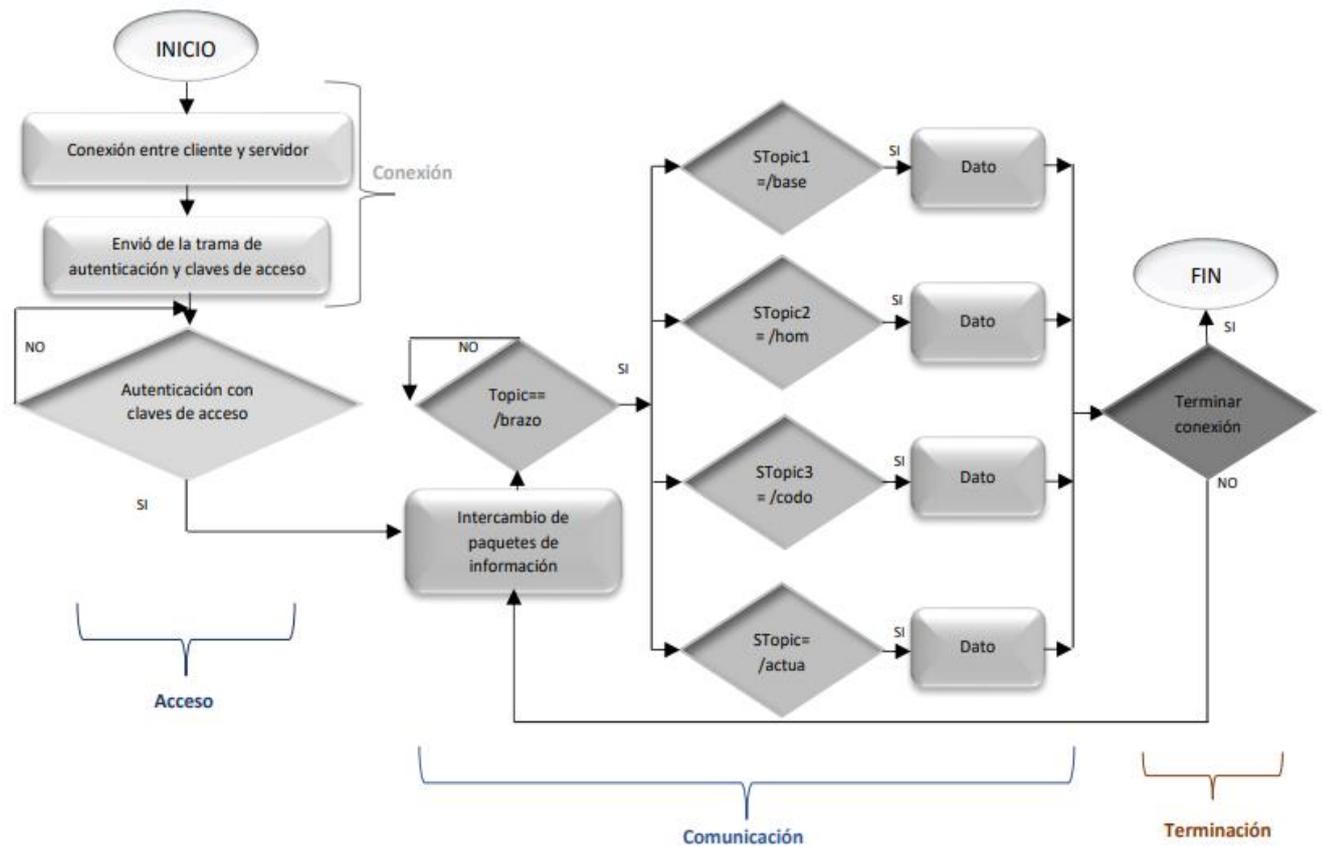


*Ilustración 8. Diagrama de fases del proceso de tareas del servidor MQTT (Publicación, Suscripción, Ping).*

*Fuente: Autores*

Como se puede observar en la ilustración anterior cuenta con 4 etapas las cuales permiten la escritura o publicación del cliente MQTT (Estación Local) sobre el cada uno de los subtópicos del bróker MQTT con el fin de que se encuentren a disposición del cliente MQTT (Estación Remota), a continuación, se detalla las etapas mencionadas:

- **Conexión.** - Permite al cliente crear una conexión de TCP/IP con el bróker utilizando el puerto 1883 o 8883 determinado para la misma, cabe recalcar que el servidor reutilizara la conexión establecida para él envío constante de flujo de datos.
- **Acceso.** - realiza una autenticación ya sea con claves de acceso (user, password) o certificados del servidor, lo cuales al ser validados por el servidor dándole un estado de “CONNECT/CONNACK” lo que permite el intercambio de paquetes de información entre el servidor y el cliente, reutilizando la conexión creada.
- **Comunicación.** – Durante esta fase los clientes tienen la potestad de realizar las tareas de suscripción, publicación y ping sobre los tópicos del servidor, es importante aclarar que los clientes deben definir de forma jerárquica la dirección de tópico o Subtopico sobre el cual desean realizar la tarea.
- **Terminación.** – Esta fase permite al cliente terminar la sesión MQTT, enviando un mensaje de “DISCONEXION” al servidor.

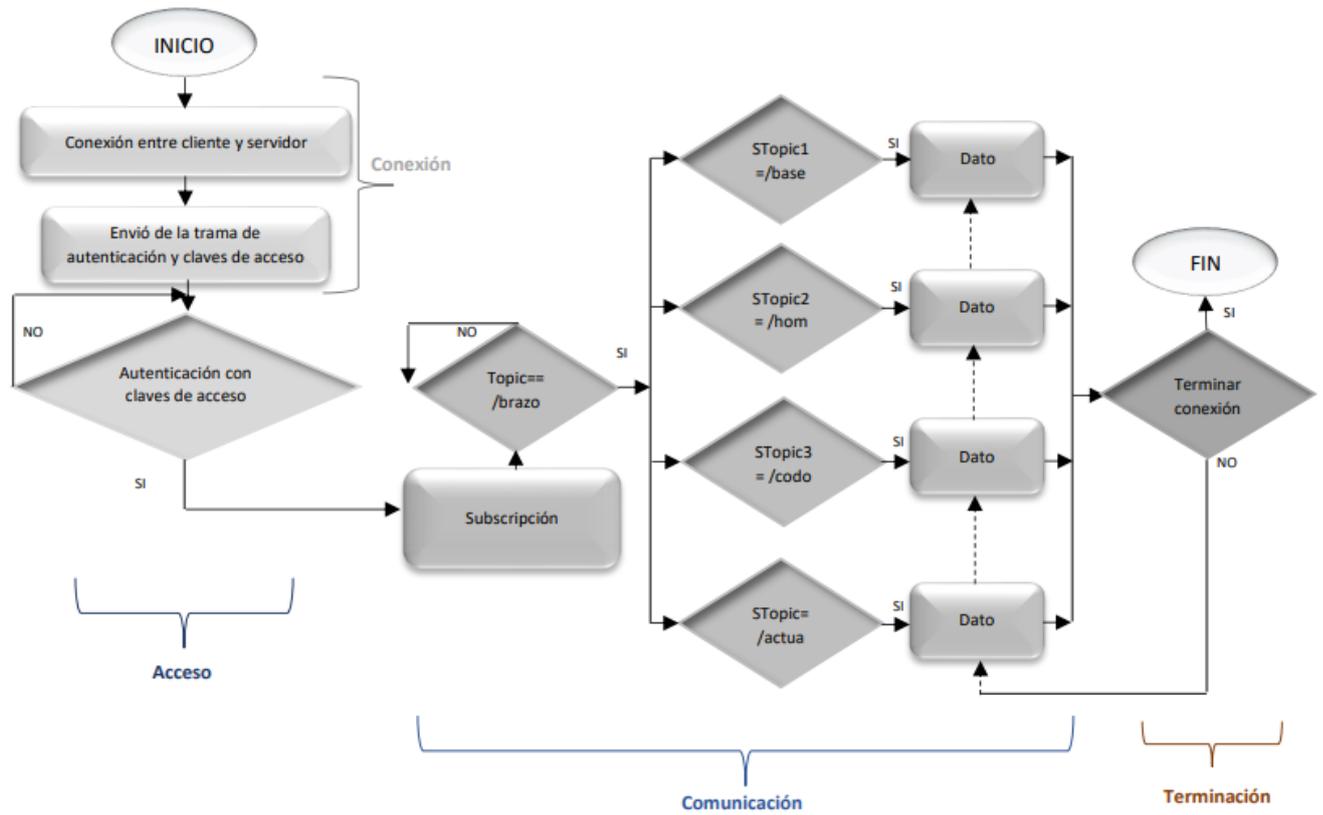


*Ilustración 9. Diagrama del proceso de Publicación en el bróker MQTT.*

*Fuente: Autores*

### 3.8.1.1.3 LECTURA EN LOS SUBTÓPICOS DEL SERVIDOR MQTT (SUBSCRIPCIÓN)

Finalmente, para la lectura de datos el cliente MQTT ubicado sobre la Estación Remota, realizará una subscripción sobre los subtópicos mencionados en la tabla 1, con el objetivo de establecer una comunicación en tiempo real con la Estación Local, para esto se sigue el proceso que detalla en la ilustración 8. Se realiza una variación sobre la etapa de comunicación la cual consiste en el cambio de tarea, es decir la sustitución de “PUBLICACIÓN” por “SUBSCRIPCIÓN”, manteniendo las fases de conexión, acceso y terminación tal como se muestra en la ilustración.

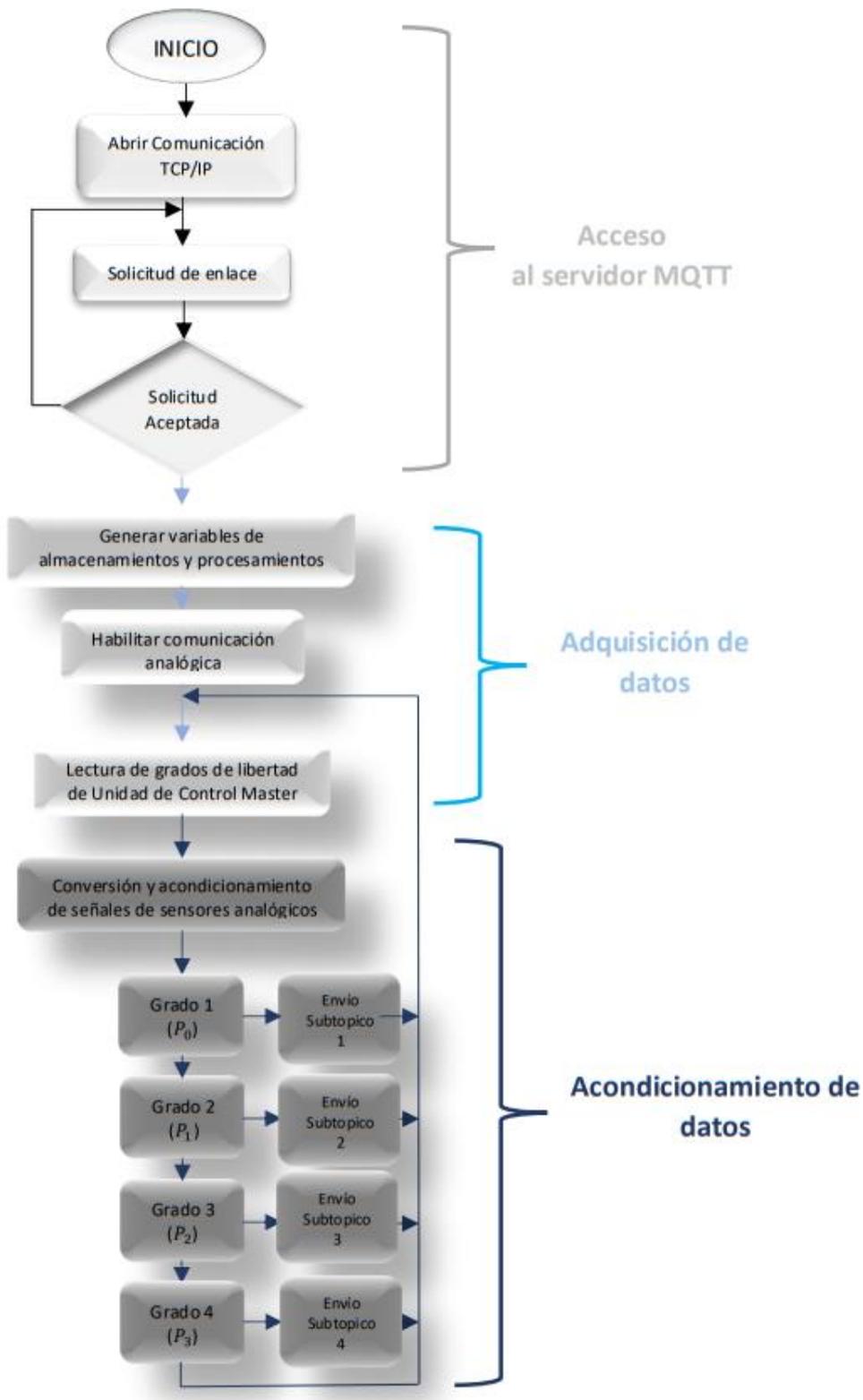


*Ilustración 10. Diagrama del proceso de Subscripción en el bróker MQTT.*

**Fuente:** Autores

### 3.8.2 ESTACION LOCAL

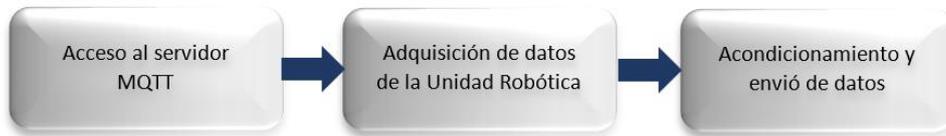
#### 3.8.2.1 ALGORITMO DE ADQUISICIÓN E INTERPRETACIÓN DE DATOS



*Ilustración 11. Algoritmo de adquisición y acoplamiento de datos.*

*Fuente: Autores*

Como se puede observar en la ilustración anterior el algoritmo cuenta con 3 etapas las cuales permiten recolectar los datos de la unidad robótica que es manipulada por el usuario, para posteriormente poder ser interpretadas y acondicionadas con el fin de subir al bróker MQTT.

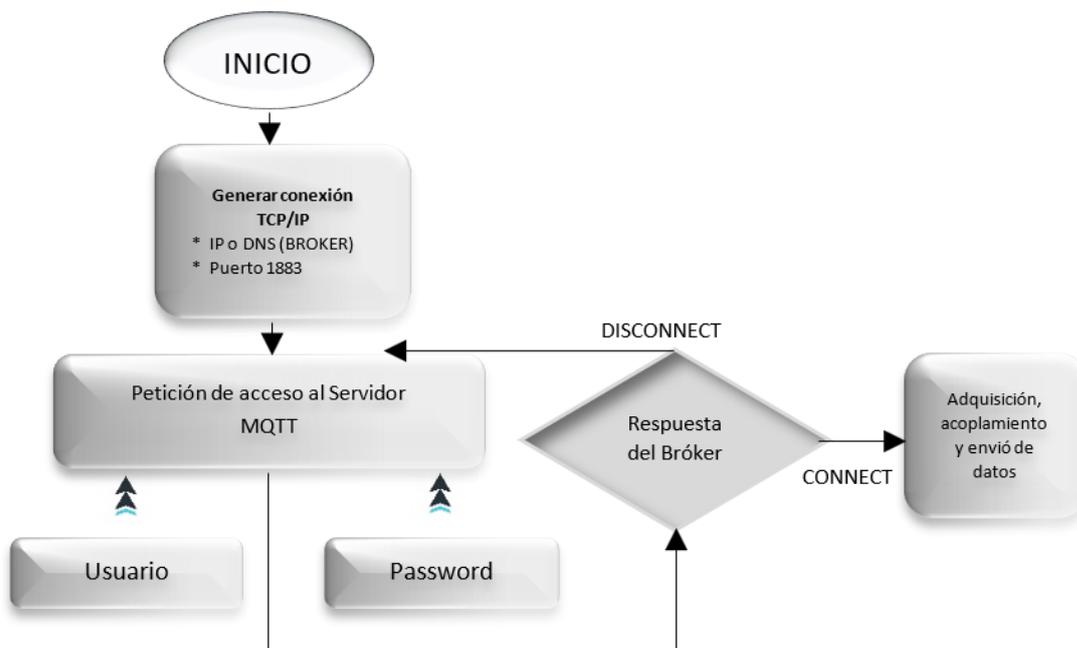


*Ilustración 12. Parámetros generales para la adquisición, acondicionamiento y envío de datos.*

*Fuente: Autores*

### 3.8.2.1.1 ACCESO AL SERVIDOR MQTT

Para poder realizar los procesos de adquisición, acondicionamiento y envío de datos es necesario establecer un acceso con el servidor MQTT. Este proceso se realiza mediante una petición de conexión hacia la dirección del bróker a través del protocolo de comunicación TCP/IP a través del puerto 1883. Ya establecida la conexión, el algoritmo enviará un usuario y contraseña para poder realizar una autenticación, a esto el servidor responderá con un mensaje de estado “CONNECT” o “DISCONNECT” según sea el caso, en caso de que la respuesta sea negativa el algoritmo reenviará la petición de conexión hasta establecerla.



*Ilustración 13. Diagrama de acceso al servidor.*

*Fuente: Autores*

### 3.8.2.1.2 ADQUISICIÓN DE DATOS

Una vez establecido el acceso al bróker MQTT, se comienza con la adquisición de datos, para esta tarea Arduino habilitará 4 pines de lectura analógica, uno por cada grado de libertad (GDL) con el que cuenta la unidad o brazo robótico de control, posterior a esto el algoritmo inicia y mantiene de forma infinita la lectura de los 4 componentes electrónicos lineales (potenciómetros) que cumplen la función de sensar los valores de movimiento generados en la Unidad Robótica. Para cumplir con este objetivo Arduino utiliza la instrucción “*AnalogRead (pin\_analógico)*”.

*Tabla 2. Asignación de sensores y pines para la adquisición de datos*

*Fuente: Autores*

LECTURA DE SENSORES ANALÓGICOS			
# GRADO	NOMBRE DE GDL	PIN ANALÓGICO	UNIDAD DE ENTRADA
GDL 1	Base	$A_0$	V
GDL 2	Hombro	$A_1$	V
GDL 3	Codo	$A_2$	V
GDL 4	Actuador	$A_3$	V

Una vez asignada la lectura infinita de cada uno de los grados de libertad Arduino realizara la conversión de los valores analógicos expresados en voltajes a datos digitales a través de un conversor ADC, para que estos puedan ser procesado por la etapa de acondicionamiento del algoritmo.

### 3.8.2.1.3 ACONDICIONAMIENTO DE DATOS

Adquiridos ya los datos de los componentes sensores de la unidad o brazo robótico de control es necesario que sean acondicionados, para esto Arduino usará la instrucción “*map (límite inferior, equivalencia inferior, límite superior, equivalencia superior)*”:

- **Límite inferior.** – Permite establecer un límite inferior del valor de lectura realizado por los pines analógicos de Arduino.
- **Equivalencia inferior.** – A partir del límite inferior genera una equivalencia en grados.
- **Límite superior.** – permite establecer un límite máximo de los valores tomados de la lectura de los pines analógicos del Arduino.

- **Equivalencia superior.** – A partir del límite superior genera una equivalencia máxima expresada en grados.

Al mapear el algoritmo generara una equivalencia en grados de las lecturas digitales realizadas, con esto acondiciona los datos de movimiento del brazo robótico de control eliminando lecturas que exceden los rangos de movimiento que pueda soportar el brazo robótico master ubicado sobre la Estación Remota.

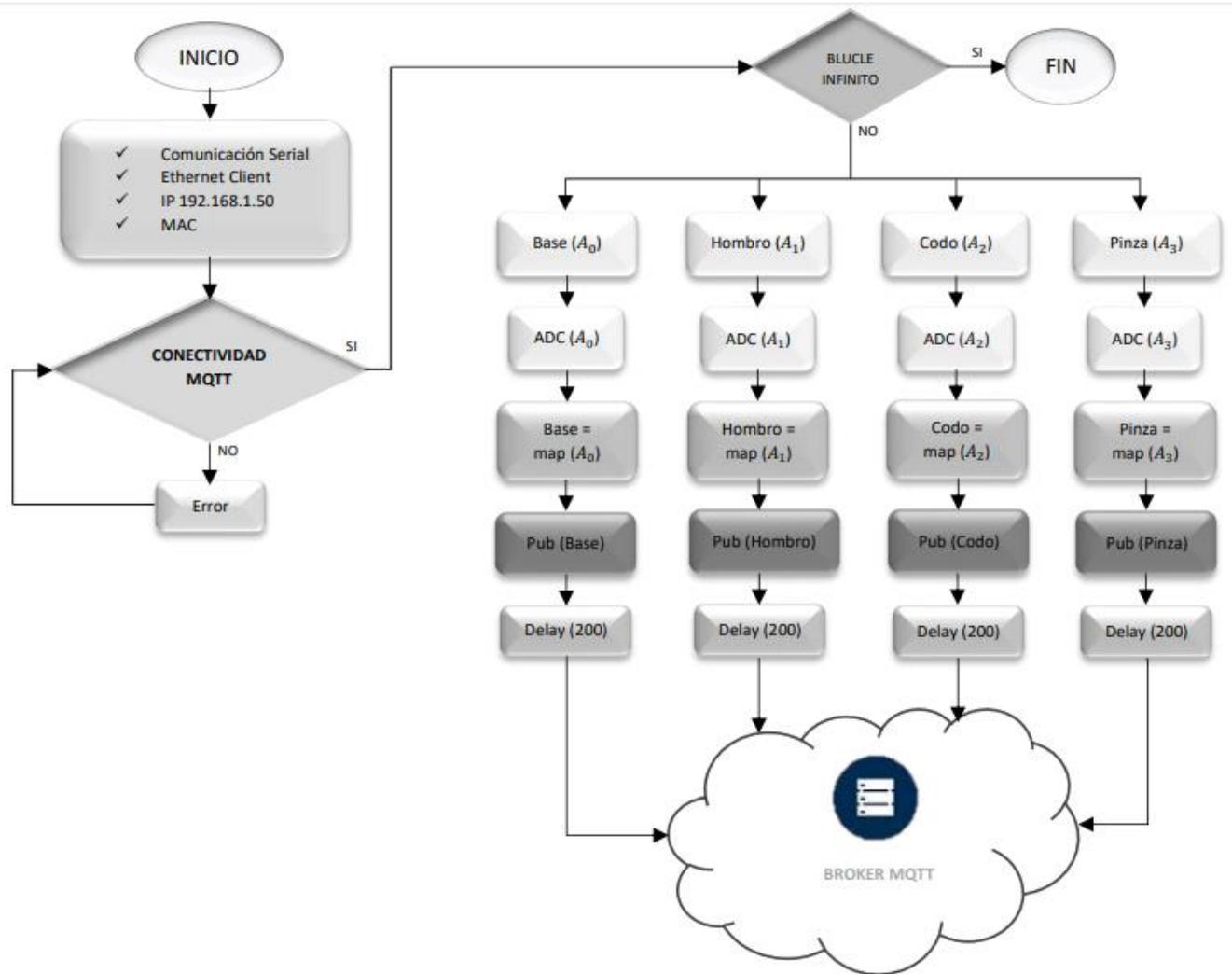
**Tabla 3.** Equivalencias máximas y mínimas de valores de lectura de movimiento expresada en grados.

*Fuente:* Autores

MAPEO DE LAS LECTURAS DE MOVIMIENTO DEL BRAZO ROBÓTICO DE CONTROL		
ENTRADA	E. MINIMA	E. MAXIMA
Base	0°	180°
Hombro	0°	70°
Codo	0°	60°
Actuador	0°	45°

Finalmente el algoritmo se encargara de enviar y publicar los datos ya procesados hacia cada uno de los subtópicos del bróker MQTT, para esto utiliza la instrucción “*client.publish(N\_topico,data)*”

- **N\_topic.** - Se especifica de forma jerárquica el nombre de tópico y Subtópico sobre el cual se desea realizar la publicación.
- **Data.** - Es el dato que va a ser publicado sobre el bróker MQTT.



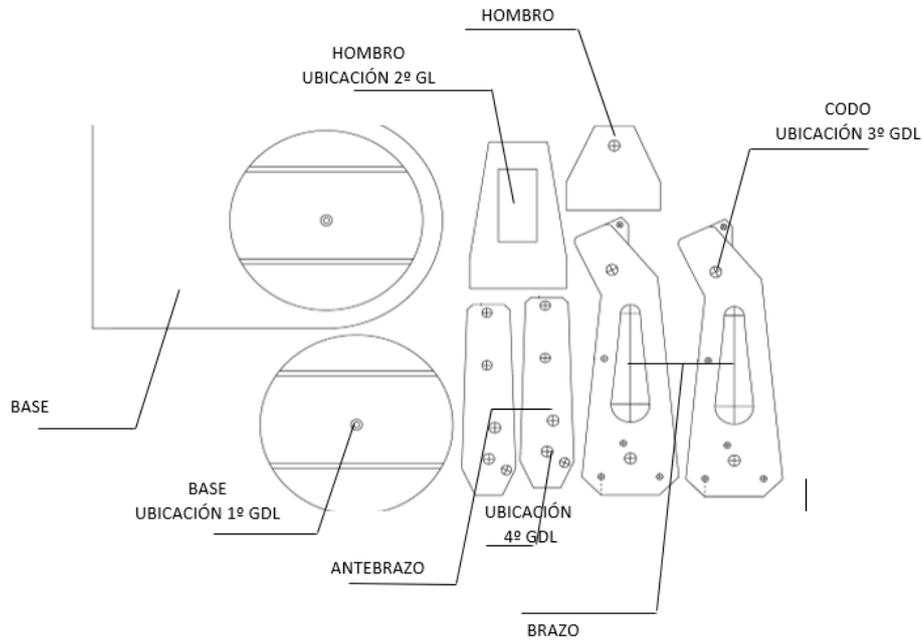
*Ilustración 14. Diagrama del algoritmo de adquisición, acondicionamiento y envío de datos.*

*Fuente: Autores*

### 3.8.2.2 UNIDAD ROBÓTICA (BRAZO ROBÓTICO DE CONTROL)

#### 3.8.2.2.1 DISEÑO DE LOS COMPONENTES DE LA UNIDAD ROBÓTICA

El diseño del brazo robótico master se realizó a partir de un boceto ubicado en la red publicado por un el autor (fortzero, 2019), del cual se modificó y reestructuro el diseño de forma que se acoplaran a las necesidades del proyecto, con el fin de obtener nuevas piezas para la construcción del prototipo del Brazo Robótico de control como se muestra en la siguiente ilustración.

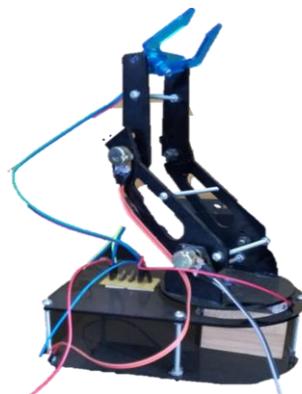


*Ilustración 15. Componentes del Brazo Robótico de Control.*

*Fuente: Autores*

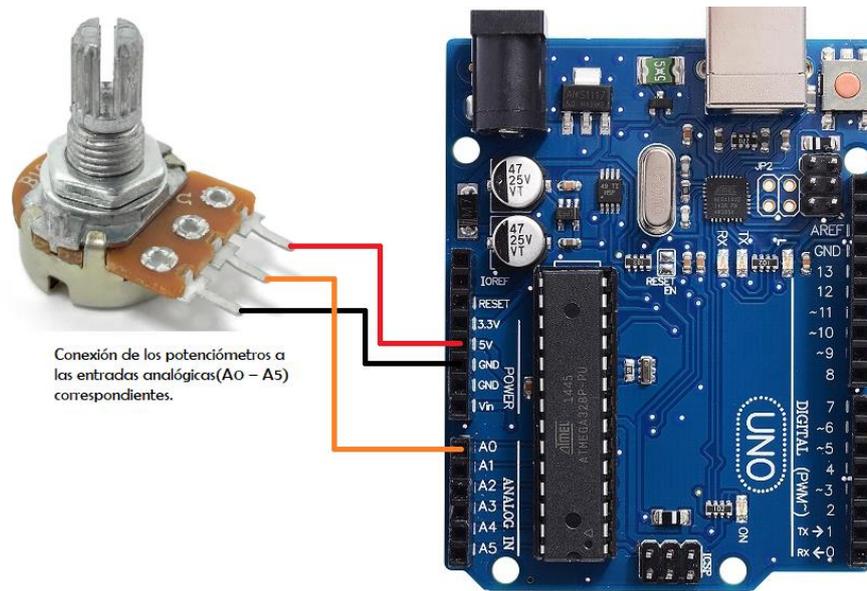
### 3.8.2.2.2 MONTAJE DE LA UNIDAD ROBÓTICA

Una vez diseñados los componentes se realiza la fabricación sobre acrílico negro mediante una cortadora Laser para obtener cada una de las piezas que conforman el prototipo de brazo robótico de control. Para el montaje del brazo de control se unen las piezas mediante la utilización de los potenciómetros, los cuales van a estar ubicados en cada grado de libertad que conforma el brazo.



*Ilustración 16. Unidad Robótica o Brazo Robótico de Control*

*Fuente: Autores*

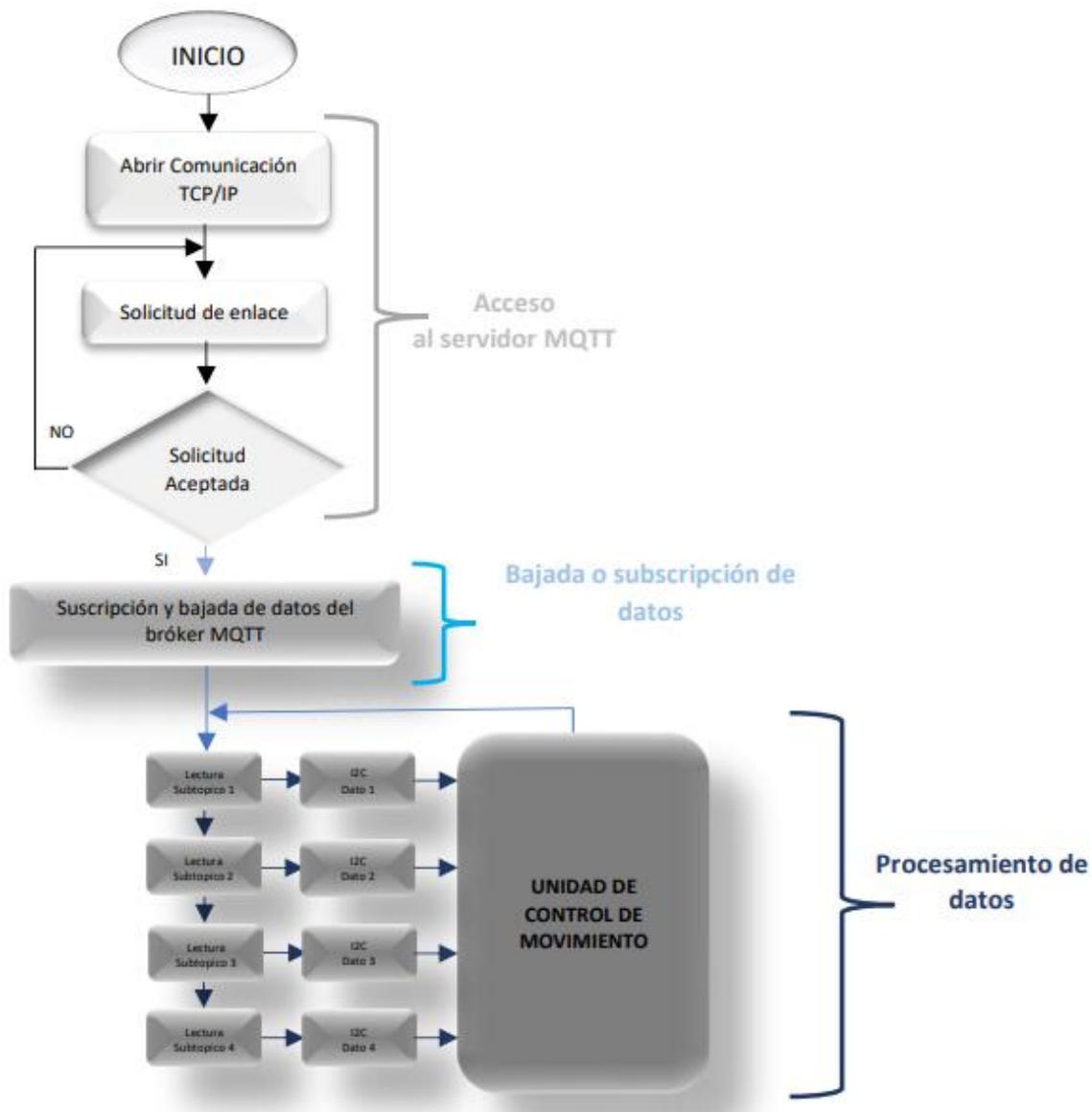


*Ilustración 17. Diagrama esquemático de la Estación Local.*

*Fuente: Autores*

### 3.8.3 ESTACIÓN REMOTA

#### 3.8.3.1 ALGORITMO DE BAJADA Y PROCESAMIENTO DE DATOS



*Ilustración 18. Diagrama general del algoritmo de baja y procesamiento de datos.*

*Fuente: Autores*

El algoritmo de bajada y procesamiento de datos, consta de acceso al servidor al MQTT, proceso el cual se detalla en el apartado anterior véase ilustración 18.

##### 3.8.3.1.1 SUBSCRIPCIÓN O BAJADA DE DATOS DE BRÓKER MQTT

Una vez establecida el acceso al bróker MQTT, el algoritmo debe generar una suscripción sobre cada uno de los tópicos y subtópicos determinados para este proyecto (véase la tabla 1). Al realizarse la suscripción se establece una comunicación en tiempo real entre la Estación

Local y la Estación Remota, teniendo acceso de los datos que son publicados sobre cada uno de los subtópicos del bróker, finalmente esta etapa del algoritmo toma los datos del subtópico conforme son escritos y los lleva hacia la Estación Remota con la ayuda de la siguiente instrucción “*client.subscribe(N\_Tópico)*”

- ***N\_Tópico.*** – Se define de forma jerárquica la dirección de los tópicos y subtópicos a los cuales se desea acceder.

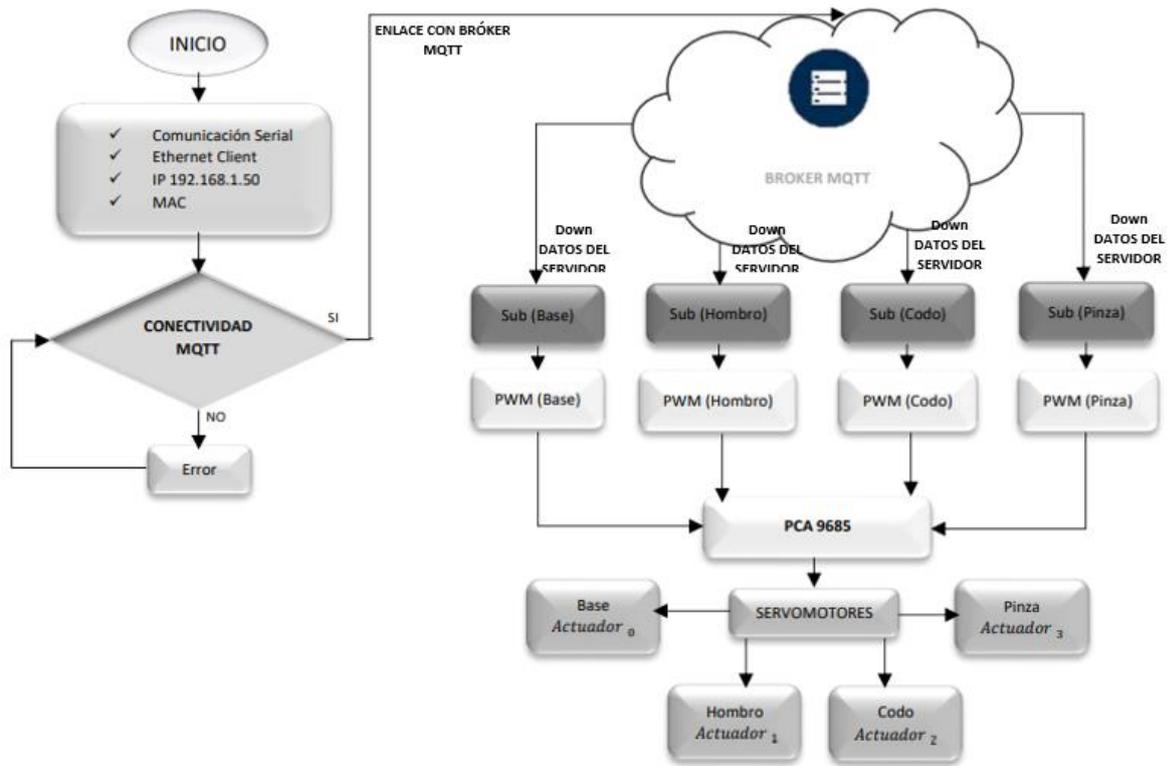
### 3.8.3.1.2 PROCESAMIENTO DE DATOS DE BRÓKER MQTT

Teniendo ya los datos sobre la Estación Remota, la última etapa del algoritmo realiza el procesamiento de los mismo, esto lo hace distribuyendo los datos según su subtópico hacia cada uno de los actuadores que forman parte de la Unidad Robótica master a través de la unidad de control de movimiento con la que cuenta la Estación Remota.



*Ilustración 19. Diagrama general del procesamiento de los datos obtenidos del bróker MQTT.*

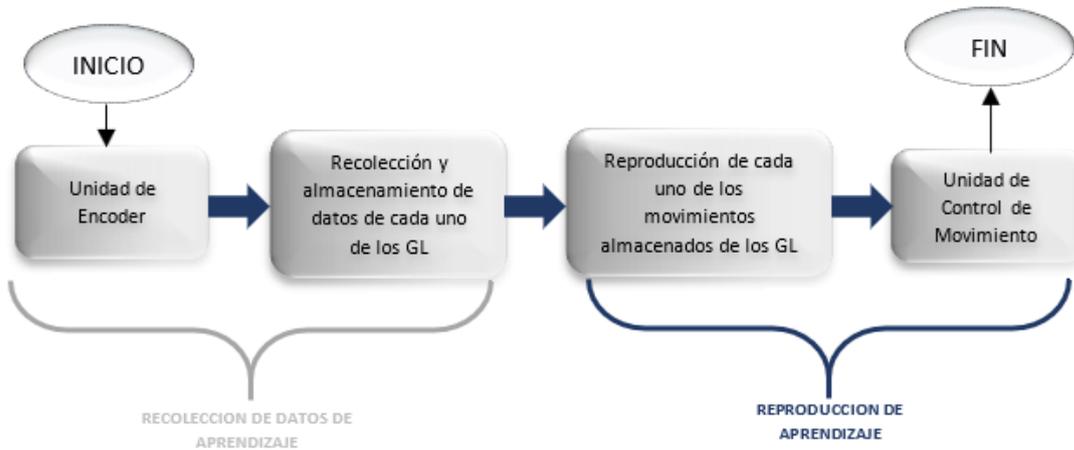
*Fuente: Autores*



*Ilustración 20.* Diagrama general de funcionamiento del algoritmo de suscripción y procesamiento de datos.

*Fuente:* Autores

### 3.8.3.2 ALGORITMO DE APRENDIZAJE POR DEMOSTRACIÓN



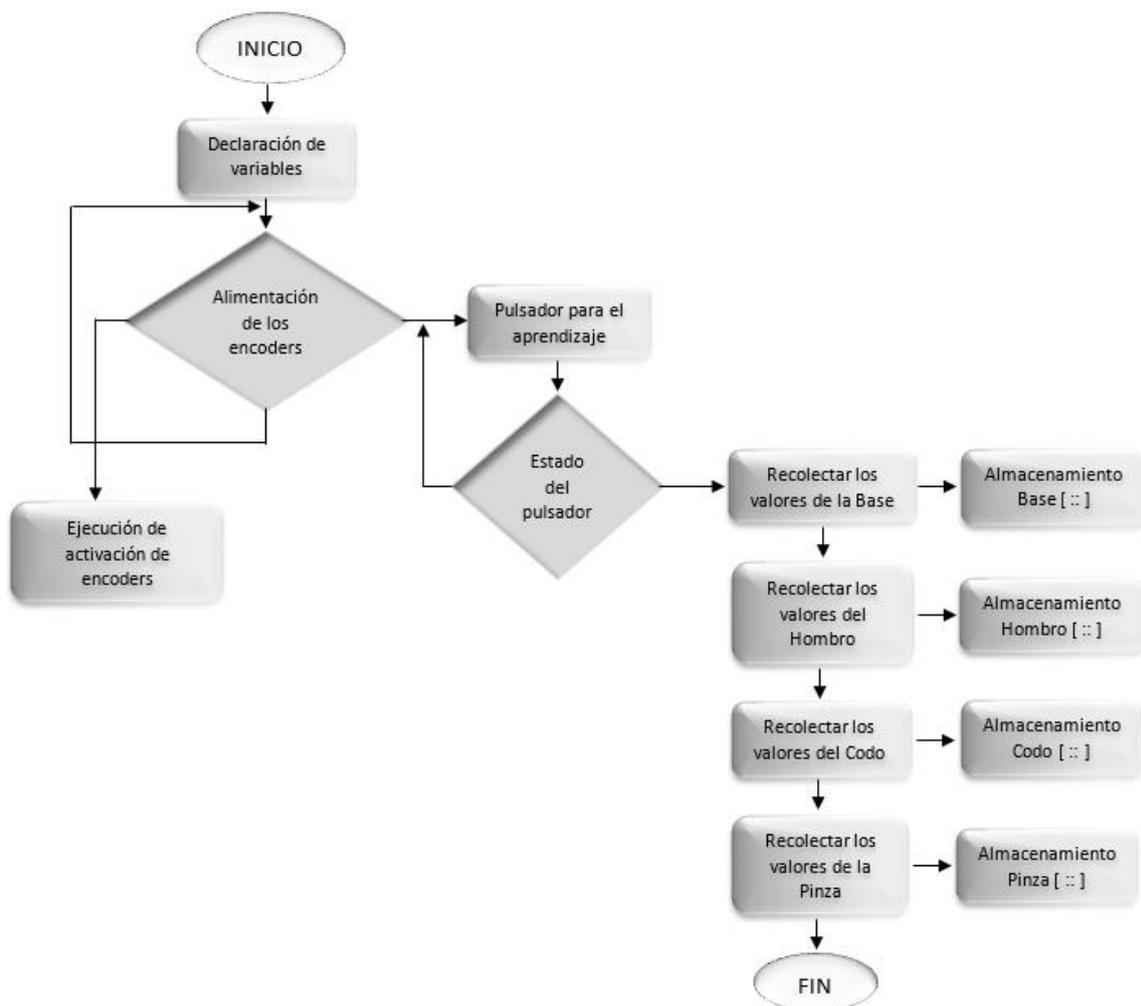
*Ilustración 21.* Diagrama general del algoritmo de Aprendizaje Por Demostración.

*Fuente:* Autores

### 3.8.3.2.1 RECOLECCIÓN DE DATOS DEL APRENDIZAJE

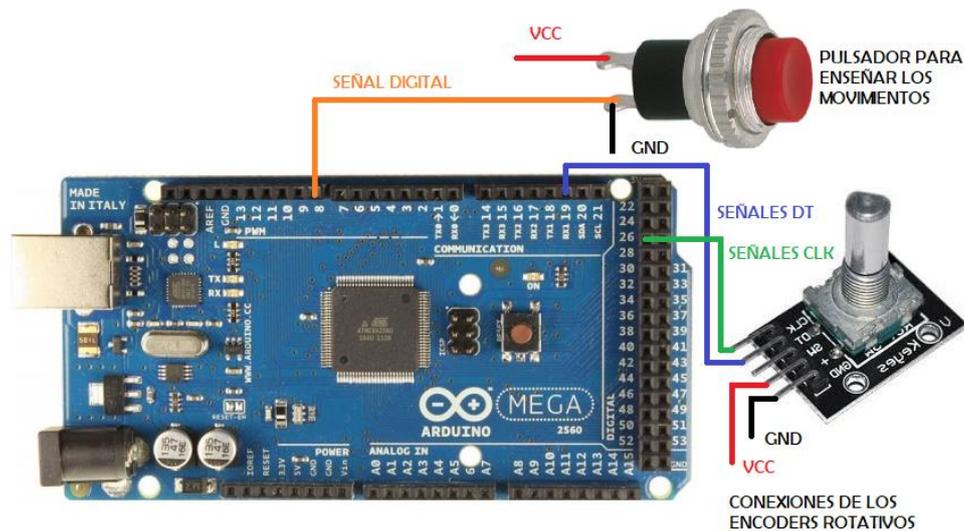
Para que el algoritmo pueda comenzar su funcionamiento es necesario que el usuario debe cumplir el prerequisite de petición de “enseñanza de movimiento” a través de un botón ubicado sobre la unidad robótica.

A continuación, los sensores de movimientos (unidades de encoders) son alimentados, permitiéndole a la etapa de recolección establecer peticiones de datos de lectura, esto se logra a través de los pines digitales a los cuales cada uno de los encoders se encuentra realizando una conexión física. Al recibir los datos proporcionados, el algoritmo continúa el procesamiento de dichos datos al almacenarlos sobre vectores de datos tal y como se muestra en la siguiente ilustración.



*Ilustración 22. Diagrama de la recolección y almacenamiento de datos.*

**Fuente:** Autores



*Ilustración 23. Diagrama de conexiones de la etapa de recolección de datos de aprendizaje.*

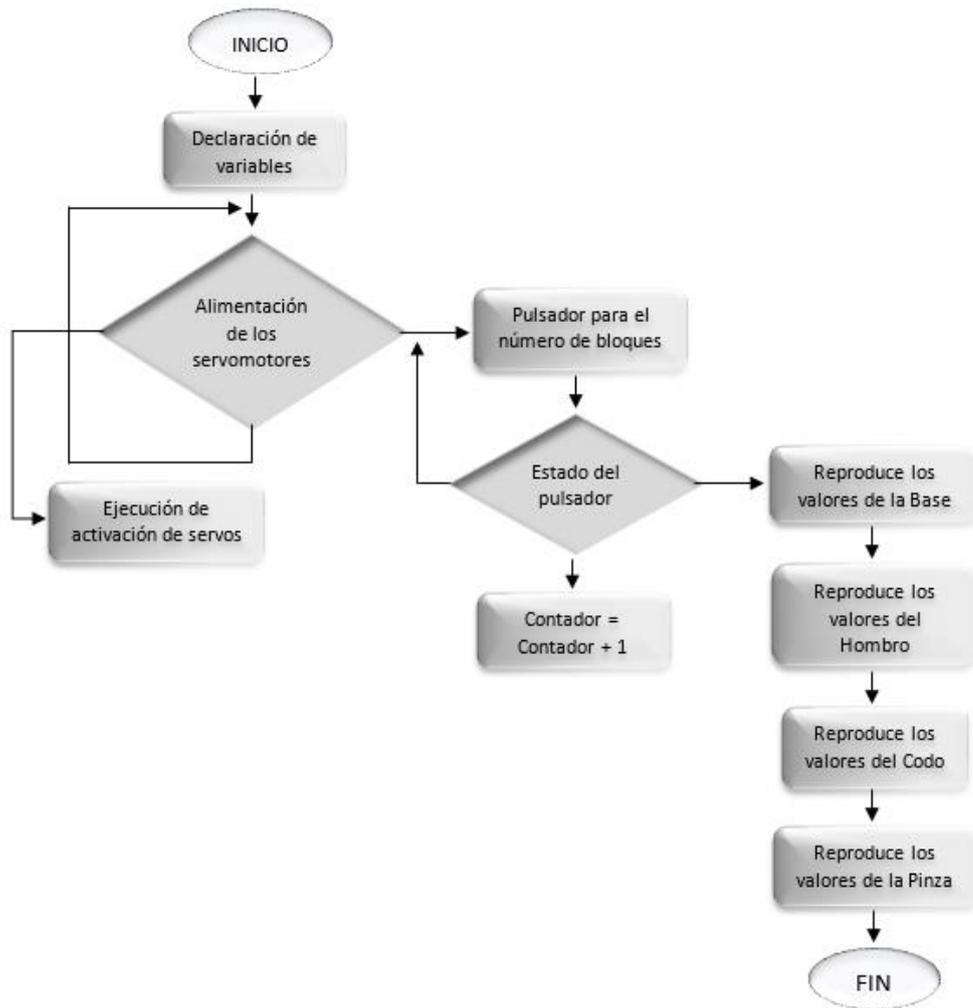
*Fuente: Autores*

### 3.8.3.2.2 REPRODUCCIÓN DE DATOS DEL APRENDIZAJE

Inicialmente se deben cumplir los siguientes prerequisites para que el algoritmo puede realizar sus funciones:

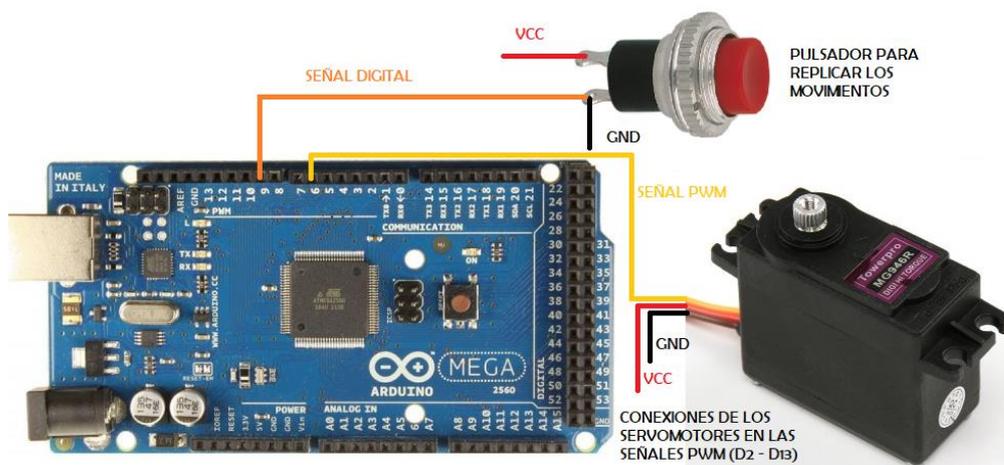
- Contar con datos de una enseñanza previa.
- El usuario debe presionar el botón de “*replica de movimientos*”

A continuación, la etapa de reproducción detecta la petición del usuario, de esta forma el algoritmo accede a cada uno de los vectores de datos de cada uno de los grados de libertad, seguido a esto Arduino enviará cada uno de estos datos hacia la unidad de control de movimiento quien se encargará de reproducir el aprendizaje como se muestra en la ilustración siguiente.



*Ilustración 24. Diagrama de general de la reproducción de datos de aprendizaje.*

*Fuente: Autores*



*Ilustración 25. Diagrama de conexiones de la etapa de reproducción de movimientos del aprendizaje.*

*Fuente: Autores*

### 3.8.3.3 UNIDAD ROBÓTICA (BRAZO ROBÓTICO MASTER)



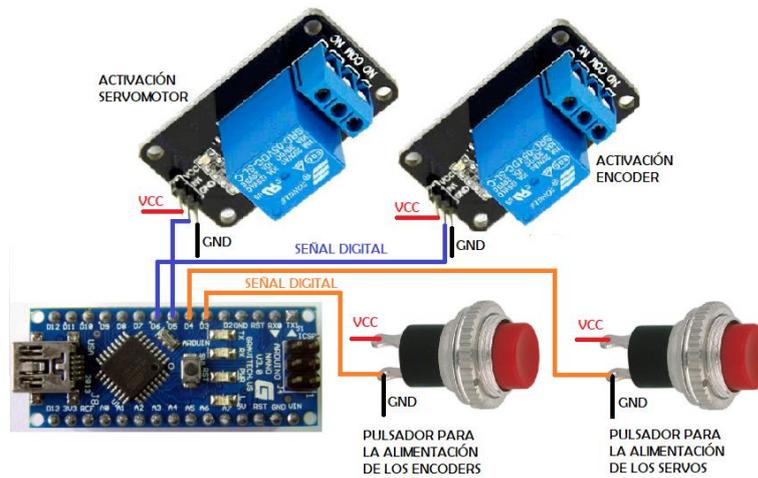
*Ilustración 26. Unidad robótica*

*Fuente: Autores*

#### 3.8.3.3.1. UNIDAD DE ALIMENTACIÓN

La unidad de alimentación, realiza un control de sobre las fuentes de energía tanto de la unidad de encoder, como de la unidad de control de movimiento, esto permite que cuando:

- El usuario al presionar el botón “enseñanza de movimiento” la unidad de encoders sea alimentada y entre en funcionamiento, desenergizando así la unidad de control de movimiento permitiendo al usuario realizar la manipulación del brazo robótico con libertad.
- El usuario al presionar el botón “reproducción de la enseñanza” la unidad de control de movimiento sea alimentada y entre en operación para poder reproducir los movimientos, además de la deshabilitación de la unidad encoders.

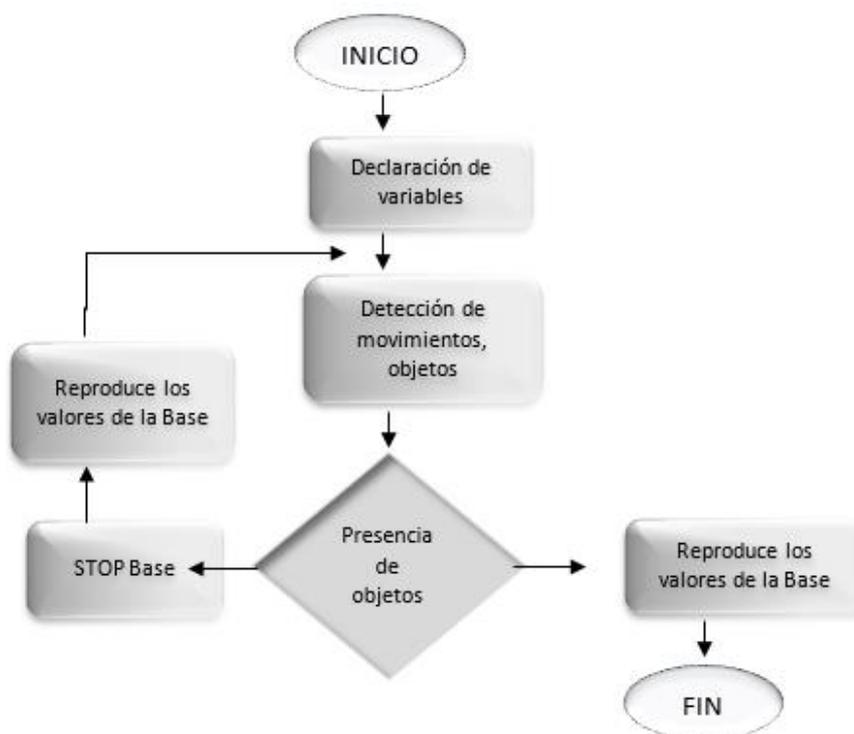


*Ilustración 27. Diagrama de conexiones de la unidad de alimentación.*

*Fuente: Autores*

### 3.8.3.3.2. UNIDAD DE SENSORES

La unidad de sensores permite la detección de obstáculos que el ambiente de trabajo del brazo robótico pueda presentar, para ello utiliza sensores infrarrojos ubicados tanto en la parte derecha como en la parte izquierda del mismo. En caso que el sensor detecte un objeto cercano de la unidad robótica esté genera un estado de parada de todo el sistema hasta cuando el obstáculo sea retirado.

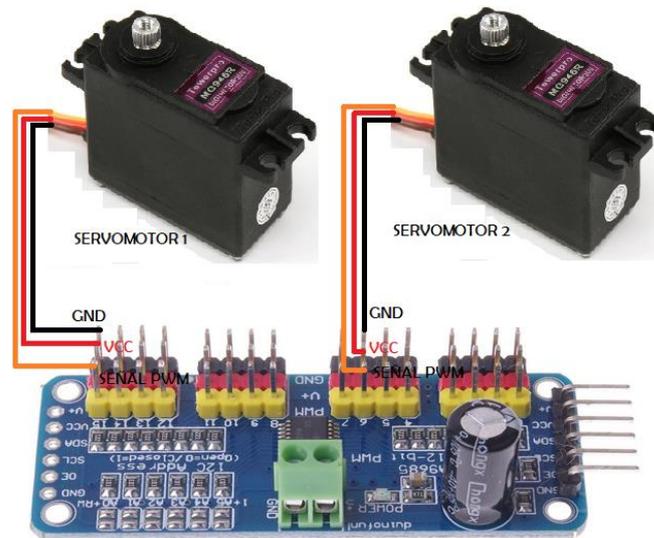


*Ilustración 28. Diagrama general de la unidad de sensores (Detección de obstáculos).*

*Fuente: Autores*

### 3.8.3.3.3. UNIDAD DE CONTROL DE MOVIMIENTO

La unidad de control de movimiento se encuentra conformada por una PCA 9685, esta tarjeta electrónica da alimentación a los servomotores y proporciona la señal PWM para el control del sistema de teleoperación.



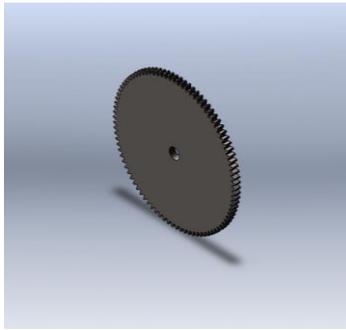
*Ilustración 29.* Diagrama de conexión de la unidad de control de movimiento y actuadores(servomotores).

*Fuente:* Autores

### 3.8.3.3.4. BRAZO ROBÓTICO

#### 3.8.3.3.4.1. DISEÑO DE LOS COMPONENTES DEL BRAZO ROBÓTICO

El diseño del brazo robótico master se realizó a partir de un boceto ubicado en la red publicado por un el autor (fortzero, 2019), del cual se modificó y reestructuro el diseño de forma que se acoplaran a las necesidades del proyecto, con el fin de obtener nuevas piezas para la construcción del prototipo del Brazo Robótico de control como se muestra en la siguiente (ilustración 32). Además, se realizó el diseño de engranes y piñones que serán colocados sobre las unidades de encoders y los grados de libertad del brazo robótico, con el fin de ajustar la resolución de movimiento del brazo robótico como se muestran en las siguientes ilustraciones.



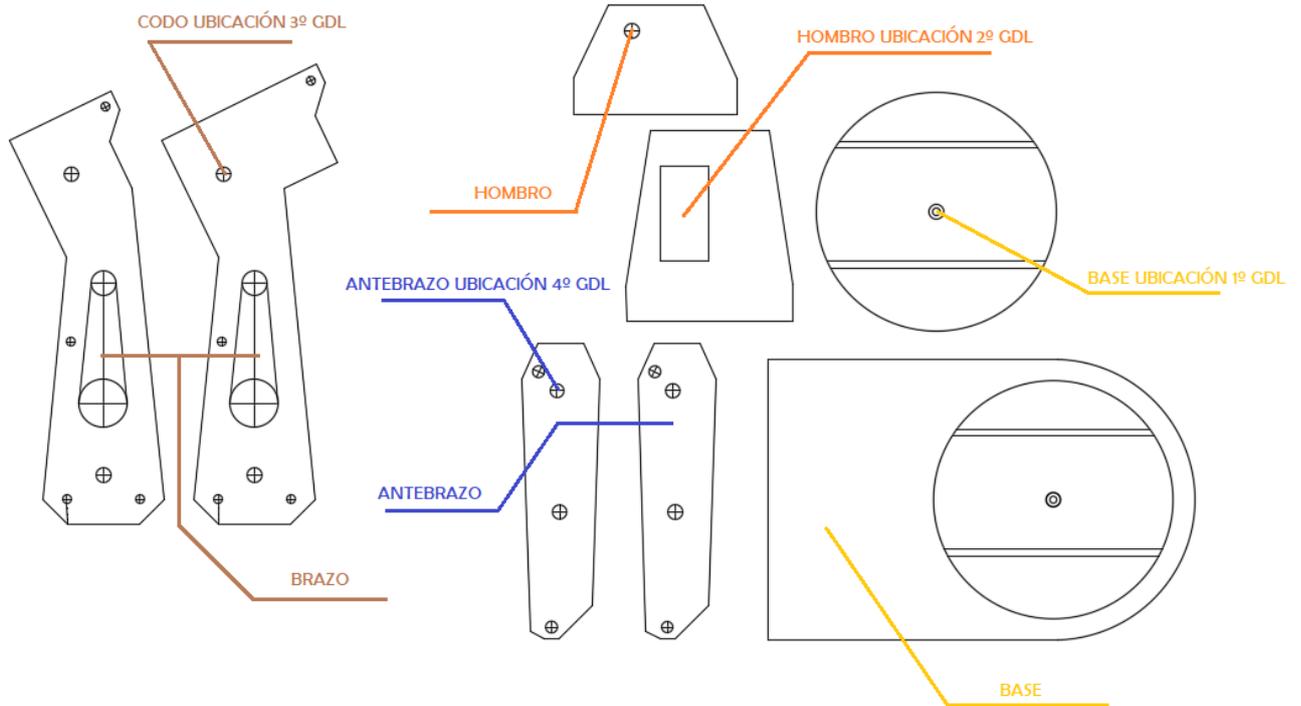
**Ilustración 30.** Engrane realizado en SOLIDWORK

**Fuente:** Autores



**Ilustración 31.** Piñón realizado en SOLIDWORK

**Fuente:** Autores



**Ilustración 32.** Componentes del Brazo Robótico de Master.

**Fuente:** Autores

### 3.8.3.3.4.2. DESCRIPCIÓN CINEMÁTICA DEL BRAZO ROBÓTICO

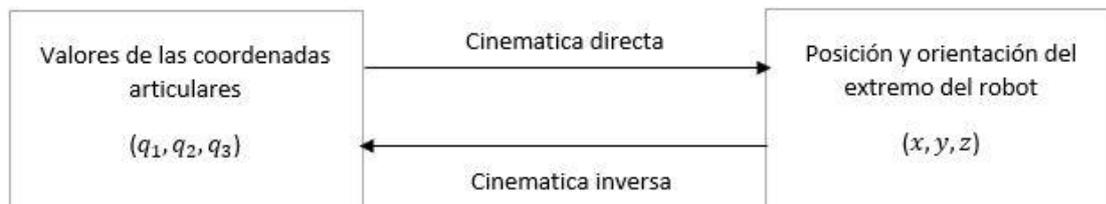
La estructura mecánica tiene una cierta similitud con la anatomía del brazo de un ser humano, entre las articulaciones existe un movimiento explícito de rotación. Con los servomotores utilizados se delimitaron para dichos movimientos una rotación en cada grado de libertad como se muestra en la tabla 4.

*Tabla 4. Resolución de GDL del brazo robótico.*

*Fuente: Autores*

GDL	MOVIMIENTOS	ROTACIÓN
1	BASE (Horario - Antihorario)	0° - 180°
2	HOMBRO (Arriba - Abajo)	20° - 120°
3	CODO (Arriba - Abajo)	20° - 60°

Con respecto al sistema de referencia situado en la base, se realiza un análisis del movimiento del brazo robótico utilizando la cinemática inversa, la cual determina la configuración que debe poseer el brazo para la posición y orientación del extremo conocidas, (véase ilustración 31).



*Ilustración 33. Diagrama de relación entre cinemática inversa y directa*

*Fuente: (Rodríguez J.C, 2016)*

En el modelo cinemático del brazo robótico se utilizó el esquema de Devanit – Hartenberg para su representación, debido a que esta relaciona la estructura cinemática de cada uno de los GDL, tomando un sistema de referencia por cada articulación (Weber y Darmstadt, 2010).

Al relacionar un sistema anterior ortogonal con un actual, se puede obtener la representación de una matriz de conversión genérica la cual se presenta en la Ecuación 1.

**Ecuación 1. Ecuación DH**

**Fuente:** (Buitrago G, 2015)

$$A_i^{i-1} = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i S\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

En la siguiente Ecuación, se referencia cada sistema con la base del brazo robótico, donde  $n$  es la cantidad de sistemas ortogonales del brazo robótico master (Abdel-Makel y Othman, 1999).

**Ecuación 2. Sistema de referencia ortogonales**

**Fuente:** (Buitrago G, 2015)

$$A_n^0 = \prod_{i=1}^n A_i^{i-1}$$

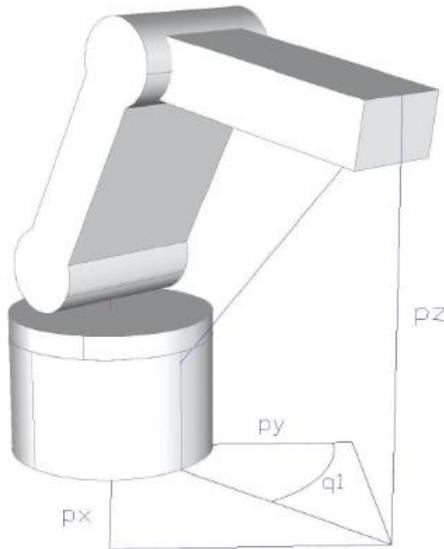
En la tabla 5, se describe de forma simbólica los parámetros con los que el brazo robótico master cuenta (articulaciones, GDL). El objetivo de obtener el modelo cinemático del brazo robótico master, es poder hallar la posición actual, para luego poder llevarla a una nueva posición.

**Tabla 5. Parámetros DH para el brazo robótico Master**

**Fuente:** Autores

Articulación	$\theta$	D	a	$\alpha$
1	q1	l1	0	0
2	90°	d2	0	90°
3	0°	d3	0	0

Después de estimar dicha posición, se halla la cinemática inversa del efector final, utilizando el método geométrico descrita por (Weber y Darmstadt, 2010), para ello, las ecuaciones se formulan tomando en consideración la (ilustración 31).



**Ilustración 34.** Diagrama esquemático de los primeros 3 GDL.

**Fuente:** (Rodríguez J.C, 2016)

Las ecuaciones formuladas relacionan las variables de las articulaciones con la posición actual y la estructura del brazo robótico Master. En la primera articulación el ángulo de rotación se obtiene de manera inmediata ya que representa la posición del brazo robótico como se muestra en la Ecuación 3, los valores de las articulaciones 2 y 3 son dependientes de las configuraciones que puedan adquirir estas (codo arriba o codo abajo). Esto se puede observar en la Ecuación 4

**Ecuación 3.** Primera articulación

**Fuente:** (Rodríguez J.C, 2016)

$$q_1 = \arctg\left(\frac{p_y}{p_x}\right)$$

**Ecuación 4.** Segunda y Tercera articulación

**Fuente:** (Rodríguez J.C, 2016)

$$q_2 = \arctg\left(\frac{p_z}{\pm\sqrt{p_x^2 + p_y^2}}\right) - \arctg\left(\frac{l_3 \sin(q_3)}{l_2 + l_3 \cos(q_3)}\right)$$

$$q_3 = \arctg\left(\frac{\pm\sqrt{1 - \cos^2(q_3)}}{\cos(q_3)}\right)$$

### 3.8.3.3.4.2. MONTAJE DE LA UNIDAD ROBÓTICA

Una vez diseñados los componentes se realiza la fabricación sobre acrílico negro mediante una cortadora Laser para obtener cada una de las piezas que conforman el prototipo de brazo robótico de control. Para el montaje del brazo de control se unen las piezas mediante la utilización de las unidades de encoders, los cuales van a estar ubicados en cada grado de libertad acoplados mediante los engranes y piñones.



*Ilustración 35. Brazo Robótico Master*

*Fuente: Autores*

## **CAPITULO IV**

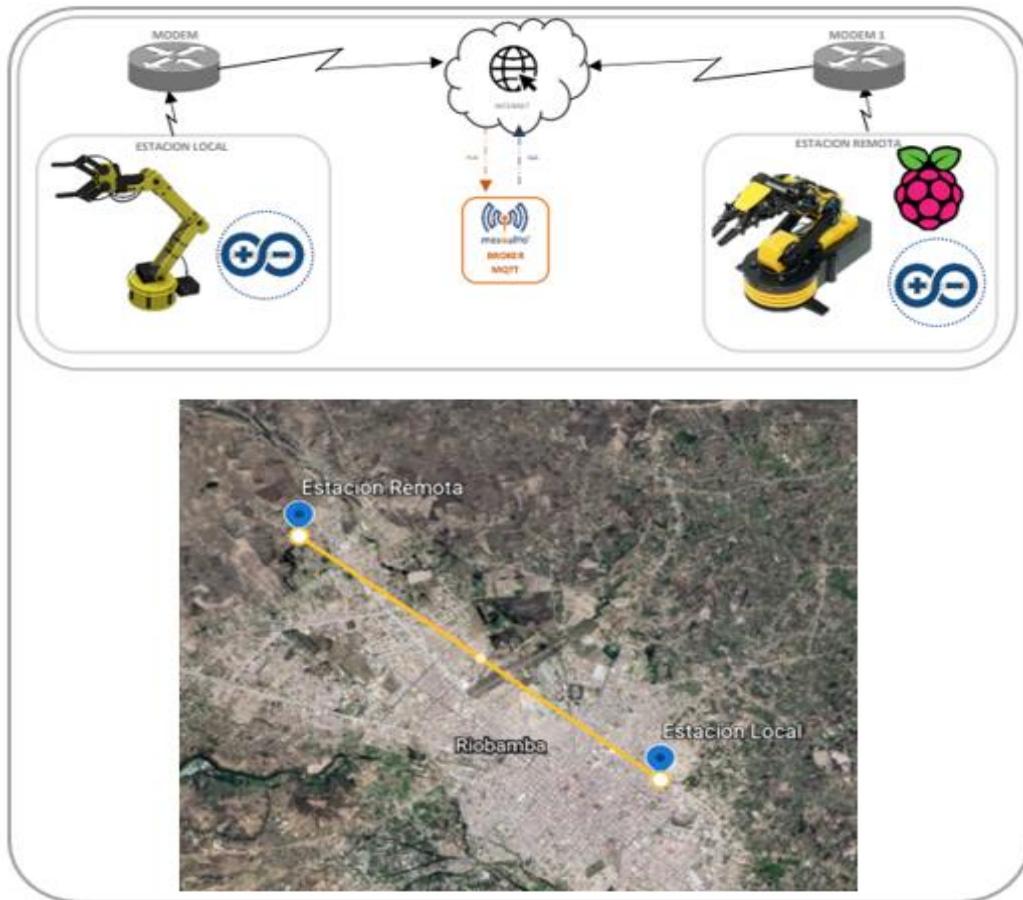
### **4. RESULTADOS Y DISCUSIÓN**

En este capítulo se presenta los diferentes resultados obtenidos al realizar las pruebas sobre cada uno de los sistemas que intervienen sobre las funciones de Teleoperación y Aprendizaje por Demostración.

#### **4.1 RESULTADOS EN EL SISTEMA DE TELEOPERACIÓN**

##### **4.1.1 PRUEBA DEL SISTEMA DE TELEOPERACIÓN AL VACÍO**

Uno de los parámetros principales tomados en cuenta en la Teleoperación es el poder establecer una comunicación entre la estación local y estación remota que permita la transmisión de los datos de forma íntegra para esto se realizaron pruebas de envíos de paquetes sobre cada uno de los tópicos con los que cuenta el Bróker MQTT desde la estación local, para posteriormente ser publicados sobre la estación remota, de esta forma poder observar si existe pérdida de paquetes de información, para esto se utilizó la herramienta MQTT Lens los resultados obtenidos se muestran en la tabla 6.



*Ilustración 36. Escenario para pruebas de Teleoperación.*

*Fuente: Autores*

La anterior figura muestra el escenario sobre el cual fue desarrolladas las pruebas de la teleoperación, esto con el fin de poder establecer además la latencia existente entre la estación local y remota hacia el servidor MQTT, para esto se utilizó comandos ping.

*Tabla 6. Comprobación del éxito de la comunicación con datos publicados y suscritos*

*Fuente: Autores*

PUBLICADOR				SUBSCRIPTOR				LATENCIA
Base	Hombro	Codo	Actuador	Base	Hombro	Codo	Actuador	
150	150	150	300	150	150	150	300	50 ms
200	200	200	350	200	200	200	350	48 ms
250	250	250	400	250	250	250	400	35 ms
300	300	300	450	300	300	300	450	75 ms
350	350	350	500	350	350	350	500	28 ms
400	400	400	550	400	400	400	550	48 ms

Los resultados obtenidos detallan que un total de 24 paquetes de información enviados, los 24 paquetes son entregados con éxito y 0 de paquetes son perdidos por el servidor, además se pudo visualizar que los datos tanto de publicación como de suscripción en cada uno de los subtópicos no presentan cambios, obteniendo como resultado una comunicación exitosa.

La latencia con la que cuenta el sistema al realizar las pruebas se puede observar que presenta un valor máximo de 75 milisegundos, los cuales son considerados relativamente bajos, esto basado en diferentes estudios realizados (Campero J, 2016) (Hervas C., 2018) (Requena, r, 2014) en donde se determina que los tiempos de latencia para una comunicación en aplicaciones de tiempo real excelente debe ser menor a 150 ms. Datos que son respaldados por MQTT.org.

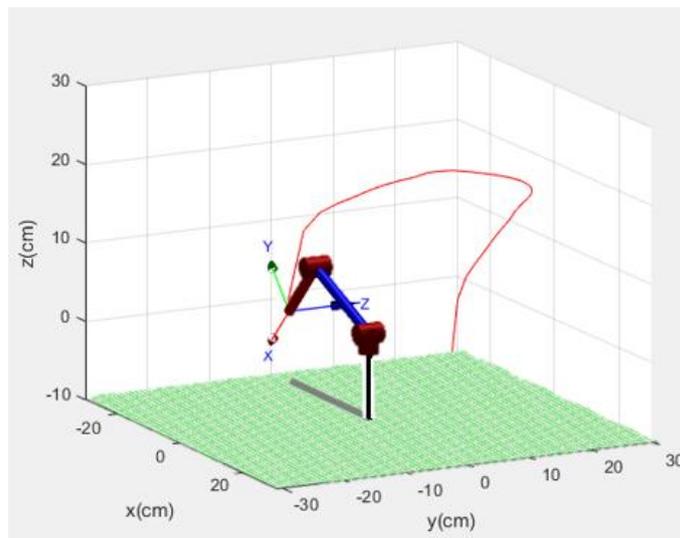
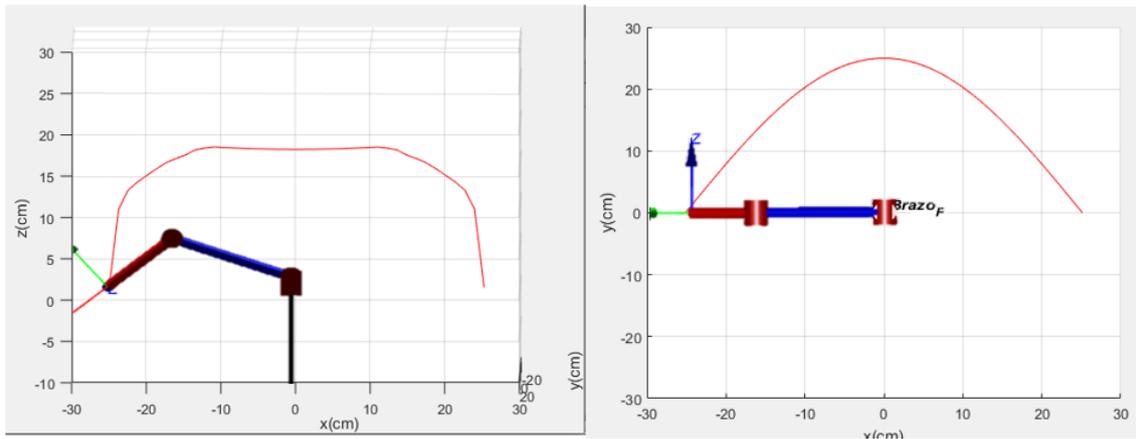
```
Símbolo del sistema - ping -t www.cloudmqtt.com
Haciendo ping a www.cloudmqtt.com [13.226.45.121] con 32 bytes de datos:
Respuesta desde 13.226.45.121: bytes=32 tiempo=71ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=71ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=99ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=132ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=164ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=99ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=128ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=72ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=71ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=99ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=72ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=71ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=72ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=95ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=71ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=74ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=74ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=74ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=75ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=71ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=103ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=71ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=73ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=76ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=88ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=107ms TTL=244
Respuesta desde 13.226.45.121: bytes=32 tiempo=110ms TTL=244
```

*Ilustración 37. Métodos de verificación de resultados*

*Fuente: Autores*

#### **4.1.2 PRUEBA DE EVALUACIÓN DE MOVIMIENTO DEL BRAZO ROBÓTICO**

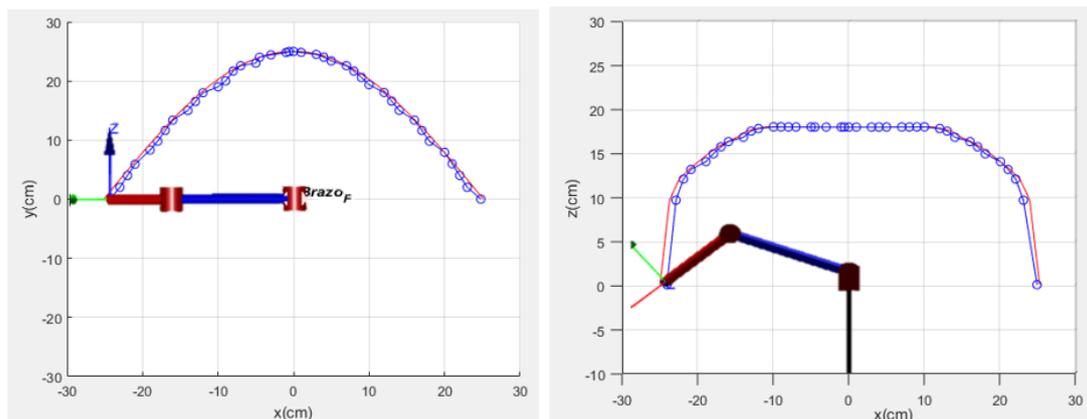
Una vez comprobado la comunicación entre la estación remota y la estación local, se realizó la prueba de manipulación del brazo de control ubicado en la estación local, con el fin que mediante el método de visualización directa poder evaluar el movimiento que es ejecutado por la unidad robótica ubicada en la estación remota.

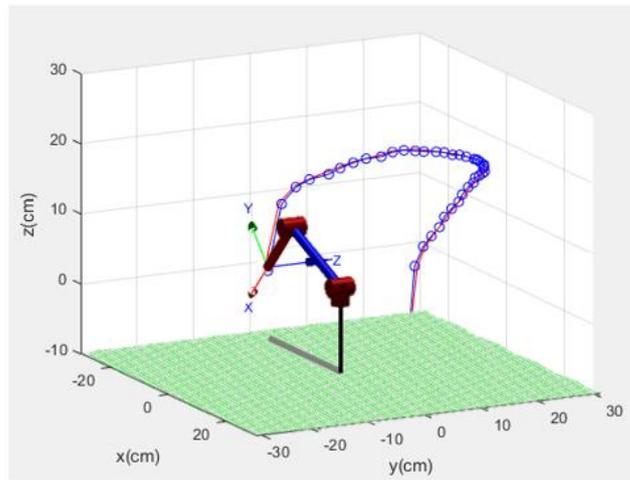


**Ilustración 38.** Trayectoria realizada por el usuario con el dispositivo de control en la estación local.

**Fuente:** Autores

Para el análisis de estos resultados se utilizó simulaciones, en donde se realiza una comparativa entre los movimientos en trayectoria realizados por el operador en la estación local y por la trayectoria replicada por el brazo robótico master ubicado en la estación remota.

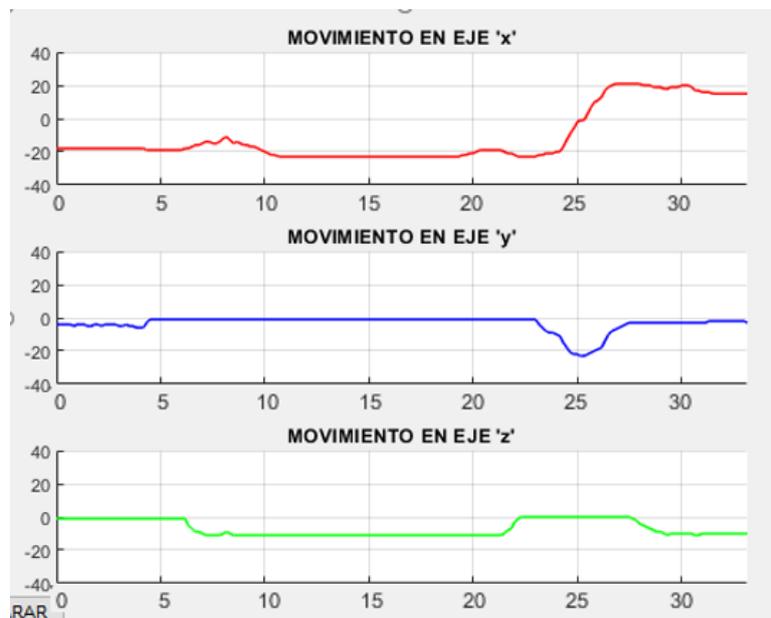
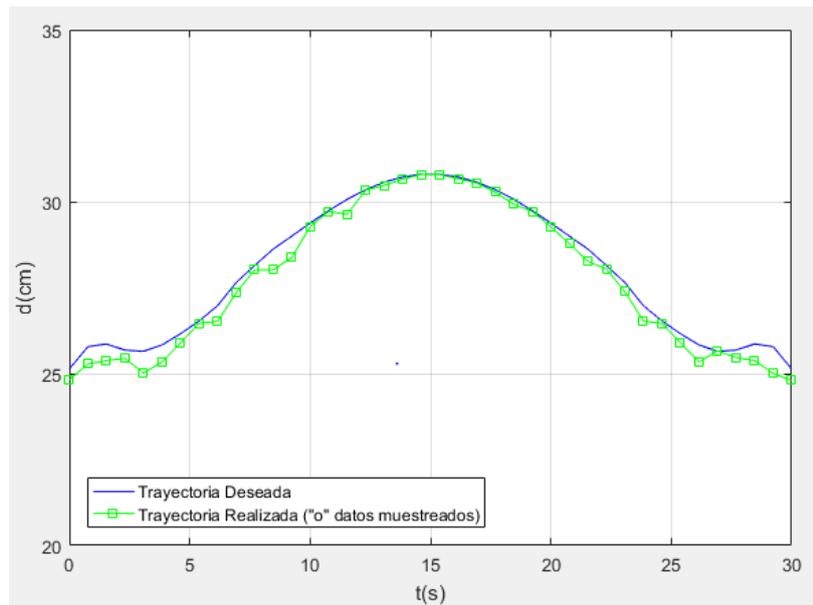




**Ilustración 39.** Trayectoria realizada por el brazo robótico en la estación remota.

**Fuente:** Autores

Los resultados obtenidos muestran que la trayectoria realizada por el brazo robótico tiene un margen de error, para poder obtener el margen de error de la trayectoria realizada se generó una comparativa entre la trayectoria deseada y la trayectoria realizada en función del tiempo y la distancia desde la base del brazo robótico hacia el efector final esta expresada en cm.



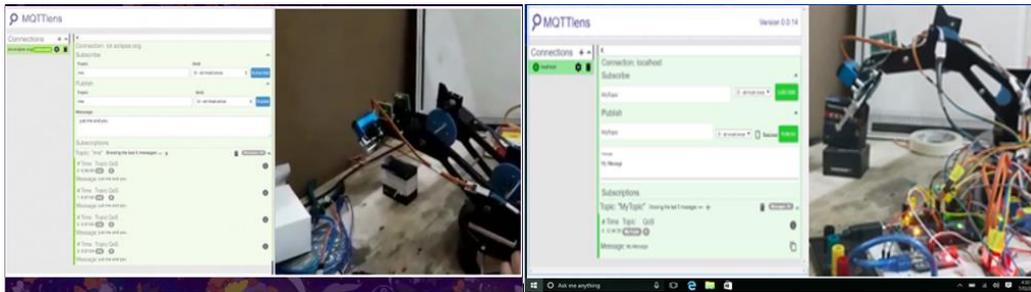
*Ilustración 40. Comparación de trayectoria deseada y trayectoria realizada.*

*Fuente: Autores*

Los datos obtenidos muestran que la distancia desde el centro de sistema hacia el efector final presenta un margen de error en la trayectoria, el cual es menor a 0.7 cm como máximo, esto debido a que los sensores utilizados en brazo de control en la estación local no cuentan con una alta resolución de lectura, se puede considerar este margen de error despreciable, obteniendo un resultado exitoso en la trayectoria de teleoperación del brazo robótico.

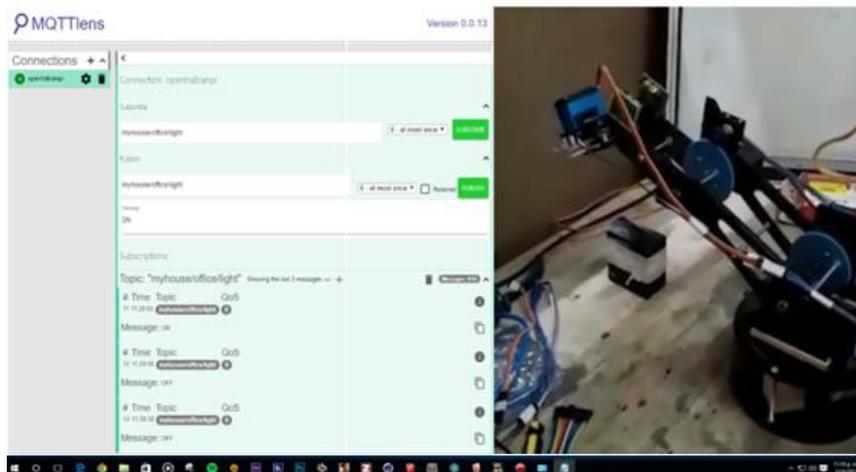
### 4.1.3 PRUEBAS DEL SISTEMA DE SUPERVISIÓN Y CONTROL

En la teleoperación es necesario contar con un sistema de supervisión y control de movimientos (Anexo 4), por este motivo se realizaron pruebas de conexión sobre el mismo para esto se utilizó el método de observación directa del comportamiento.



(a)

(b)



(c)

**Ilustración 41.** Pruebas de Supervisión y control (a) usuario control 1º GDL (b) usuario control 2º GDL (c) usuario control brazo completo

**Fuente:** Autores

De las pruebas realizadas se pudo obtener que el límite de usuarios para el control del brazo robótico es de 4, uno por cada GDL, teniendo éxito distribución del control del brazo robótico, el problema presentado en estas pruebas fue que si dos usuarios realizan una publicación sobre un mismo tópico los datos al mismo tiempo estos son alterados de forma que el control el brazo robótico master ubicado en la estación remota no es el esperado, como solución a este problema se definió que al momento de compartir control del brazo robótico master no se puede realizar publicación sobre un mismo tópico de parte de los usuarios. Así mismo de estas pruebas se pudo obtener que en las pruebas de supervisión el número de usuarios puede exceder los 4 usuarios esto debido a que al momento de realizar una suscripción a los tópicos imposibilitándoles el cambio de datos sobre los tópicos del bróker, obteniendo así un éxito en las pruebas de supervisión y control multiusuarios.

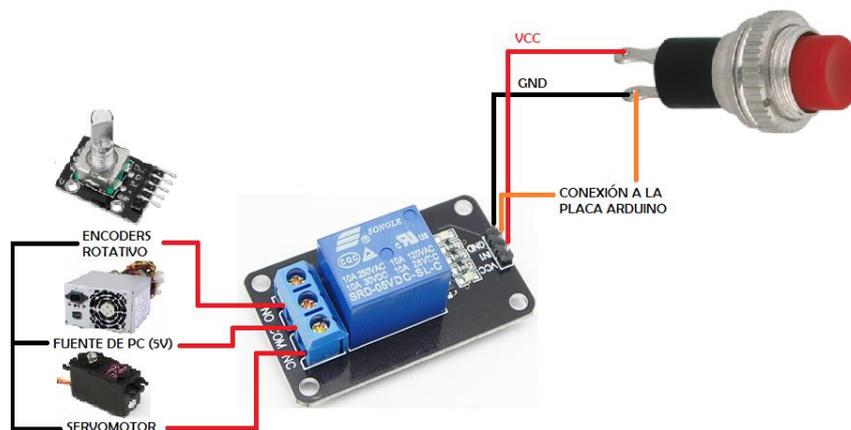
## 4.2 RESULTADOS OBTENIDOS EN EL APRENDIZAJE DEL BRAZO ROBÓTICO

En este apartado se analizan los resultados obtenidos en las diferentes etapas que intervienen como son la unidad de alimentación, la recolección de datos para capturar los movimientos de enseñanza, la reproducción de enseñanza y la detección de obstáculos para priorizar posibles colisiones generadas por algún obstáculo.

### 4.2.1 PRUEBAS DE LA ALIMENTACIÓN DEL SISTEMA

Para realizar el trabajo de aprendizaje directo, se tiene que, activar la alimentación en la unidad de encoders para su funcionamiento y deshabilitar la alimentación en la unidad de control de movimiento (servomotores), para poder generar los movimientos de enseñanza en el brazo, posterior a esto se debe deshabilitar la alimentación en los encoders con el fin de que no active una interrupción y activar la alimentación en los servos para la etapa de reproducción de la enseñanza.

Debido a eso se decidió utilizar un módulo relé, que por su configuración iba a controlar en su estado normalmente cerrado la alimentación del encoder y en la de normalmente abierto la alimentación del servo, que mediante la utilización de un pulsador este cambiaría su estado, para alimentar a la etapa de aprendizaje que corresponda dentro del funcionamiento (véase ilustración 42).



*Ilustración 42. Conexiones del módulo relé para alimentación de servomotores y encoders*

*Fuente: Autores*

El resultado que se obtuvo de esta conexión, es que no se logró deshabilitar la alimentación por completo del encoder, ya que el mismo se seguía alimentando, debido a un rebote de retroalimentación proporcionado por el PCA, el cual posee una alimentación con el Arduino para comunicación y la otra alimentación para los servomotores, esto ocasionaba que la

interrupción esté presente al estar energizado ya los servos y al momento de reproducir la enseñanza, el movimiento llamaba a la interrupción dentro del encoder provocando una colisión en todo el sistema.

Debido a lo anterior mencionado se optó por utilizar dos módulos relés para controlar la alimentación de forma independiente, con el fin de controlar el encendido y el apagado tanto en los encoders como en los servomotores de una forma no secuencial, se priorizo este cambio de estado a través de interrupciones por lo que se optó ocupar la etapa de alimentación aparte utilizando para esto un Arduino Nano y ya no dentro del Arduino Mega (véase ilustración 27).

Como resultado al apartado anterior, se obtuvo un manejo más práctico de la energización que va a necesitar el algoritmo de aprendizaje, ya que, al estar puesto en una interrupción, esta puede ser llamada al momento de presionar los pulsadores que corresponden a cada uno de los procesos que ejecuta el brazo robótico.

#### **4.2.2 PRUEBAS DE LA RECOLECCIÓN DE DATOS**

Para las pruebas de recolección de datos se tomó en consideración el uso del módulo encoder ya que al ser un módulo digital el cambio de estado podría ser controlado para obtener el posicionamiento que a través del sistema de engranajes y piñones se acoplarían al movimiento efectuado por el servomotor. Para ello se empleó una fórmula de incremento de posición para relacionar el número de vueltas con el ángulo de posicionamiento del encoder. La cual sería verificada en los 4 grados de libertad que posee el brazo robótico, siendo la siguiente.

$$POSICIÓN = POSICIÓN INICIAL + INCREMENTO DE LA POSICIÓN$$

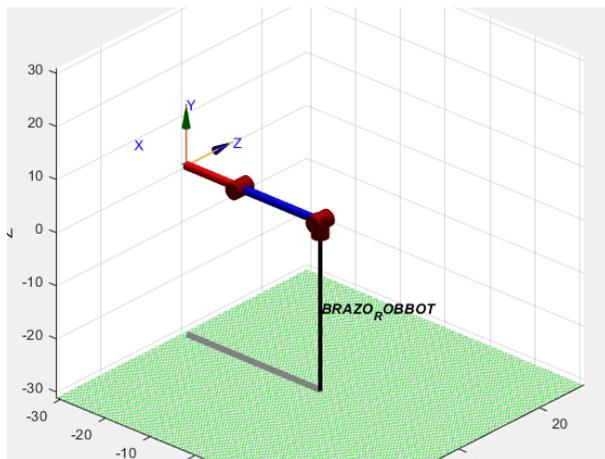
Tomando como referencia la primera prueba del primer incremento de la base se determinó un aumento de 10.8, el cual, al ser reflejados en los otros puntos no concordaba los movimientos con el posicionamiento del servo de cada grado de libertad, por lo que se hizo el análisis individual para cada uno de los puntos, obteniendo la siguiente tabla 7.

**Tabla 7.** Posicionamiento del encoder con el servomotor

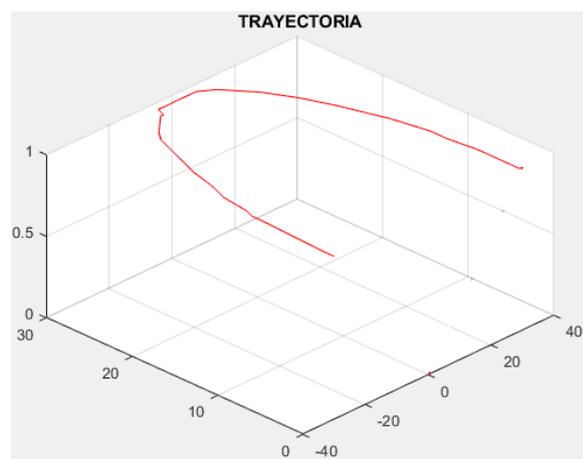
**Fuente:** Autores

FÓRMULA GENERAL			
POSICIÓN = POSICIÓN INICIAL + INCREMENTO DE LA POSICIÓN			
GRADOS DE LIBERTAD DEL BRAZO ROBÓTICO			
BASE	HOMBRO	CODO	PINZA
FÓRMULA	FÓRMULA	FÓRMULA	FÓRMULA
POSICIÓN = POSICIÓN INICIAL ± 10.8	POSICIÓN = POSICIÓN INICIAL ± 9	POSICIÓN = POSICIÓN INICIAL ± 5.4	POSICIÓN = POSICIÓN INICIAL ± 10.8

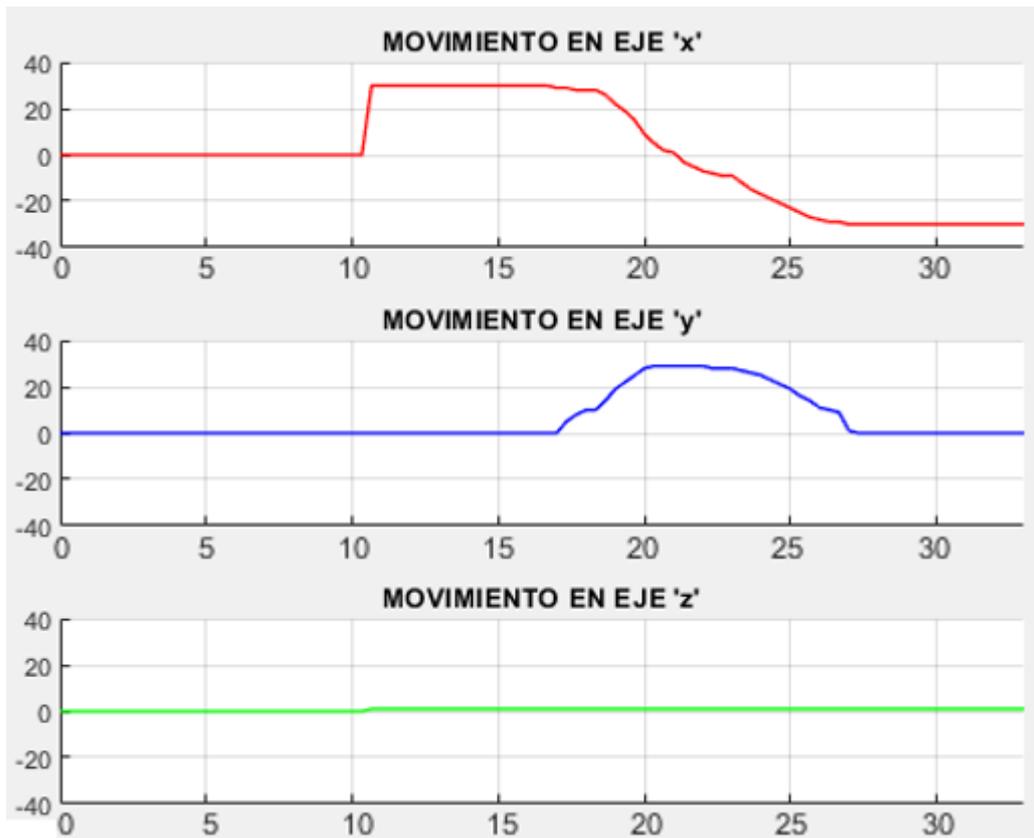
Para verificar el correcto funcionamiento del sistema de recolección de datos, tomando como base los valores propuestos en la tabla 7, que tienen como unidad de medida los centímetros (cm) y el tiempo en segundos(s), se realizaron las siguientes pruebas, que consisten en la recolección de forma individual, es decir, ir verificando los datos tomados por cada grado de libertad para esto se le asignaron trayectorias a cada uno de ellos como se muestra a continuación.



(a) Brazo robótico 1GDL (BASE)



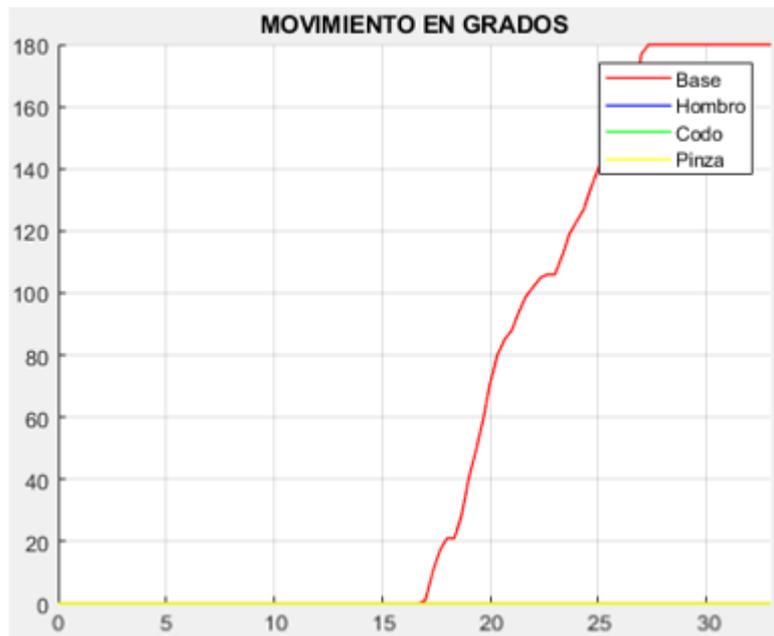
(b) Trayectoria deseada en el 1GDL (BASE)



(c) Trayectoria d(cm) y t(s) del 1GDL (BASE) representado en cada eje del plano

**Ilustración 43.** Trayectoria de los datos recolectados en la primera articulación BASE.

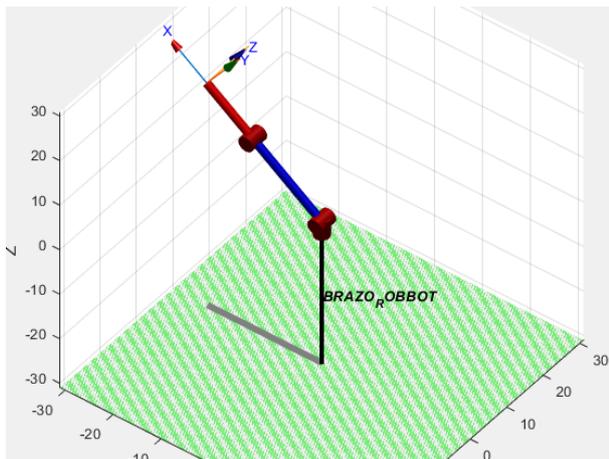
*Fuente:* Autores



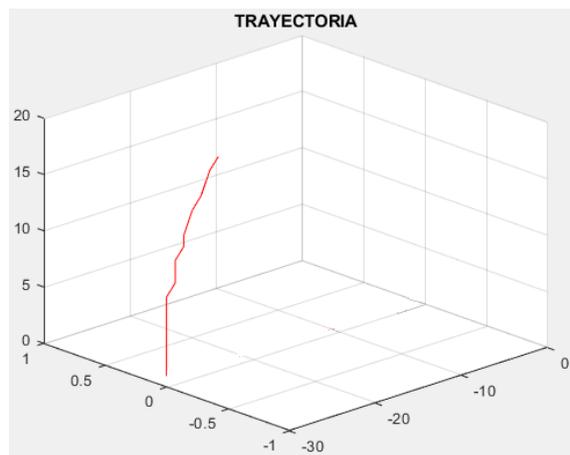
**Ilustración 44.** Trayectoria expresada en grados de los datos recolectados en la primera articulación BASE

*Fuente:* Autores

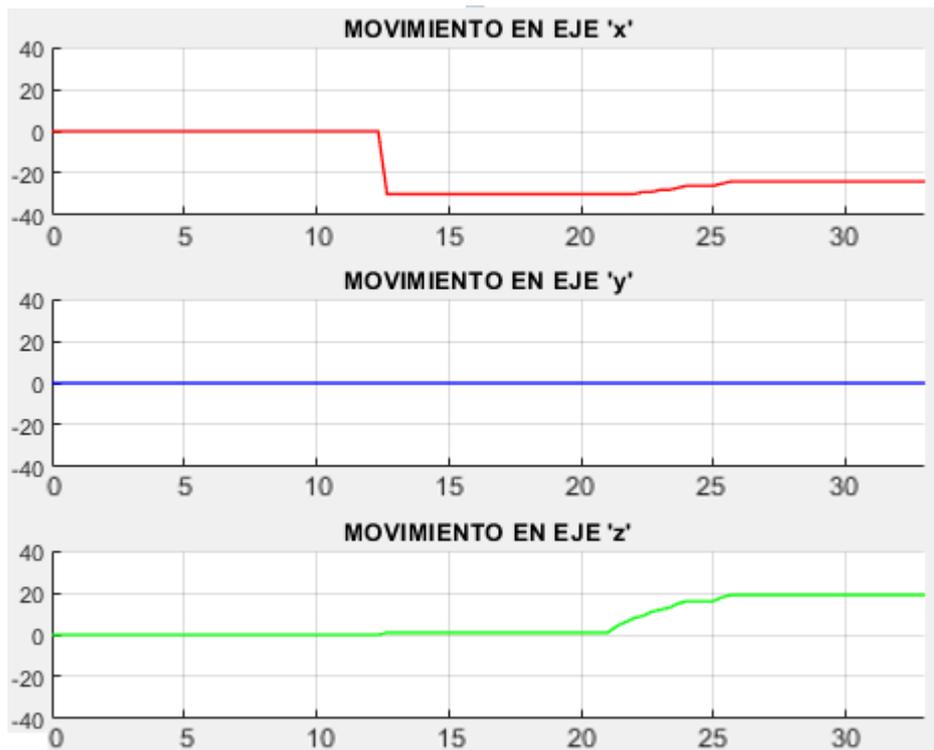
Los datos recolectados del primer grado de libertad (BASE) por tiempos de muestreo utilizando la cinemática inversa donde la distancia esta expresada en centímetros (cm) y el tiempo en segundos (s) (véase en la ilustración 43), según los ejes de referencia se visualiza que, en el eje X y en el eje Y la recolección tiene un mejor desempeño, y en el eje Z no tiene variación, la enseñanza del movimiento a realizar tiende a llegar a lo requerido por lo que es satisfactoria la tarea enseñada como se muestra en la (ilustración 44) donde su gráfica esta expresada en grados.



(a) Brazo robótico 2GDL (HOMBRO)



(b) Trayectoria deseada en el 2GDL (HOMBRO)



(c) Trayectoria del 2GDL (HOMBRO) representado en cada eje del plano

**Ilustración 45.** Trayectoria de los datos recolectados en la segunda articulación HOMBRO.

Fuente: Autores

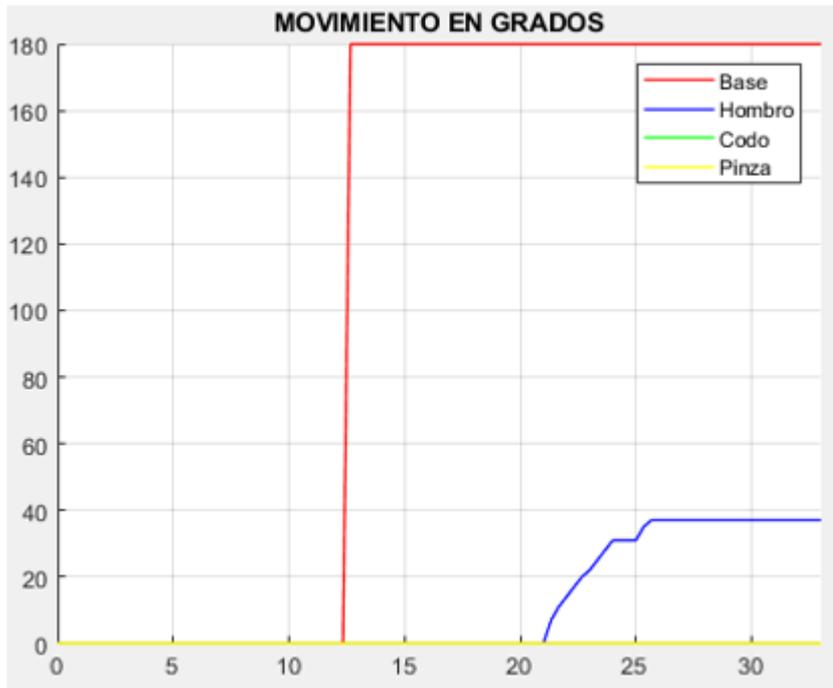
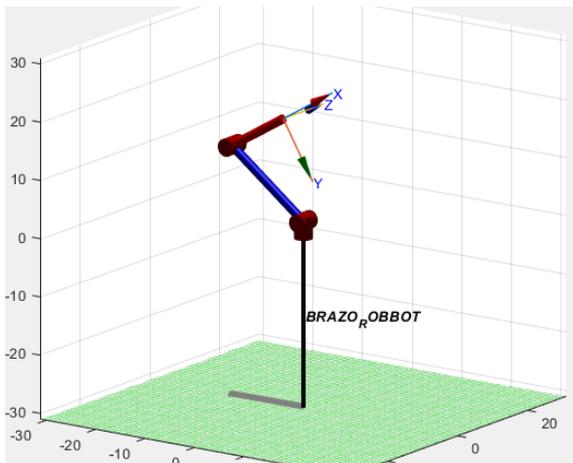


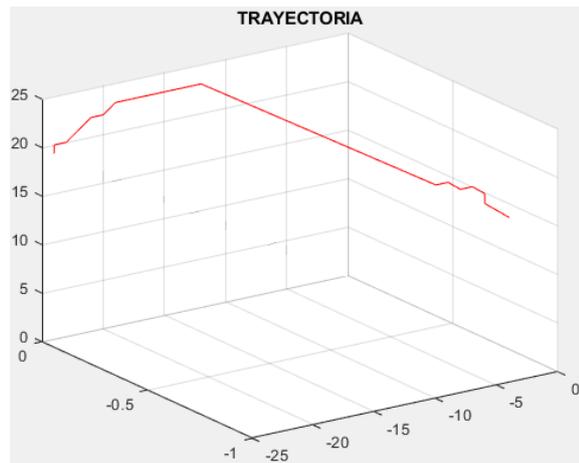
Ilustración 46. Trayectoria expresada en grados de los datos recolectados en la segunda articulación HOMBRO

Fuente: Autores

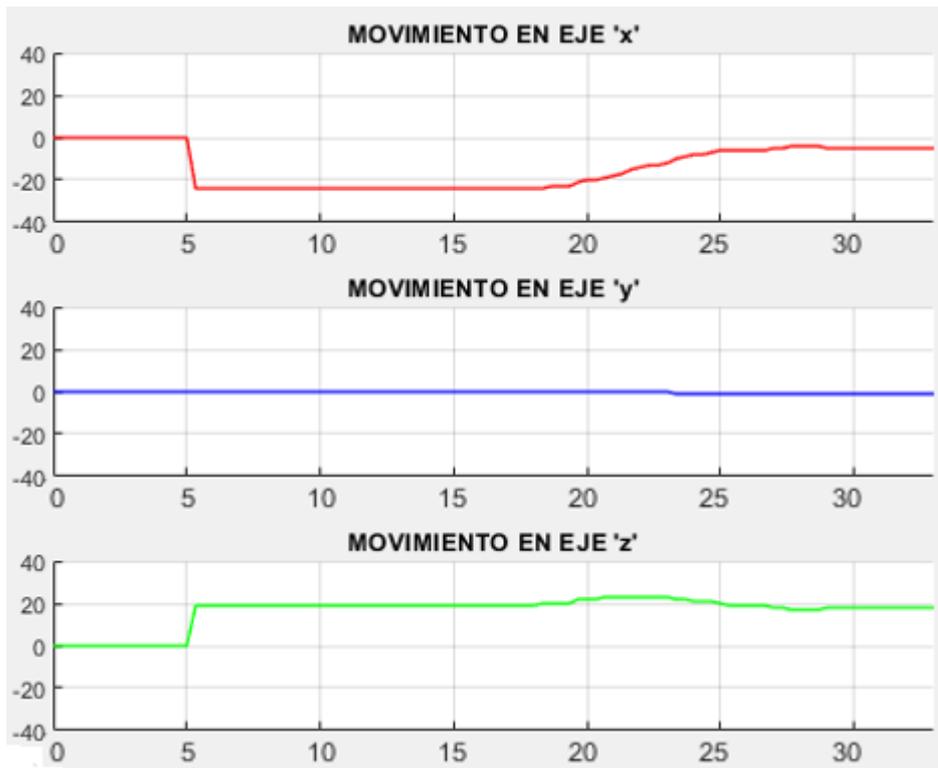
Los datos recolectados del segundo grado de libertad (HOMBRO) por tiempos de muestreo utilizando la cinemática inversa donde la distancia esta expresada en centímetros (cm) y el tiempo en segundos (s) (véase en la ilustración 45), según los ejes de referencia se visualiza que, en el eje X y en el eje Z la recolección tiene un mejor desempeño, y en el eje Y no tiene variación, la enseñanza del movimiento a realizar tiende a llegar a lo requerido por lo que es satisfactoria la tarea enseñada como se muestra en la (ilustración 46) donde su gráfica esta expresada en grados.



(a) Brazo robótico 3GDL (CODO)



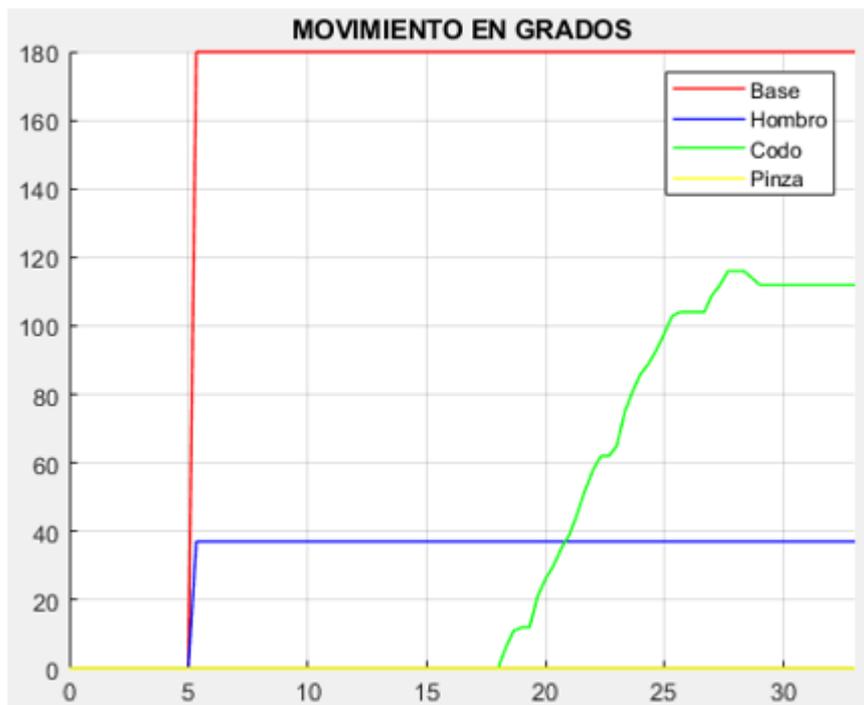
(b) Trayectoria deseada en el 3GDL (CODO)



(c) Trayectoria del 3GDL (CODO) representado en cada eje del plano

**Ilustración 47.** Trayectoria de los datos recolectados en la tercera articulación CODO.

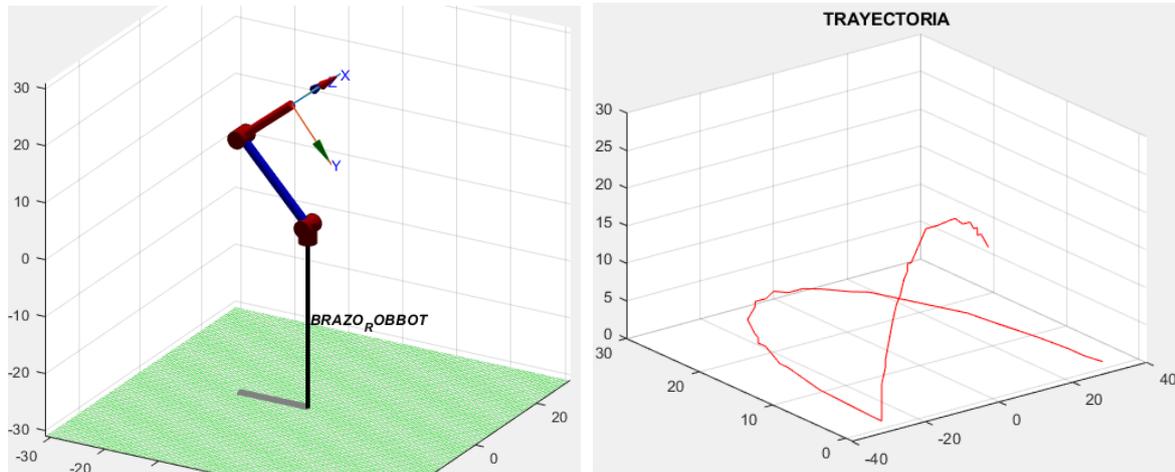
*Fuente:* Autores



**Ilustración 48.** Trayectoria expresada en grados de los datos recolectados en la tercera articulación CODO

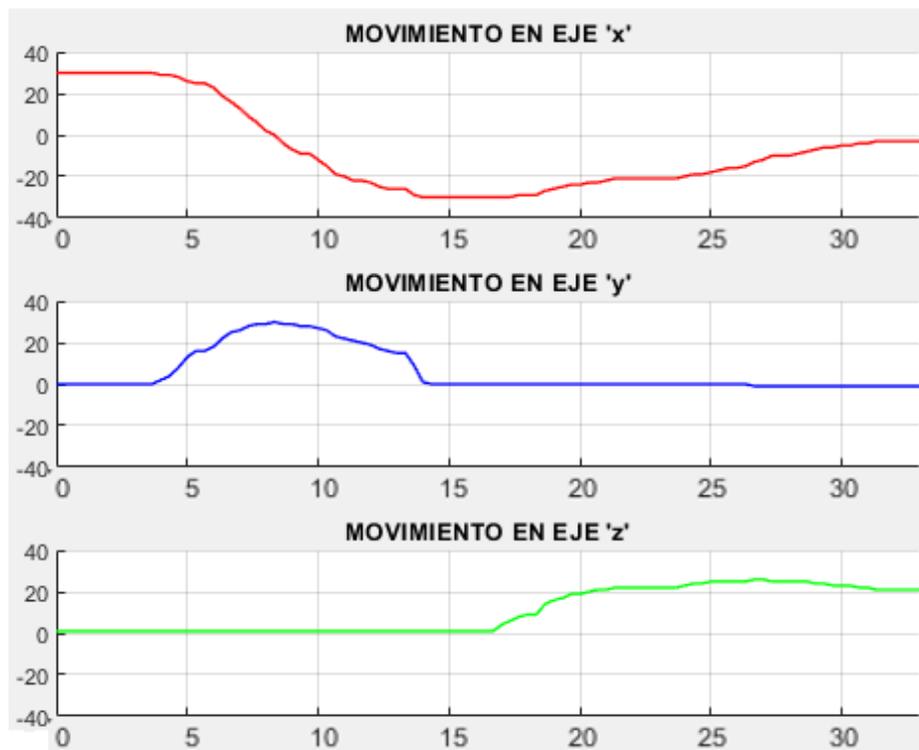
*Fuente:* Autores

Los datos recolectados del tercer grado de libertad (CODO) por tiempos de muestreo utilizando la cinemática inversa donde la distancia esta expresada en centímetros (cm) y el tiempo en segundos (s) (véase en la ilustración 47), según los ejes de referencia se visualiza que, en el eje X y en el eje Z la recolección tiene un mejor desempeño, y en el eje Y no tiene variación, la enseñanza del movimiento a realizar tiende a llegar a lo requerido por lo que es satisfactoria la tarea enseñada como se muestra en la (ilustración 48) donde su gráfica esta expresada en grados.



(a) Brazo robótico

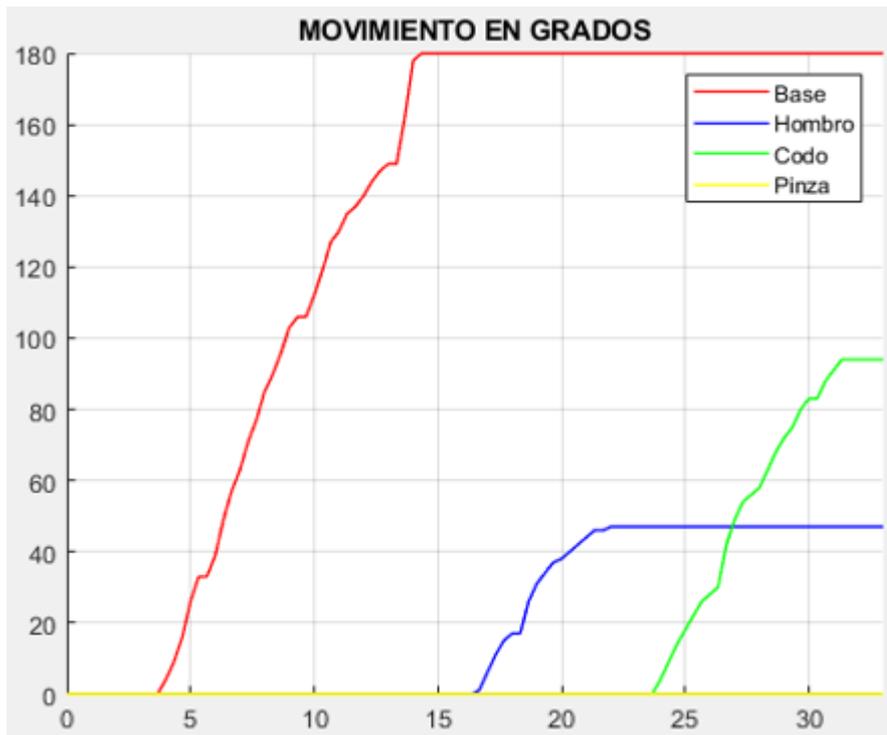
(b) Trayectoria deseada en el Brazo robótico



(c) Trayectoria completa del Brazo robótico representado en cada eje del plano

*Ilustración 49. Trayectoria completa de los datos recolectados en cada grado de libertad*

*Fuente: Autores*



*Ilustración 50. Trayectoria completa expresada en grados de los datos recolectados en cada grado de libertad*

*Fuente: Autores*

Los datos recolectados de la trayectoria completa, es decir, en cada grado de libertad por tiempos de muestreo utilizando la cinemática inversa donde la distancia esta expresada en centímetros (cm) y el tiempo en segundos (s) (véase en la ilustración 49), según los ejes de referencia se visualiza que, en el eje X, en el eje Y, en el eje Z la recolección tiene un mejor desempeño, la enseñanza de los movimientos a realizar tiende a llegar a lo requerido por lo que es satisfactoria la tarea enseñada como se muestra en la (ilustración 50) donde su gráfica esta expresada en grados.

Para mejorar la recolección de datos y la precisión del movimiento del encoder, se utilizaron las interrupciones dentro de la placa Arduino Uno, como resultado, realizando las pruebas con los 4 grados de libertad, los datos se tomarían siempre y cuando, no exista la interrupción generada por el movimiento de aprendizaje, a su vez la capacidad del vector era moderada.

Al trabajar con la placa Arduino Uno, la misma que posee dos interrupciones, se realizaron las conexiones de dos grados de libertad en cada interrupción de manera consecutiva es decir, la base con el hombro y el codo con la pinza, al momento de realizar las pruebas dio como

resultado la obtención de los datos en cada punto, pero al haber un movimiento no dado al grado de libertad correspondiente este activaba la interrupción al que estaba asociado, por ende, el sistema se paraba hasta salir de esa interrupción, lo cual no sería la forma más óptima para el método de enseñanza en la recolección de datos de movimiento.

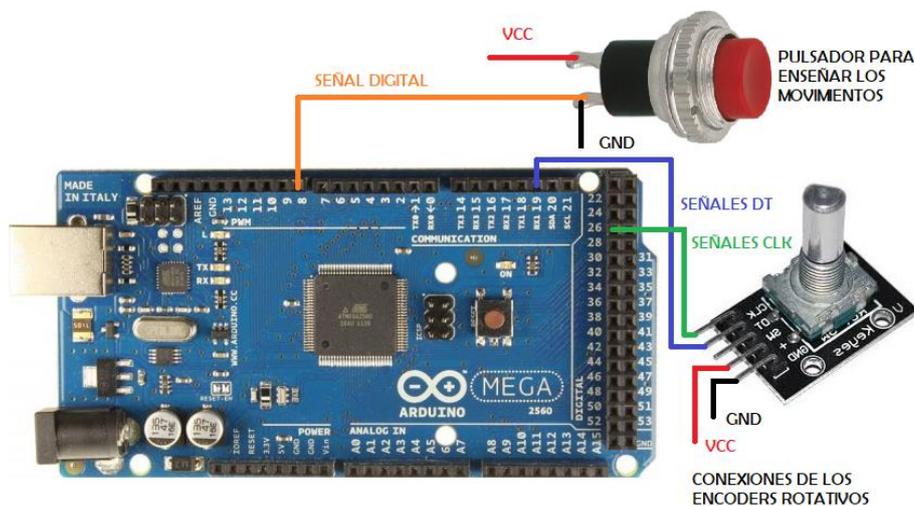
Debido a esto y a las limitaciones del hardware para incrementar las demás etapas dentro de la placa Arduino Uno, se consideró trabajar con el Arduino Mega 2560, ya que esta posee más de dos interrupciones (véase la Tabla 8) las cuales van hacer necesarias para optimizar el funcionamiento de todas las etapas.

**Tabla 8.** Interrupciones que posee ARDUINO

*Fuente: Autores*

ARDUINO	VECTOR DE INTERRUPTIONES					
	INT 0	INT 1	INT 2	INT 3	INT 4	INT 5
UNO	PIN 2	PIN 3				
MEGA	PIN 2	PIN 3	PIN 21	PIN 20	PIN 19	PIN 18
NANO	PIN 2	PIN 3				

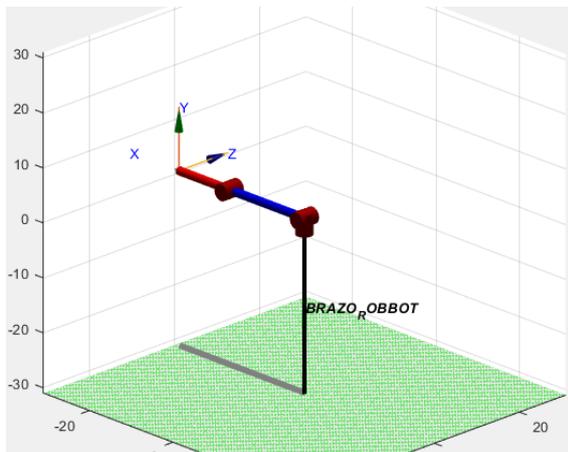
Consecuentemente con el Arduino Mega 2560, se utilizaron 4 de los pines de interrupción para conectar cada uno de los encoder correspondientes a los grados de libertad, dando como resultado el manejo de los valores de recolección de forma individual, mejorando la forma de aprendizaje al manipular el brazo robótico (véase ilustración 51).



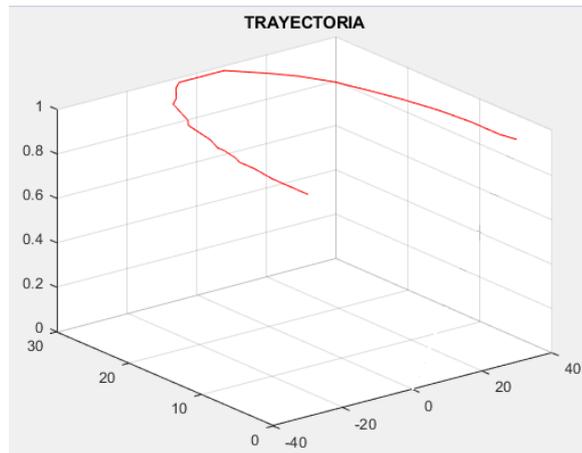
**Ilustración 51.** Conexiones de encoder a pines de interrupción

*Fuente: Autores*

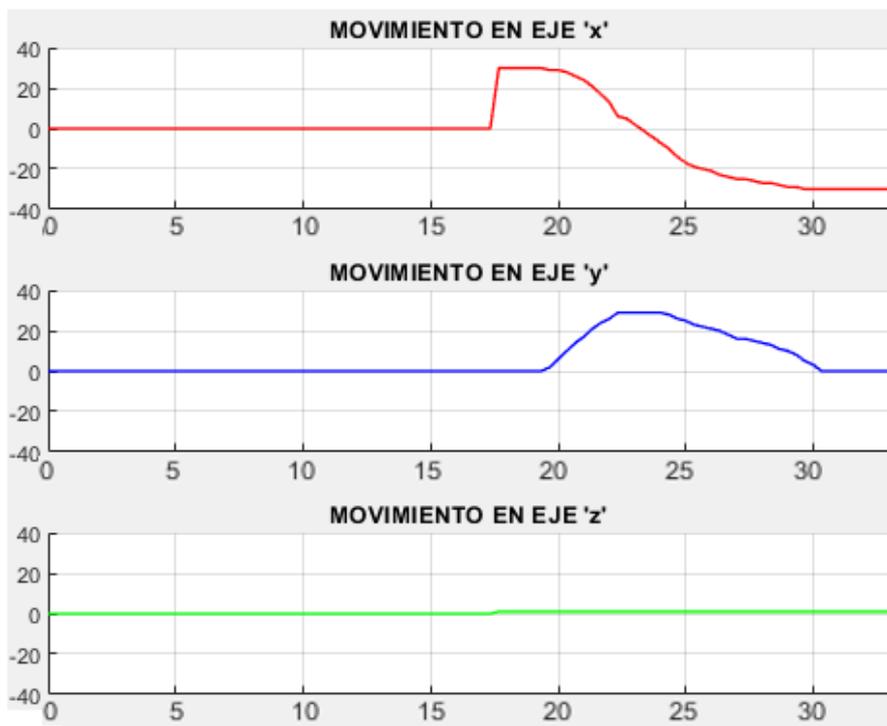
Seguido de lo anterior, se realizaron pruebas que nos ayudan a determinar la simultaneidad de todos los grados de libertad los mismos que tienen como unidad de medida en la distancia los centímetros (cm) y el tiempo en segundos (s) incluyendo con esto, la independencia de la velocidad de movimiento, el cual se lo va analizar en metros sobre segundo (m/s), es decir, que el brazo robótico tenga la capacidad de ejecutar tanto movimientos rápidos o lentos de forma simultánea, según el escenario que lo requiera.



(a) 1 GDL BASE



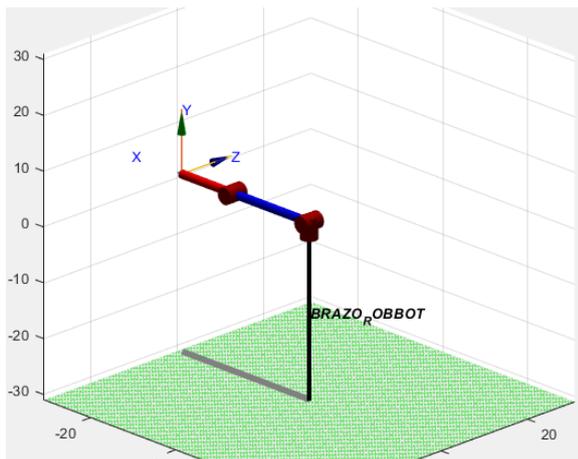
(b) Trayectoria deseada del 1 GDL BASE con una velocidad 1



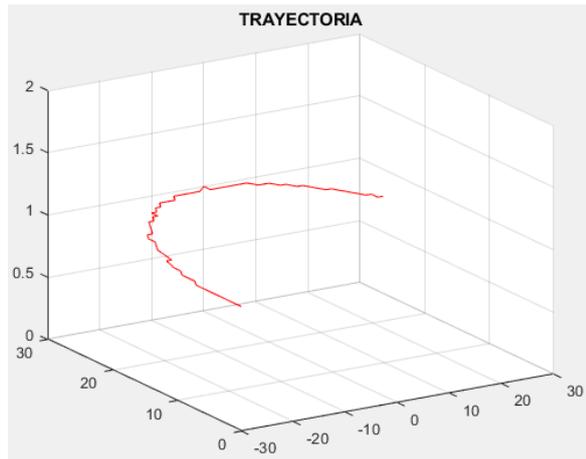
(c) Trayectoria deseada en cada eje del plano del 1 GDL BASE con una velocidad 1

**Ilustración 52.** Trayectoria de la recolección de los datos de la primera articulación Base, con una velocidad 1

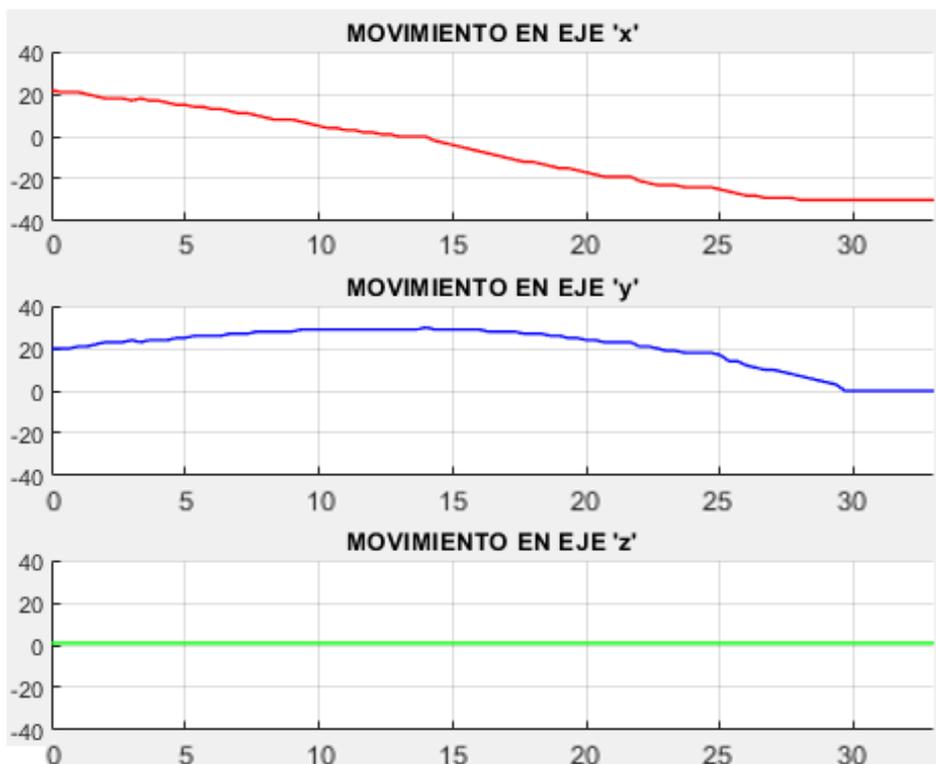
**Fuente:** Autores



(a) 1 GDL BASE



(b) Trayectoria deseada del 1 GDL BASE con una velocidad 2



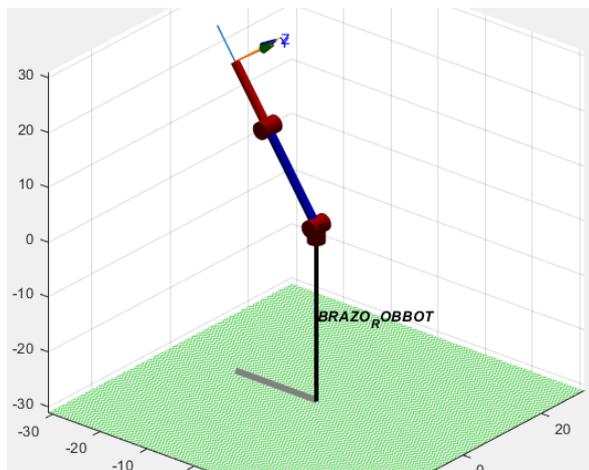
(c) Trayectoria deseada en cada eje del plano del 1 GDL BASE con una velocidad 2

**Ilustración 53.** Trayectoria de la recolección de los datos de la primera articulación Base, con una velocidad 2

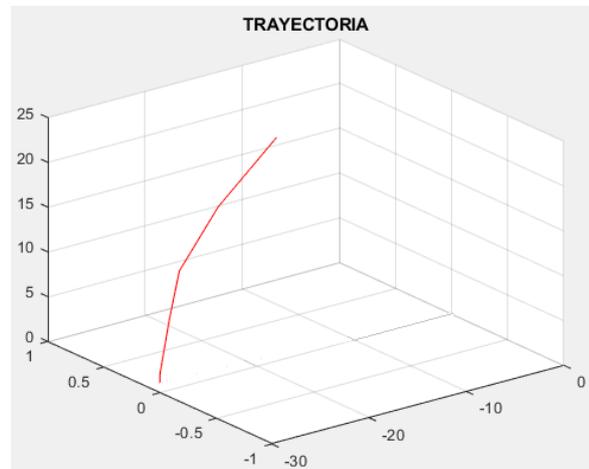
*Fuente:* Autores

Las gráficas que se muestran anteriormente, dan a conocer con que velocidad se realiza la recolección de los datos de la primera articulación (BASE), para esto se sabe que las velocidades pueden ser tanto rápidas como lentas, teniendo como unidad de medida el metro sobre segundo ( $m/s$ ), esto depende de cómo el usuario quiera realizar la acción y del escenario que se tenga, con respecto a las pruebas realizadas en la (ilustración 52), se toman puntos de

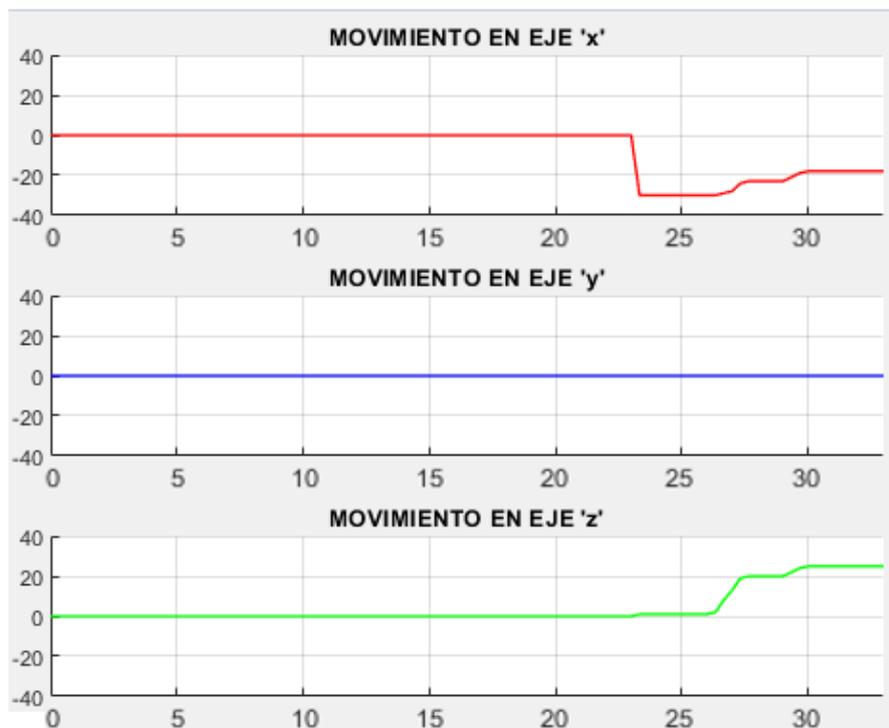
la trayectoria, es decir, un muestreo que van desde  $0,3\text{ m}$  a  $-0,21\text{ m}$  en un intervalo de tiempo de  $17\text{ s}$  a  $30\text{ s}$ , dando como resultado una velocidad 1 igual a  $0,06\text{ m/s}$ , a diferencia de la (ilustración 53), en la cual se toman puntos de la trayectoria que van desde  $0,21\text{ m}$  a  $-0,24\text{ m}$  en un intervalo de tiempo de  $0\text{ s}$  a  $25\text{ s}$ , dando como resultado una velocidad 2 igual a  $0,03\text{ m/s}$ . Por lo tanto, la recolección de los datos de la Base con la velocidad 1 es rápida a comparación con la velocidad 2 que recoge los datos de manera lenta.



(a) 2 GDL HOMBRO



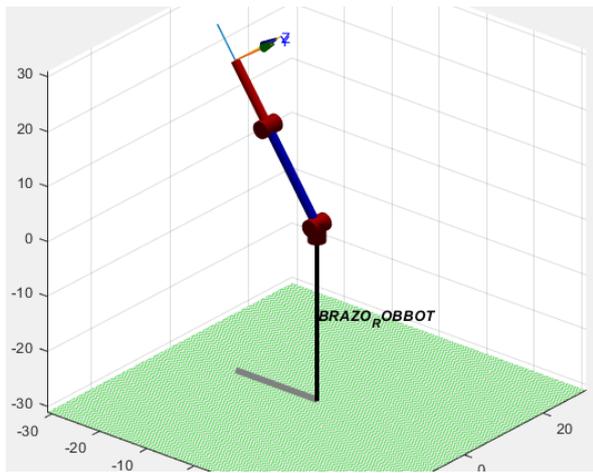
(b) Trayectoria deseada del 2 GDL HOMBRO con una velocidad 1



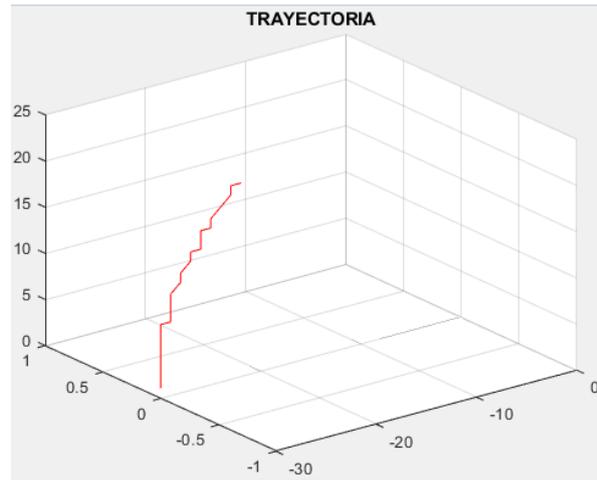
(c) Trayectoria deseada en cada eje del plano del 2 GDL HOMBRO con una velocidad 1

**Ilustración 54.** Trayectoria de la recolección de los datos de la segunda articulación Hombro, con una velocidad 1

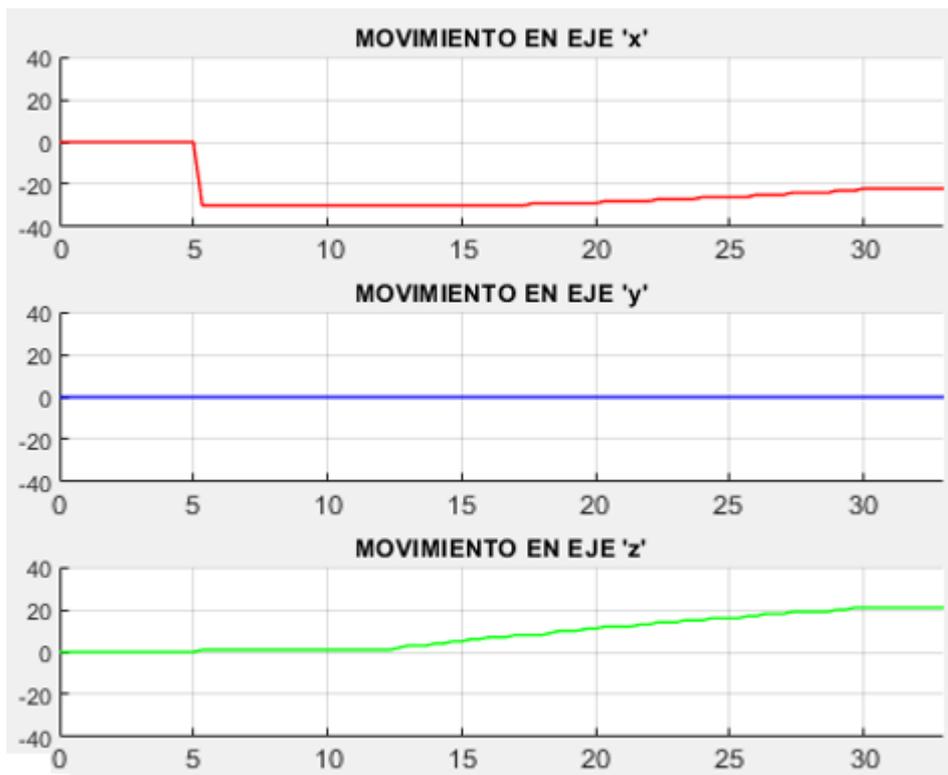
*Fuente:* Autores



(a) 2 GDL HOMBRO



(b) Trayectoria deseada del 2 GDL HOMBRO con una velocidad 2



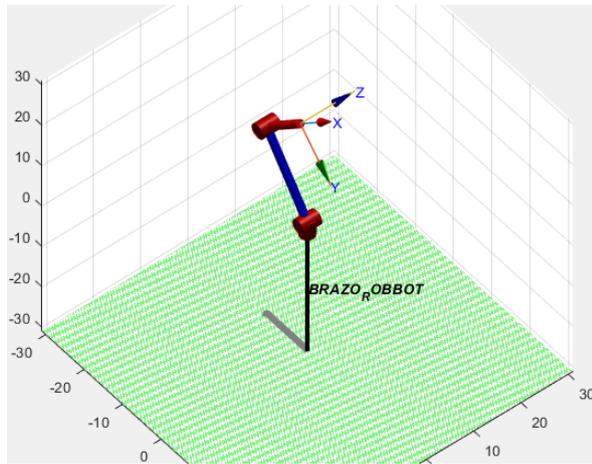
(c) Trayectoria deseada en cada eje del plano del 2 GDL HOMBRO con una velocidad 2

**Ilustración 55.** Trayectoria de la recolección de los datos de la segunda articulación Hombro, con una velocidad 2

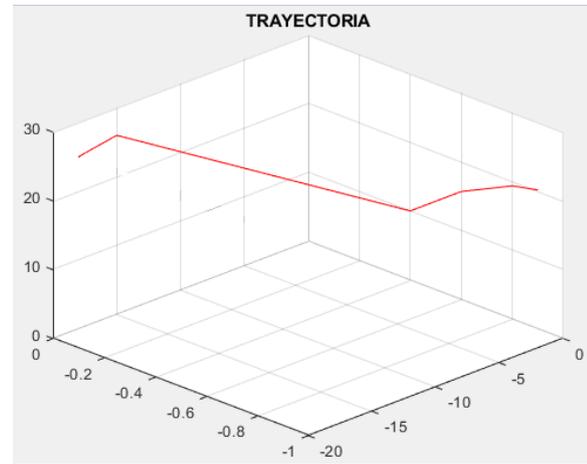
**Fuente:** Autores

Las gráficas que se muestran anteriormente, dan a conocer con qué velocidad se realiza la recolección de los datos de la segunda articulación (HOMBRO), para esto se sabe que las velocidades pueden ser tanto rápidas como lentas, teniendo como unidad de medida el metro sobre segundo ( $m/s$ ), esto depende de cómo el usuario quiera realizar la acción y del escenario que se tenga, con respecto a las pruebas realizadas en la (ilustración 54), se toman puntos de

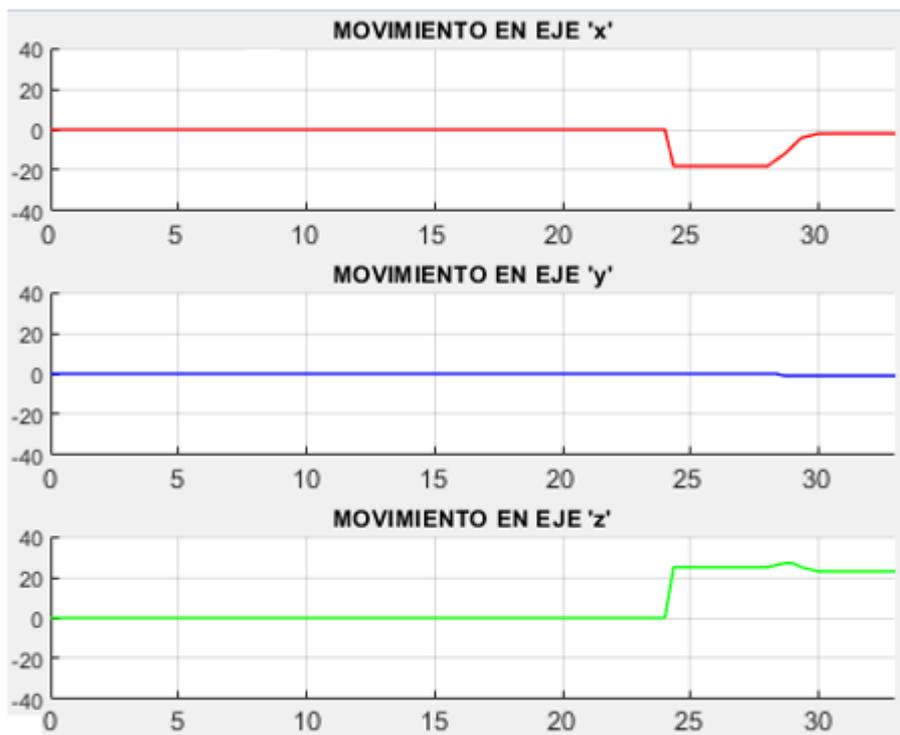
la trayectoria, es decir, un muestreo que van desde  $0m$  a  $0,22m$  en un intervalo de tiempo de  $20s$  a  $30s$ , dando como resultado una velocidad 1 igual a  $0,72 m/s$ , a diferencia de la (ilustración 55), en la cual se toman puntos de la trayectoria que van desde  $0,1m$  a  $0,21m$  en un intervalo de tiempo de  $12s$  a  $30s$ , dando como resultado una velocidad 2 igual a  $0,3 m/s$ . Por lo tanto, la recolección de los datos del Hombro con la velocidad 1 es rápida a comparación con la velocidad 2 que recoge los datos de manera lenta.



(a) 3 GDL CODO



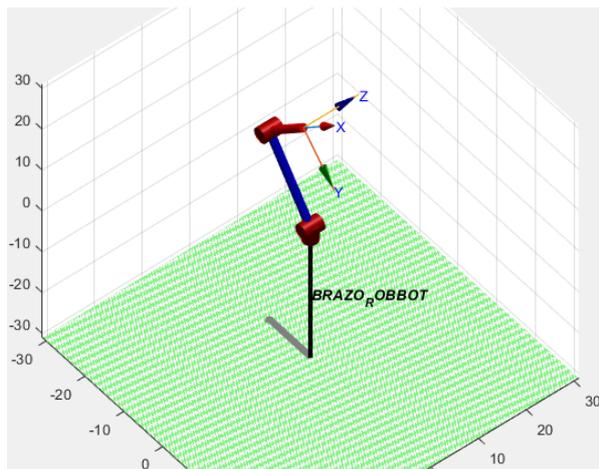
(b) Trayectoria deseada del 3 GDL CODO con una velocidad 1



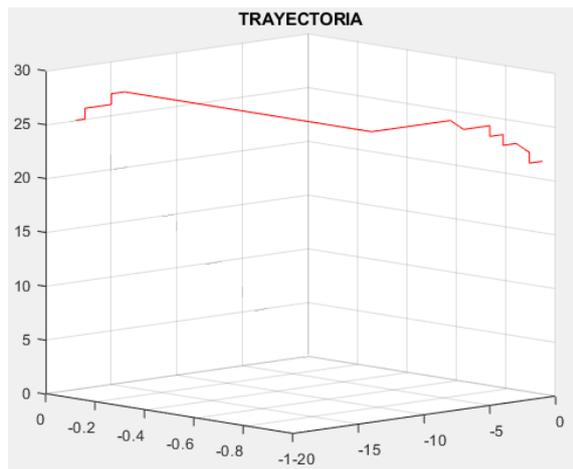
(c) Trayectoria deseada en cada eje del plano del 3 GDL CODO con una velocidad 1

**Ilustración 56.** Trayectoria de la recolección de los datos de la tercera articulación Codo, con una velocidad 1

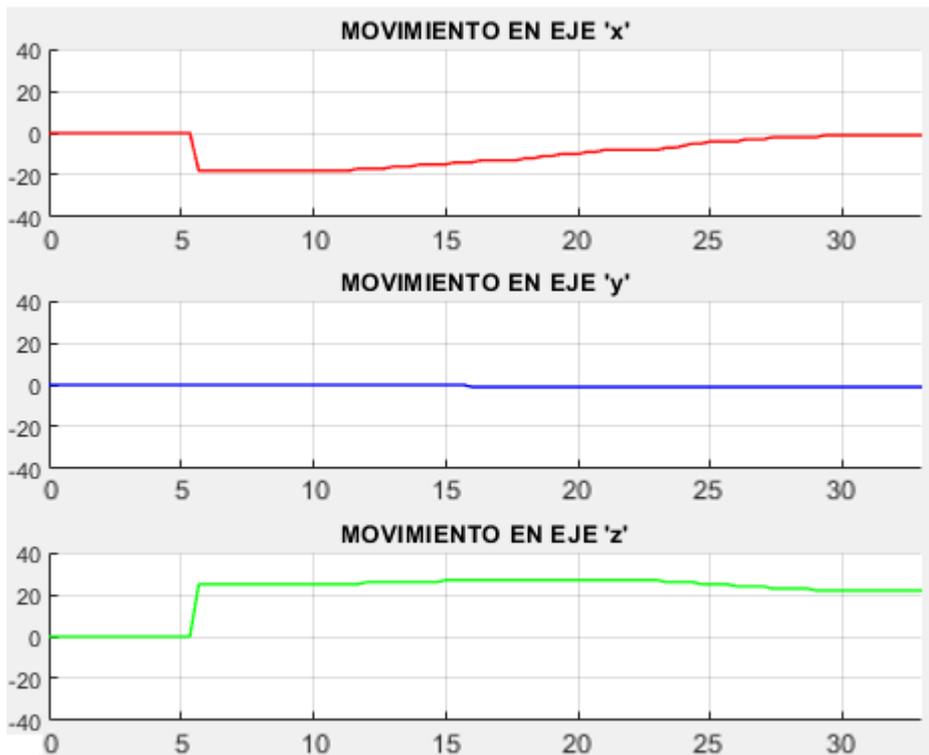
*Fuente:* Autores



(b) 3 GDL CODO



(b) Trayectoria deseada del 3 GDL CODO con una velocidad 2



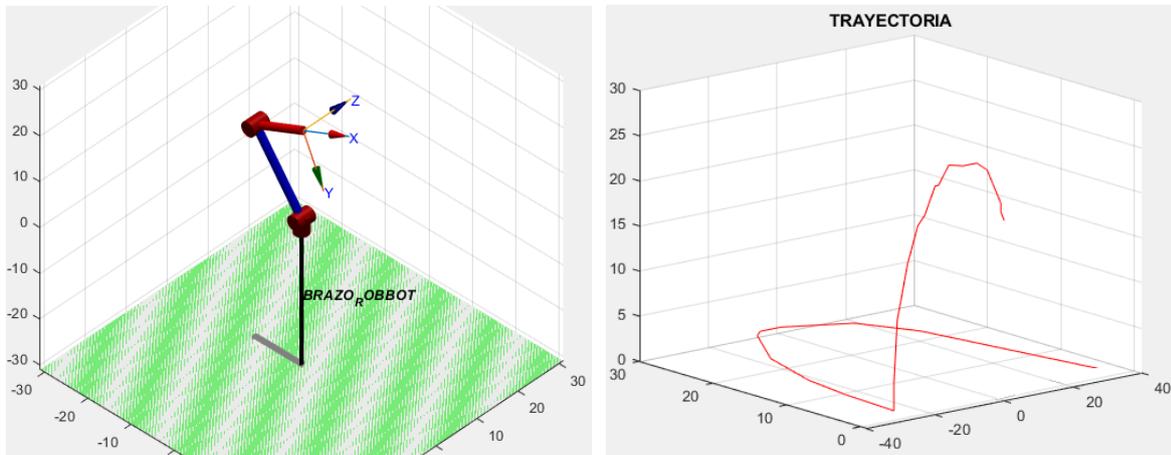
(c) Trayectoria deseada en cada eje del plano del 3 GDL CODO con una velocidad 2

**Ilustración 57.** Trayectoria de la recolección de los datos de la tercera articulación Codo, con una velocidad 2

*Fuente:* Autores

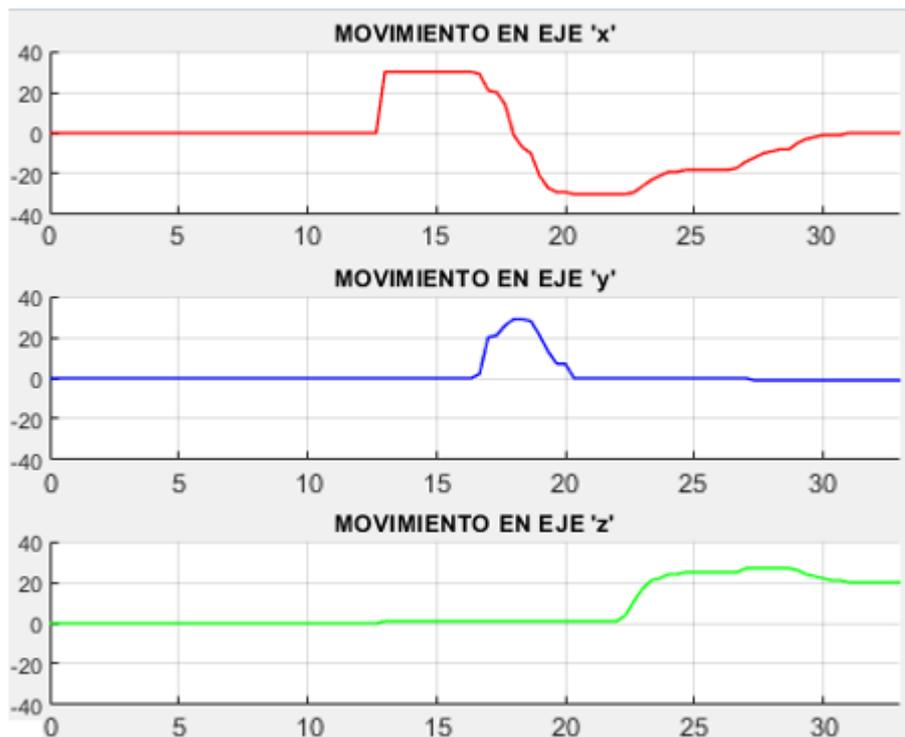
Las gráficas que se muestran anteriormente, dan a conocer con que velocidad se realiza la recolección de los datos de la tercera articulación (CODO), para esto se sabe que las velocidades pueden ser tanto rápidas como lentas, teniendo como unidad de medida el metro sobre segundo ( $m/s$ ), esto depende de cómo el usuario quiera realizar la acción y del escenario que se tenga, con respecto a las pruebas realizadas en la (ilustración 56), se toman puntos de

la trayectoria, es decir, un muestreo que van desde  $0m$  a  $0,23m$  en un intervalo de tiempo de  $20s$  a  $30s$ , dando como resultado una velocidad 1 igual a  $1 m/s$ , a diferencia de la (ilustración 57), en la cual se toman puntos de la trayectoria que van desde  $0m$  a  $0,22m$  en un intervalo de tiempo de  $5s$  a  $30s$ , dando como resultado una velocidad 2 igual a  $0,4 m/s$ . Por lo tanto, la recolección de los datos del Codo con la velocidad 1 es rápida a comparación con la velocidad 2 que recoge los datos de manera lenta.



(a) Brazo robótico

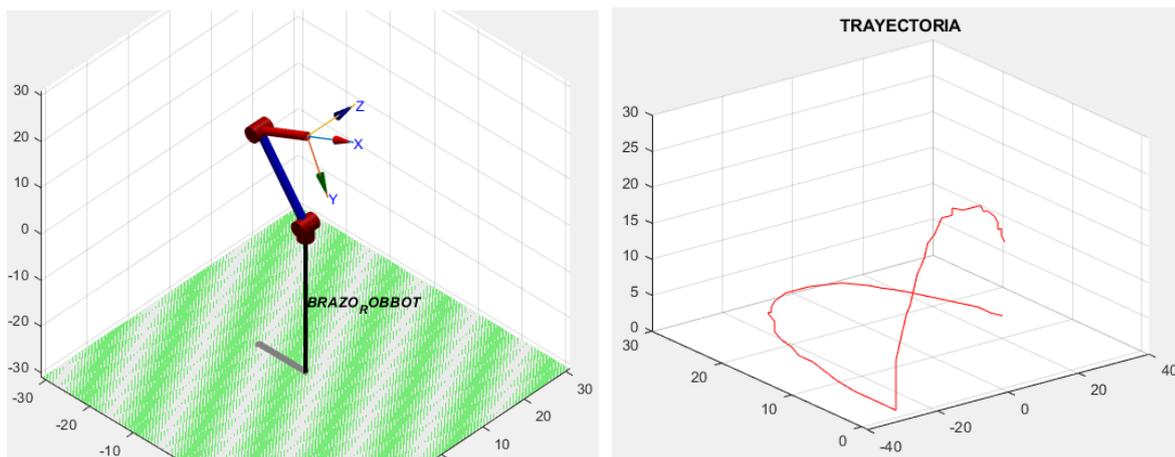
(b) Trayectoria deseada en el Brazo robótico con una velocidad 1



(c) Trayectoria completa del Brazo robótico representado en cada eje del plano con una velocidad 1

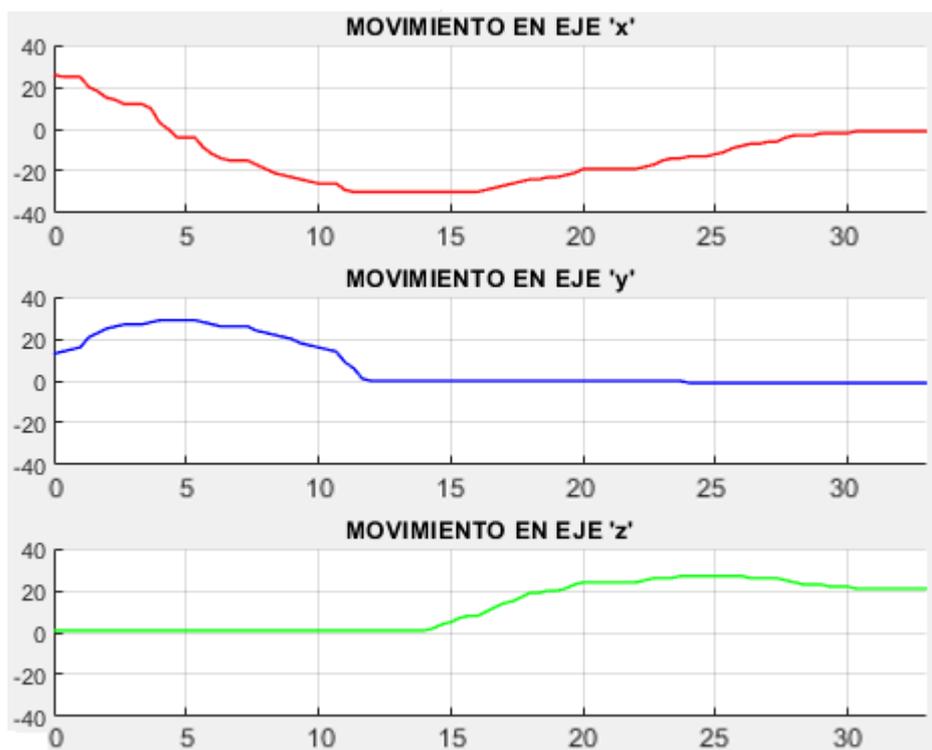
*Ilustración 58. Trayectoria completa de los datos recolectados en cada grado de libertad con una velocidad 1*

*Fuente: Autores*



(b) Brazo robótico

(b) Trayectoria deseada en el Brazo robótico con una velocidad 2



(c) Trayectoria completa del Brazo robótico representado en cada eje del plano con una velocidad 2

*Ilustración 59.* Trayectoria completa de los datos recolectados en cada grado de libertad con una velocidad 2

*Fuente:* Autores

Las gráficas que se muestran anteriormente, dan a conocer con que velocidad se realiza la recolección de los datos de la enseñanza completa del brazo robótico para esto se sabe que las velocidades pueden ser tanto rápidas como lentas, teniendo como unidad de medida el metro sobre segundo ( $m/s$ ), esto depende de cómo el usuario quiera realizar la acción y del escenario que se tenga, con respecto a las pruebas realizadas en la (ilustración 58), se toman puntos de

la trayectoria, es decir, un muestreo que van desde  $0m$  a  $0,17m$  en un intervalo de tiempo de  $10s$  a  $30s$ , dando como resultado una velocidad 1 igual a  $0,03 m/s$ , a diferencia de la (ilustración 59), en la cual se toman puntos de la trayectoria que van desde  $0,26m$  a  $0,2m$  en un intervalo de tiempo de  $0s$  a  $30s$ , dando como resultado una velocidad 2 igual a  $0,1 m/s$ . Por lo tanto, la recolección de los datos de la enseñanza completa del brazo robótico con la velocidad 1 es rápida a comparación con la velocidad 2 que recoge los datos de manera lenta.

#### 4.2.3 PRUEBAS DE LECTURA O REPRODUCCIÓN DE ENSEÑANZA

La primera prueba fue realizada en Arduino Uno mediante un pulsador y utilizando la librería servo.h, para enviar los datos de posición de enseñanza, como resultado se logró la reproducción correcta de un solo grado de libertad, pero, al ir incrementando las etapas en conjunto que conlleva el aprendizaje directo, se optó por utilizar Arduino Mega.

De igual manera en el Arduino Mega se toma en cuenta el uso de un pulsador para la reproducción de movimientos, cuya instrucción va a estar colocada dentro del void loop, con el fin de que, después de transcurrido el tiempo de enseñanza del brazo robótico, teniendo los valores guardados en un vector, se presione el pulsador y mande a reproducir todo lo enseñado de manera consecutiva.

Como comprobación de reproducción podemos visualizar desde el monitor serial de Arduino, las posiciones guardadas en los vectores, para corroborar que los valores tomados anteriormente por los encoders sean los mismos, antes de realizar la energización de los servos, como resultado de esto, se tuvo más claro que los movimientos del brazo van hacer correctos (véase la tabla 9).

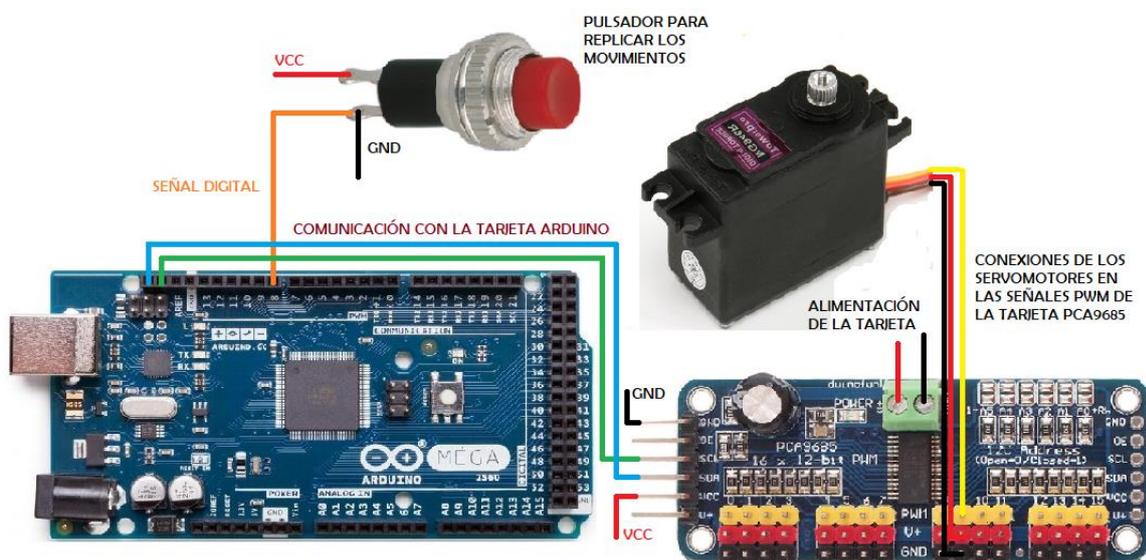
*Tabla 9. Posiciones recolectadas y posiciones reproducidas.*

*Fuente: Autores*

V. RECOLECTADOS	V. REPRODUCIDOS
Posición 0 : 0	Posición 0 : 0
Posición 1 : 0	Posición 1 : 0
Posición 2 : 10	Posición 2 : 10
Posición 3 : 10	Posición 3 : 10
Posición 4 : 21	Posición 4 : 21
Posición 5 : 32	Posición 5 : 32
Posición 6 : 43	Posición 6 : 43
Posición 7 : 54	Posición 7 : 54
Posición 8 :75	Posición 8 :75
Posición 9 : 86	Posición 9 : 86

Posición 10 : 97	Posición 10 : 97
Posición 11 : 108	Posición 11 : 108
Posición 12 : 118	Posición 12 : 118
Posición 13 : 129	Posición 13 : 129
Posición 14 : 140	Posición 14 : 140
Posición 15 : 151	Posición 15 : 151
Posición 16 : 162	Posición 16 : 162
Posición 17 : 172	Posición 17 : 172
Posición 18 : 194	Posición 18 : 194
Posición 19 : 205	Posición 19 : 205
Posición 20 : 216	Posición 20 : 216

Seguido de lo anterior mencionado, se procedió a realizar las conexiones conjuntamente con la fuente de PC, el controlador de servos PCA9685 y los servomotores (véase ilustración 60), que, al momento de presionar el pulsador, se reproducen los movimientos enseñados, teniendo en claro que el PCA9685 trabaja a 12 bits, por lo que los rangos de posición van hacer de 0 a 420 equivalentes de 0 a 180 grados, obteniendo así, la reproducción total de todas las posiciones enseñadas.

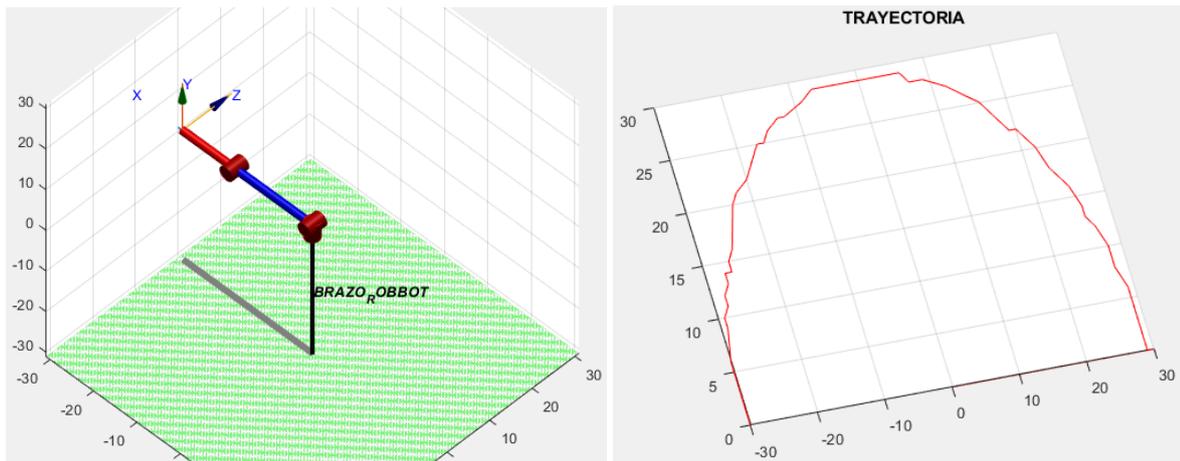


*Ilustración 60. Diagrama de conexión para la reproducción*

*Fuente: Autores*

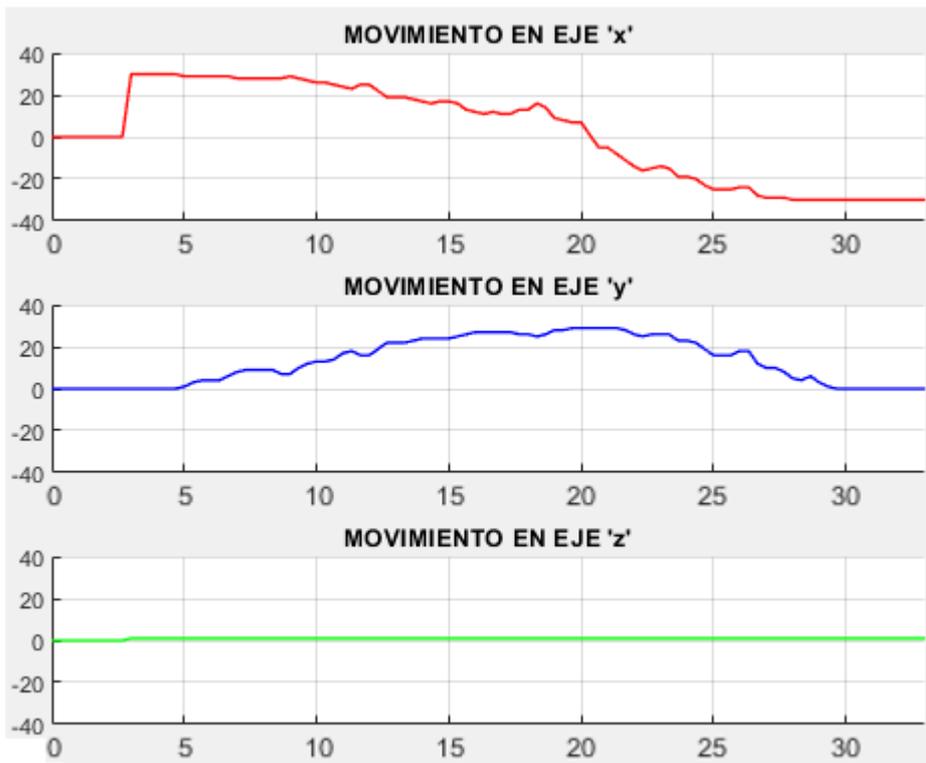
Para verificar el correcto funcionamiento del sistema de reproducción de datos, se realizaron las siguientes pruebas, que consisten en la reproducción de forma individual donde la distancia se da en centímetros (cm) como unidad de medida y el tiempo en segundos (s), es decir, ir verificando como los datos tomados se reproducen, esto se lo hace por cada grado de libertad, para dicho proceso se le asignaron trayectorias a cada uno de ellos, además se

visualiza el margen de error entre la reproducción realizada con la reproducción deseada, todo esto se muestra a continuación.



(a) Brazo robótico 1GDL BASE

(b) Trayectoria deseada del 1GDL BASE



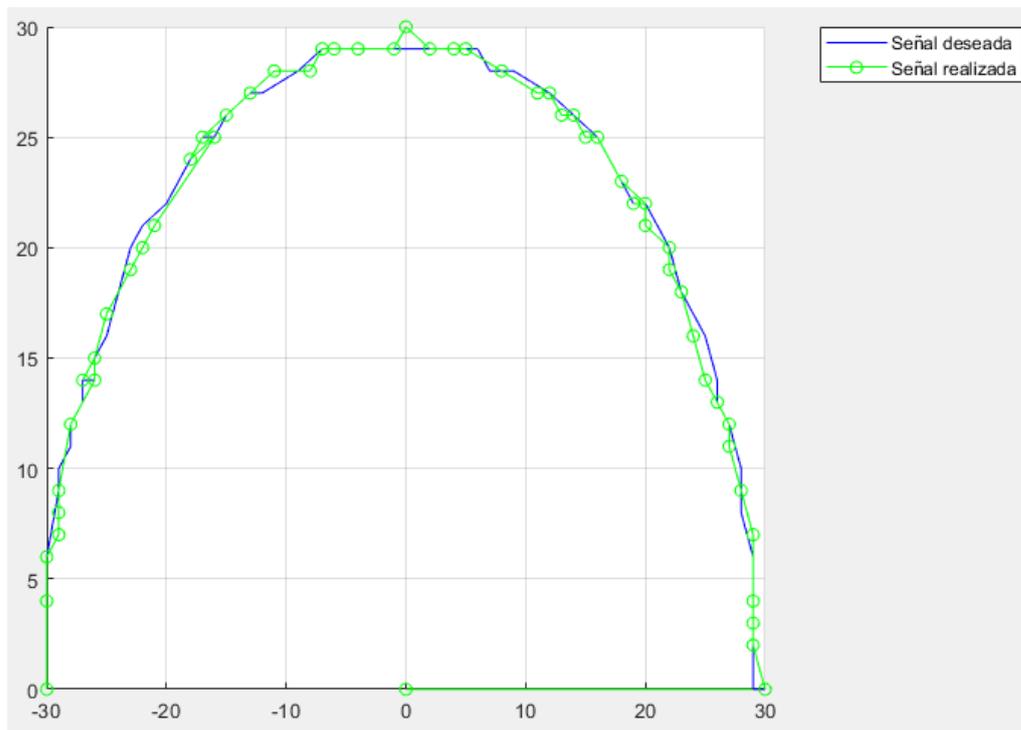
(c) Trayectoria realizada representada en cada eje del plano del 1GDL BASE

*Ilustración 61. Trayectoria realizada de la reproducción de los datos de la primera articulación Base*

*Fuente: Autores*

Los resultados obtenidos en la reproducción de los datos del primer grado de libertad BASE véase (ilustración 61), muestran que la trayectoria realizada por el 1GDL tiene un margen de

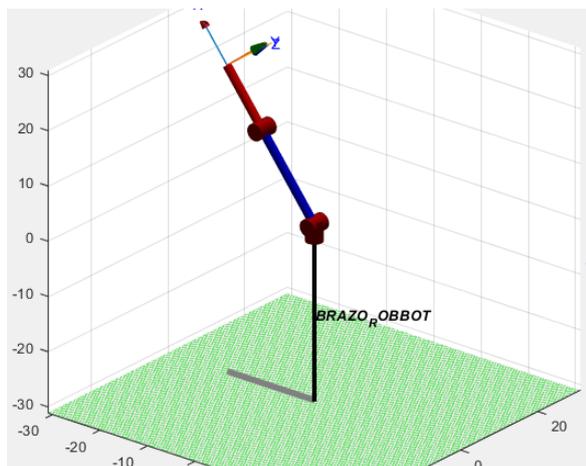
error, para poder obtener el margen de error de la trayectoria realizada se generó una comparativa entre la trayectoria deseada y la trayectoria realizada en función del tiempo.



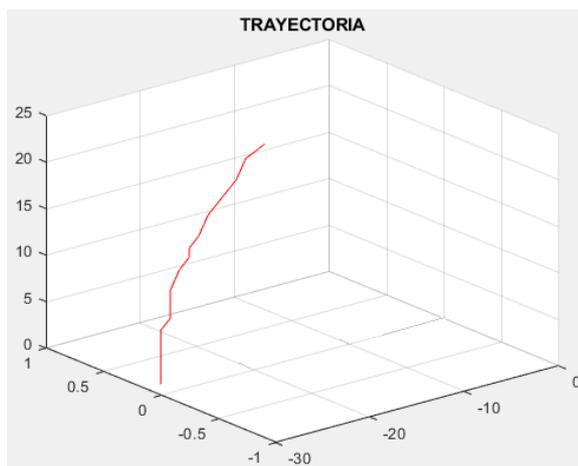
**Ilustración 62.** Trayectoria realizada de la reproducción de los datos de la primera articulación Base con respecto a la trayectoria deseada.

**Fuente:** Autores

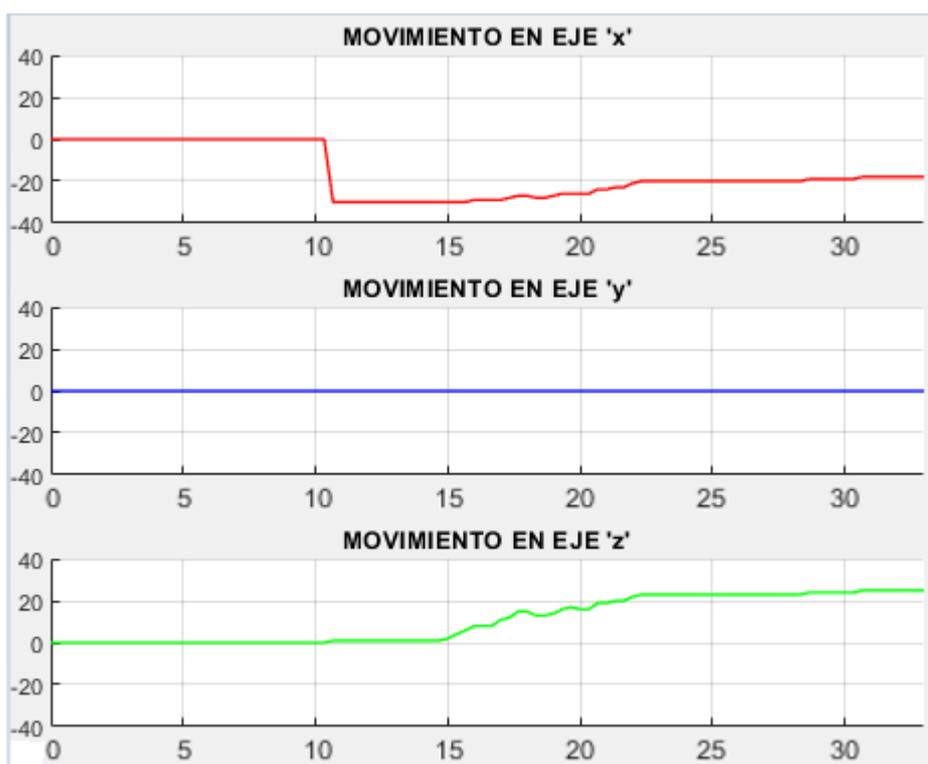
Los datos obtenidos en la reproducción del primer grado de libertad BASE véase (ilustración 62), muestran que el margen de error en la trayectoria es menor a 0.5 cm como máximo, esto debido a que en la recolección de los datos, algunos valores suelen ser alterados, esto se da por la mala manipulación o enseñanza de cada articulación, se puede considerar a este margen de error como despreciable, obteniendo un resultado exitoso en la trayectoria de reproducción de los datos de la Base.



(a) Brazo robótico 2GDL HOMBRO



(b) Trayectoria deseada del 2GDL HOMBRO

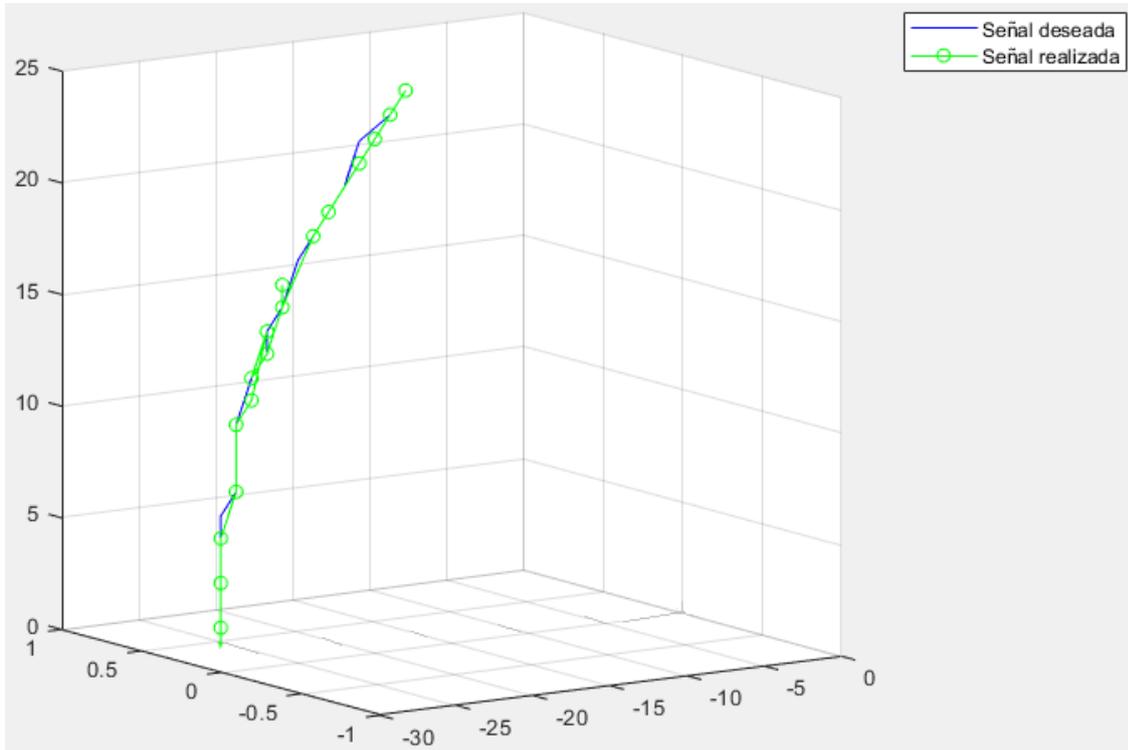


(c) Trayectoria realizada representada en cada eje del plano del 2GDL HOMBRO

*Ilustración 63. Trayectoria realizada de la reproducción de los datos de la segunda articulación Hombro*

**Fuente:** Autores

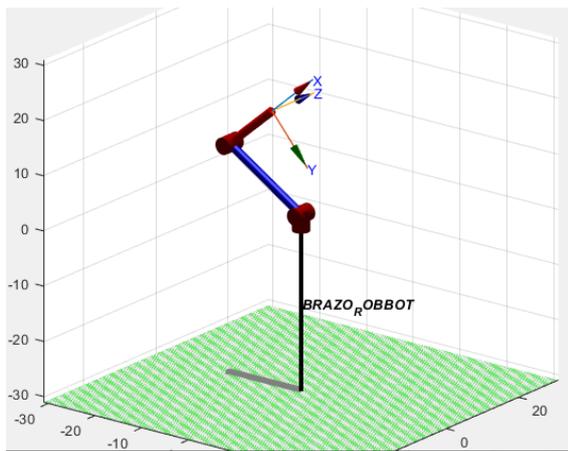
Los resultados obtenidos en la reproducción de los datos del segundo grado de libertad HOMBRO véase (ilustración 63), muestran que la trayectoria realizada por el 2GDL tiene un margen de error, para poder obtener el margen de error de la trayectoria realizada se generó una comparativa entre la trayectoria deseada y la trayectoria realizada en función del tiempo.



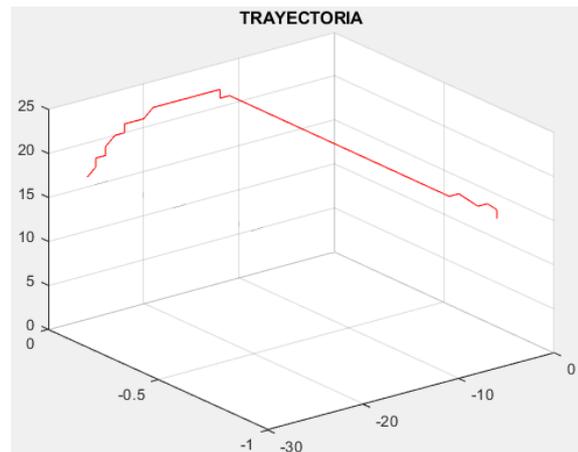
**Ilustración 64.** Trayectoria realizada de la reproducción de los datos de la segunda articulación Hombro con respecto a la trayectoria deseada.

**Fuente:** Autores

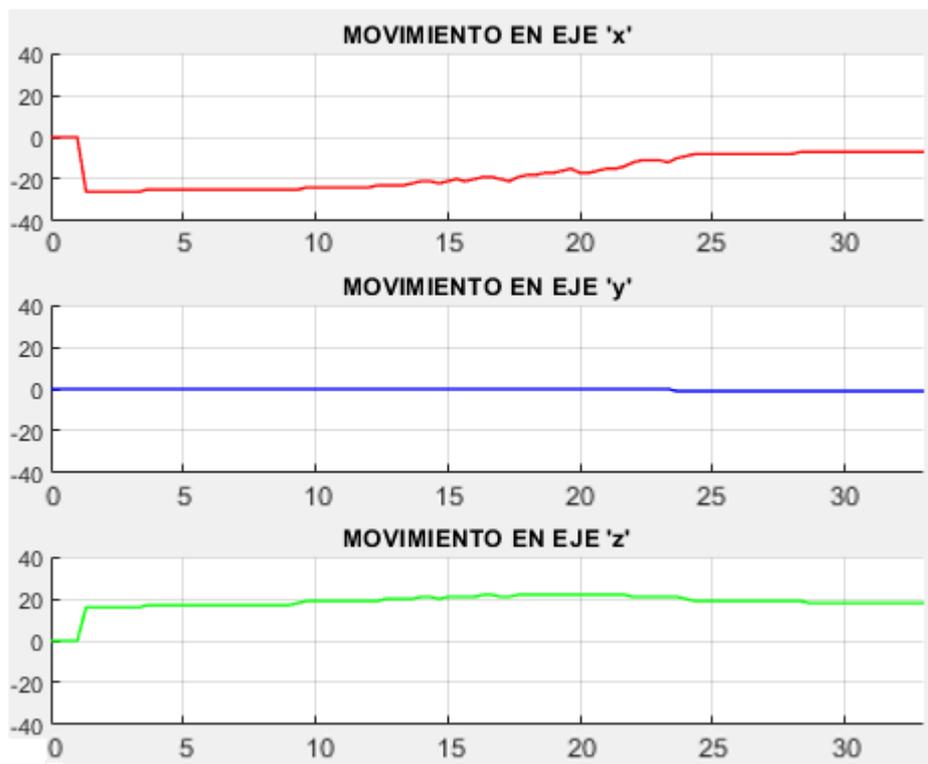
Los datos obtenidos en la reproducción del segundo grado de libertad HOMBRO véase (ilustración 64), muestran que el margen de error en la trayectoria es menor a 0.3 cm como máximo, esto debido a que en la recolección de los datos, algunos valores suelen ser alterados, esto se da por la mala manipulación o enseñanza de cada articulación, se puede considerar a este margen de error como despreciable, obteniendo un resultado exitoso en la trayectoria de reproducción de los datos del Hombro.



(a) Brazo robótico 3GDL CODO



(b) Trayectoria deseada del 3GDL CODO

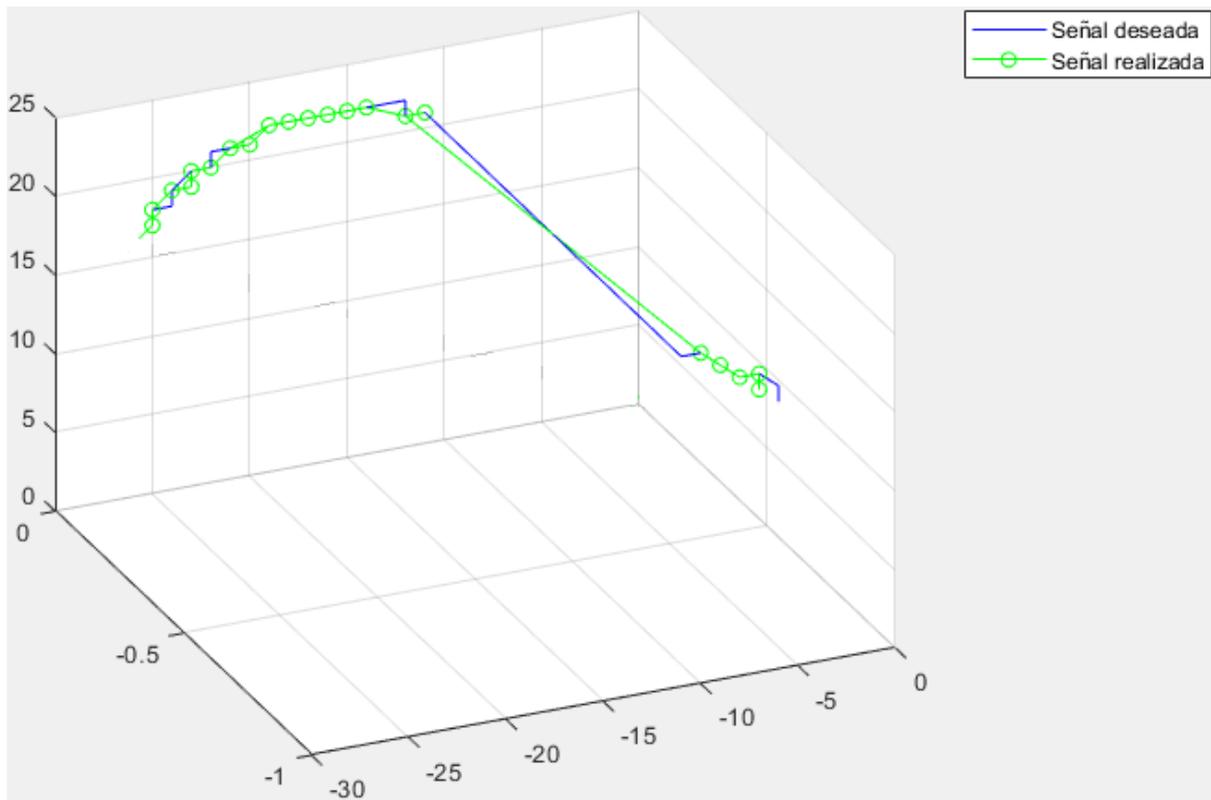


(c) Trayectoria realizada representada en cada eje del plano del 3GDL CODO

*Ilustración 65. Trayectoria realizada de la reproducción de los datos de la tercera articulación Codo*

*Fuente: Autores*

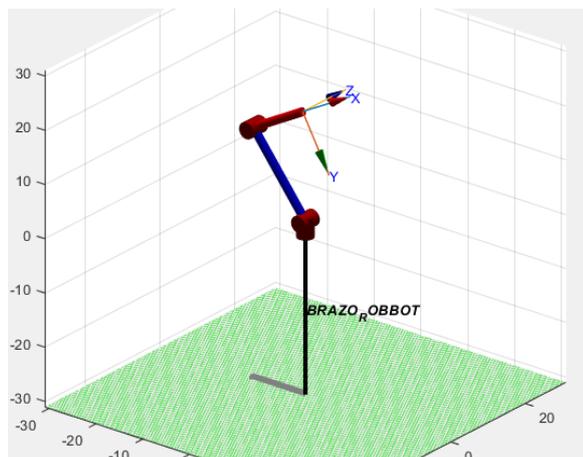
Los resultados obtenidos en la reproducción de los datos del tercer grado de libertad CODO véase (ilustración 65), muestran que la trayectoria realizada por el 3GDL tiene un margen de error, para poder obtener el margen de error de la trayectoria realizada se generó una comparativa entre la trayectoria deseada y la trayectoria realizada en función del tiempo.



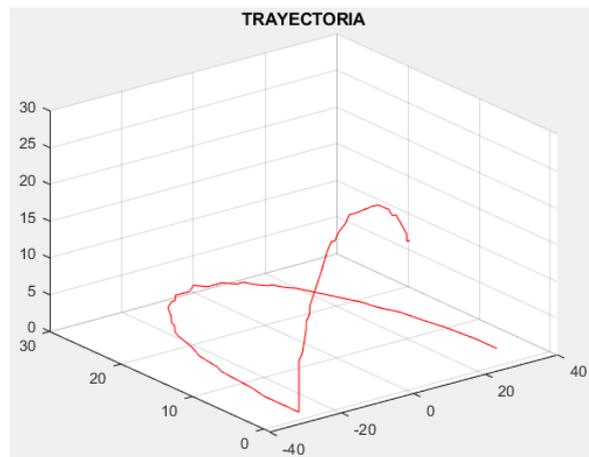
**Ilustración 66.** Trayectoria realizada de la reproducción de los datos de la segunda articulación Hombro con respecto a la trayectoria deseada.

**Fuente:** Autores

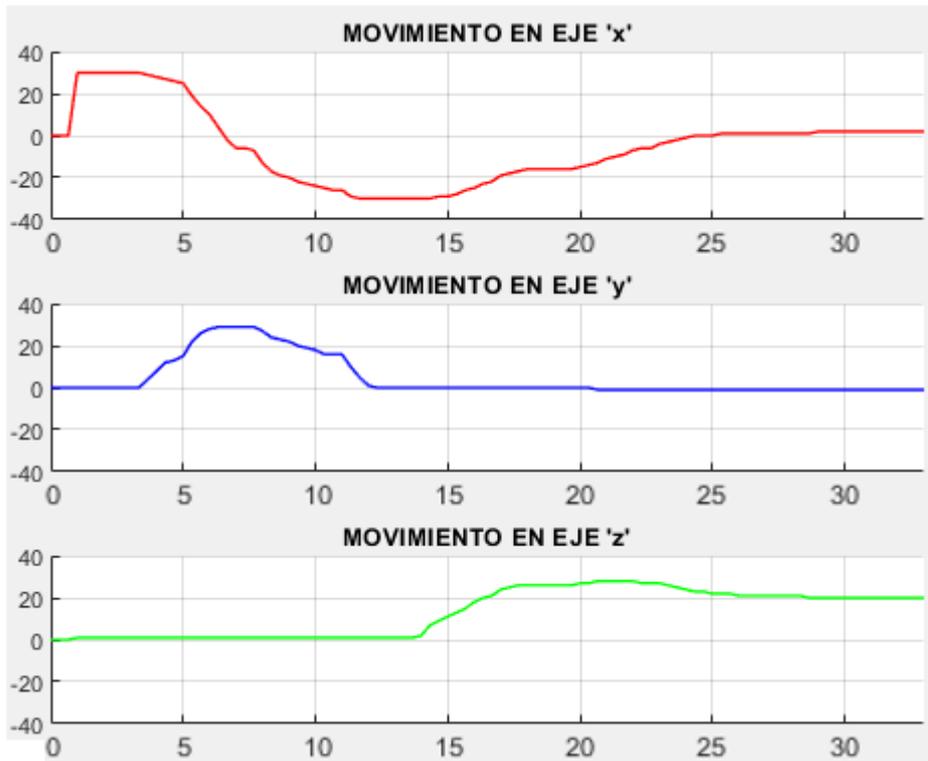
Los datos obtenidos en la reproducción del tercer grado de libertad CODO véase (ilustración 66), muestran que el margen de error en la trayectoria es menor a 0.6 cm como máximo, esto debido a que en la recolección de los datos, algunos valores suelen ser alterados, esto se da por la mala manipulación o enseñanza de cada articulación, se puede considerar a este margen de error como despreciable, obteniendo un resultado exitoso en la trayectoria de reproducción de los datos del Codo.



(a) Brazo robótico



(b) Trayectoria completa deseada del Brazo robótico

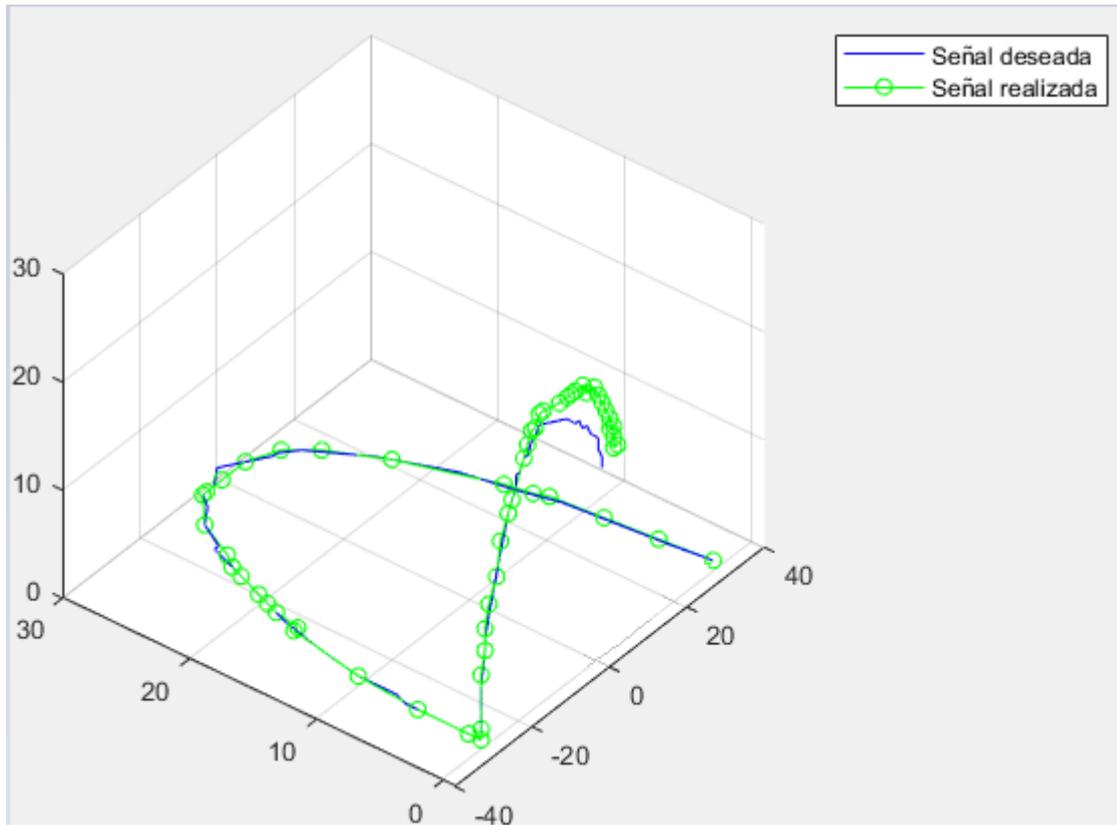


(c) Trayectoria realizada completa representada en cada eje del plano del Brazo robótico

**Ilustración 67.** Trayectoria realizada de la reproducción de los datos de cada grado de libertad del Brazo robótico

*Fuente:* Autores

Los resultados obtenidos en la reproducción de los datos de cada grado de libertad en conjunto del brazo robótico véase (ilustración 67), muestran que la trayectoria realizada por todo el brazo tiene un margen de error, para poder obtener el margen de error de la trayectoria realizada se generó una comparativa entre la trayectoria deseada y la trayectoria realizada en función del tiempo.



**Ilustración 68.** Trayectoria realizada de la reproducción de los datos de cada grado de libertad del Brazo robótico con respecto a la trayectoria deseada.

**Fuente:** Autores

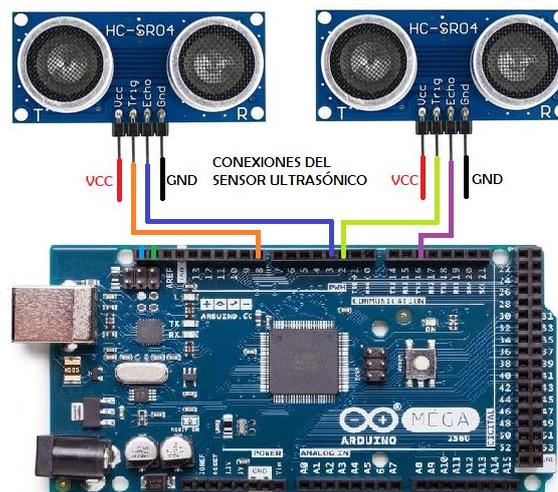
Los datos obtenidos en la reproducción de cada grado de libertad en conjunto del brazo robótico véase (ilustración 68), muestran que el margen de error en la trayectoria es menor a 0.8 cm como máximo, esto debido a que en la recolección de los datos, algunos valores suelen ser alterados, esto se da por la mala manipulación o enseñanza de cada articulación, se puede considerar a este margen de error como despreciable, obteniendo un resultado exitoso en la trayectoria de reproducción de los datos del Brazo robótico.

#### 4.2.4 PRUEBAS PARA LA DETECCIÓN DE OBSTÁCULOS

Existen varios sensores los cuales pueden ayudar a realizar la tarea de detectar un objeto, con el fin de completar esta etapa de aprendizaje y brindar una acción de seguridad. Para estas pruebas se consideró un sensor ultrasónico, un sensor de movimiento PIR y un sensor Infrarrojo de obstáculos.

Al analizar la propuesta de tener un sensor con la finalidad de detectar un obstáculo, esta acción debe ser priorizada, con el fin de pausar o detener la reproducción de enseñanza, por el mismo motivo se empleó interrupciones para su programación.

Como primera prueba se utilizó el sensor ultrasónico, cuyo funcionamiento es el de emitir un sonido y medir el tiempo que tarda en regresar, el mismo da a conocer que tan próximo esta un objeto del sensor en un rango de 20cm, como resultado, el sensor está transmitiendo y recibiendo señales desde que entra en funcionamiento, al estar programado a través de interrupciones, este es llamado en cada nueva lectura, lo cual hace que el sistema no salga de la interrupción y no complete el ciclo de reproducción enseñado (véase ilustración 69).



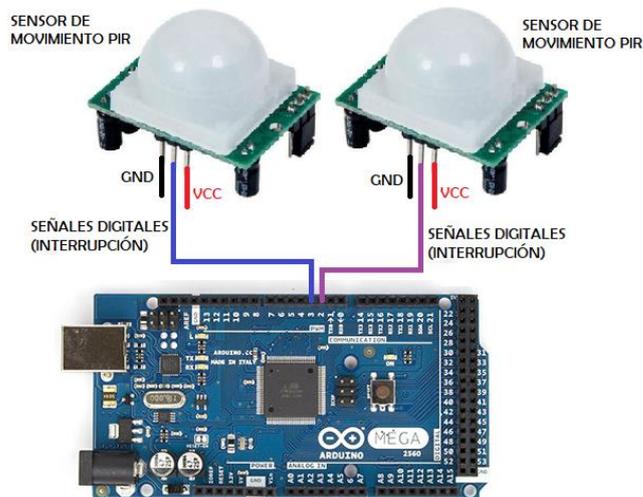
*Ilustración 69. Conexiones del sensor ultrasónico*

*Fuente: Autores*

Luego de los resultados vistos con el primer sensor, se decidió ocupar en esta segunda prueba el sensor de movimiento PIR, el cual va a generar un cambio de estado si existe un movimiento dentro de su campo de visión, como resultado, el sensor PIR al contar con el lente de Fresnel amplía su visión y sensibilidad, ocasionando que, a pesar de que el obstáculo era

retirado, este aun activaba la interrupción al tomar una señal de movimiento en otro rango de visión no directa (véase ilustración 70).

Por lo que se optó retirar el lente de Fresnel para que el cambio de estado por el movimiento realizado se obtenga en una sola dirección, con lo cual la lectura del sensor se perdía al quitar el obstáculo permitiendo que la reproducción de enseñanza siga desde donde este se hallaba.



*Ilustración 70. Conexiones del sensor PIR*

*Fuente: Autores*

Al ser estático el modo de uso del sensor PIR, se vieron dificultades, ya que el brazo robótico va a estar en movimiento, y dicho sensor funciona correctamente cuando se lo coloca en una posición fija, en un lugar específico. Por tal motivo se realizó una tercera prueba con el sensor infrarrojo de obstáculos, el mismo que tiene una buena respuesta y es muy estable incluso con luz ambiente o total oscuridad, se operó el funcionamiento del mismo a una distancia de 2cm, detectando un obstáculo de manera rápida y al retirar el mismo, luego de un tiempo, retoma el aprendizaje enseñado por el usuario.

En la siguiente tabla 10, se presenta una comparativa entre los 3 sensores utilizados para la detección de obstáculos.

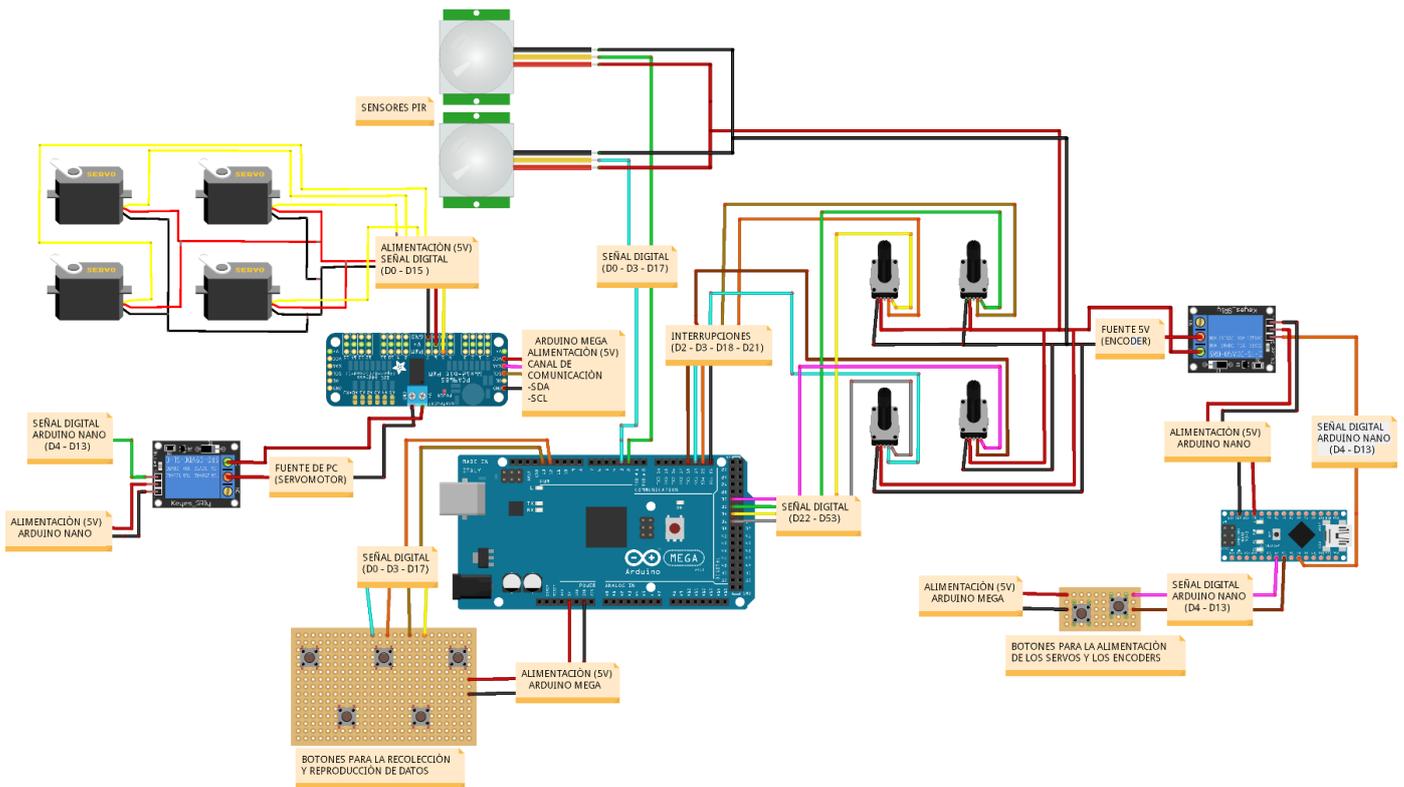
Tabla 10. Tabla comparativa de los sensores de detección

Fuente: Autores

SENSORES PARA LA DETECCIÓN DE OBSTÁCULOS			
SENSORES	ESPECIFICACIONES TÉCNICAS		
	ESTADO DE ACTIVACIÓN	CALIBRACIÓN	DISTANCIA DE OPERACIÓN
ULTRASÓNICO	HIGH	NO ES NECESARIO	2cm ~ 450cm
PIR	HIGH	ENTRE 20 SEG Y 1 MIN	2m ~ 450m
INFRARROJO	LOW	NO ES NECESARIO	2cm ~ 30cm

#### 4.2.5 PRUEBA DE TODA LA ETAPA DE APRENDIZAJE

Considerando cada una de las pruebas en individual del sistema de aprendizaje, este está listo para una prueba en conjunto, teniendo debidamente todas las conexiones realizadas como se muestra a continuación (véase ilustración 71).

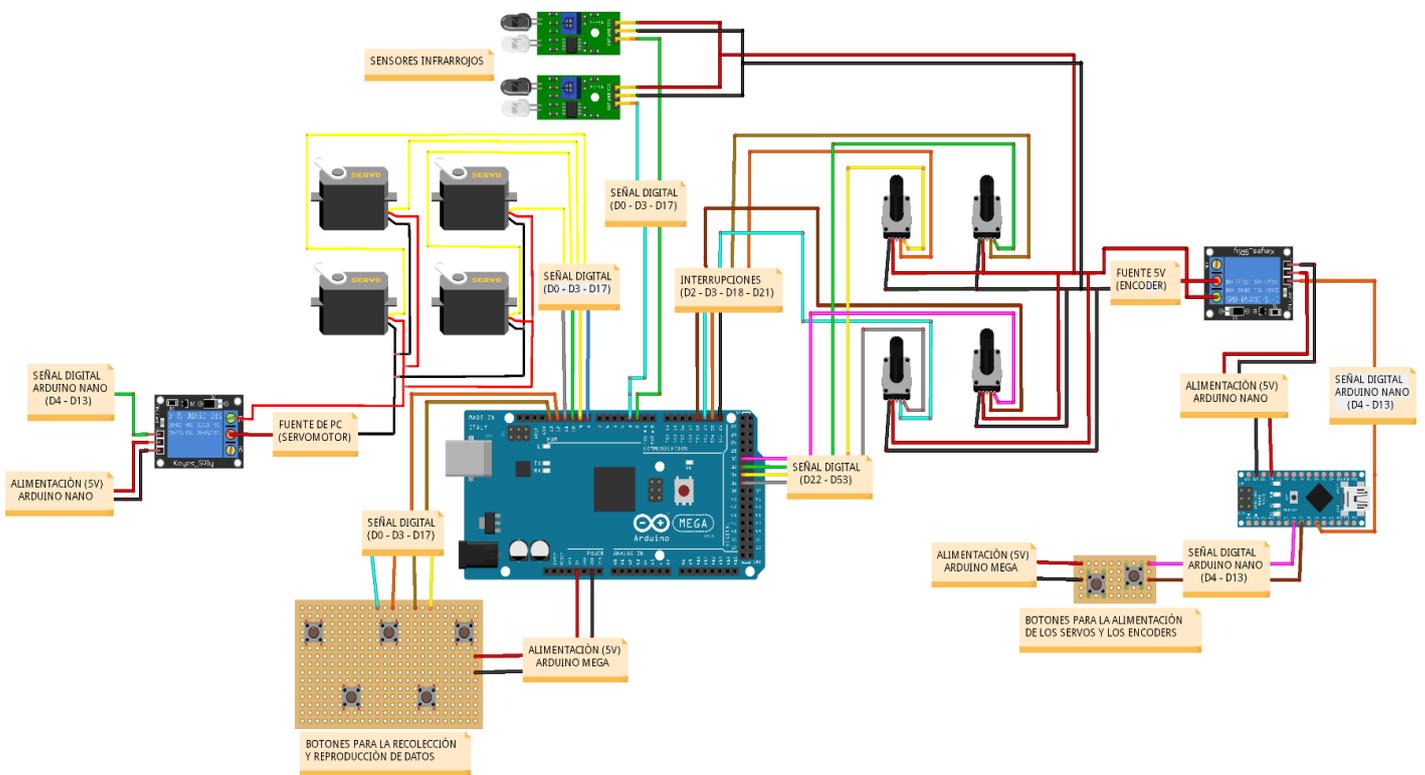


fritzing

Ilustración 71. Conexiones de todas las etapas que conforman el aprendizaje directo

Fuente: Autores

Como resultado a todas las partes conjuntamente conectadas se estableció los valores de alimentación, seguidamente con la etapa de recolección de datos, y una vez completado lo antes mencionado, se procedió a realizar la etapa de reproducción, esto se da al presionar el pulsador para el paso de voltaje hacia los servomotores, y luego de esto, pulsar el botón para visualizar la reproducción de los movimientos enseñados al brazo robótico, fue ahí cuando el mismo dejó de funcionar, revisando las conexiones, todo estaba correctamente conectado, se procedió a la revisión de la programación, al igual estaba bien no se encontró ningún error, por lo que, se optó a realizar pruebas con los pines de interrupción de la placa Arduino Mega con cada uno de los grados de libertad por separado, donde al revisar el datasheet del Arduino Mega, se vio que este comparte la interrupción y las terminales SDA Y SCL respectivamente en los pines 20 y 21, los mismos que permiten la comunicación con el controlador PCA9685 a través de I2C, y al estar siendo utilizada de dos formas diferentes los pines no envían la señal de comunicación haciendo que el sistema entre en estado de interrupción, por lo tanto el brazo robótico no realizaba la tarea enseñada.



*Ilustración 72. Conexión Final de todas las etapas que conforman el aprendizaje directo*

*Fuente: Autores*

Como resultado final a cada prueba realizada (véase ilustración 72), en la etapa de alimentación se obtuvo el paso de voltaje presionando el respectivo pulsador tanto para los servomotores como para los encoders, seguido en la etapa de recolección de datos se realizaban los movimientos con mayor facilidad, y con la respectiva programación los valores se iban guardando en un vector para cada grado de libertad sabiendo que los movimientos de aprendizaje son consecutivos, luego en la etapa de reproducción, se deshabilitaban los encoders y se habilitaban los servomotores para que a través de un pulsador al momento de presionarlo cumpla dicha acción de reproducción sin el llamado a las interrupciones viendo así un aprendizaje fluido, y como etapa adicional que es la detección de obstáculos al utilizar el sensor Infrarrojo se notó una mayor precisión en lo que es, la pausa de los movimientos al estar en funcionamiento detectando un objeto, este realiza una acción para evitar algún accidente y luego de un tiempo retoma el aprendizaje enseñado por el usuario.

### **4.3 RESULTADOS DE REALIZACIÓN DE TAREAS SOBRE AMBIENTE DE TRABAJO EMULADO**

Una vez obtenidos los resultados de teleoperación y de aprendizaje por demostración se procedió a verificar el funcionamiento del brazo robótico Master en un ambiente emulado dándole pruebas al mismo de movimiento en trayectorias con y sin carga.

#### **4.3.1 PRUEBAS DE TRAYECTORIA CON CARGA Y SIN CARGA**

Estas pruebas nos permiten obtener un margen de concordancia entre la trayectoria con carga y la trayectoria sin carga, a fin de obtener una comparativa entre las mismas. Para esto se definió un peso a escala de los adoquines de hormigón los cuales varían su peso como se muestra a continuación.

Para las pruebas se consideró el peso de los adoquines a una escala de 5:1, dando como resultados:

- Adoquín Holandés rectangular de 6cm

Peso real: 2.83kg

$$\begin{array}{ccc}
 5 & \swarrow & \searrow & 1 \\
 2.83kg & & & x
 \end{array}$$

$$x = \frac{2.83kg}{5}$$

$$x = 0.5kg$$

Peso a escala: 0.5kg

- Adoquín Holandés rectangular de 8cm

Peso real: 3.50kg

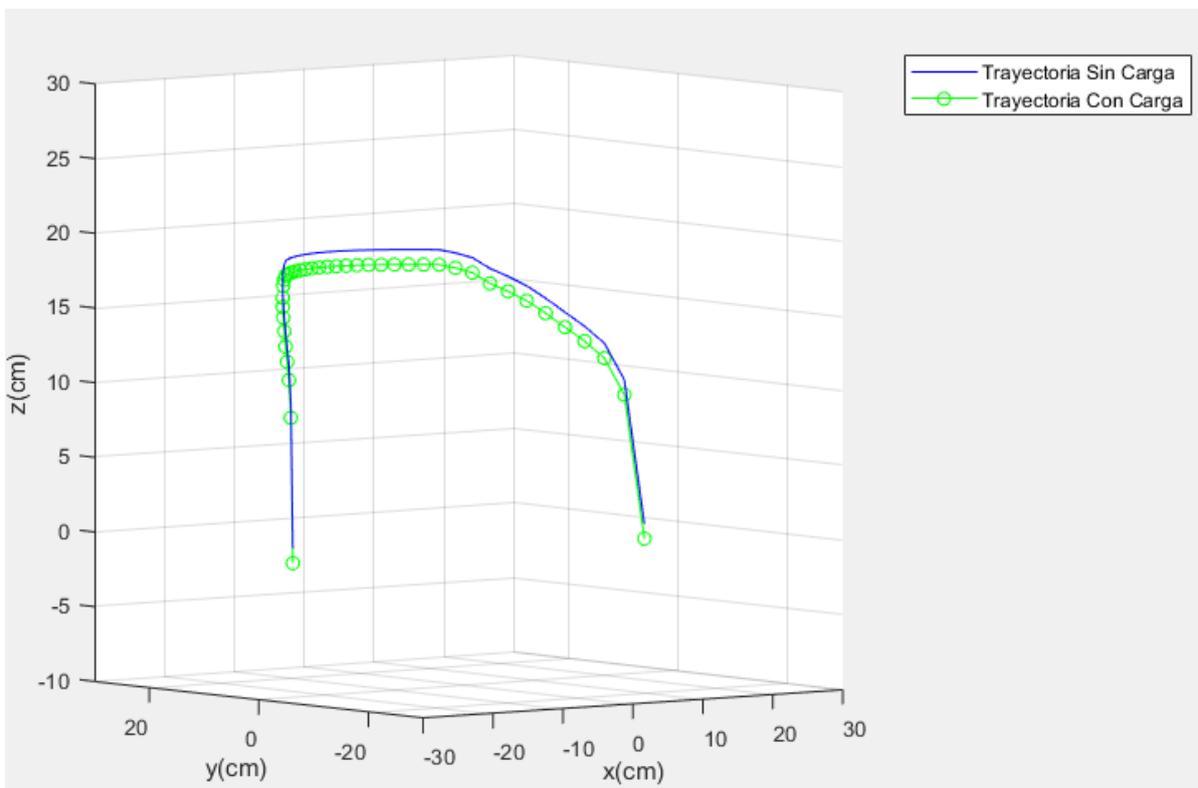
$$\begin{array}{c} 5 \swarrow \quad \searrow 1 \\ 3.50kg \quad x \end{array}$$

$$x = \frac{3.50kg}{5}$$

$$x = 0.7kg$$

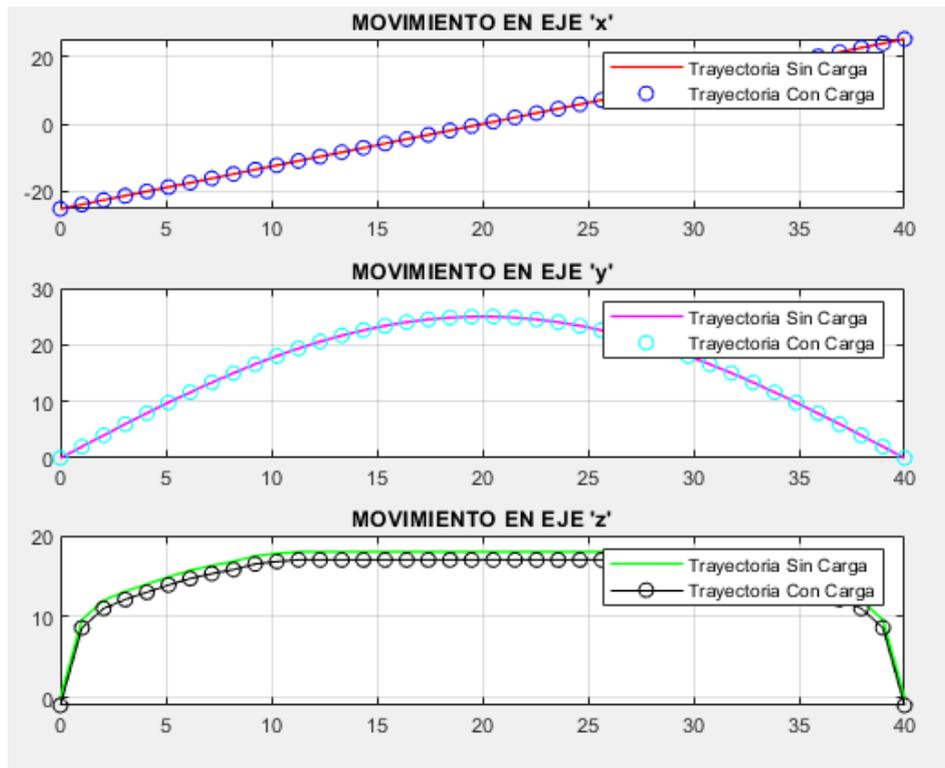
Peso a escala: 0.7kg

La primera prueba realizada se efectuó con un adoquín a escala con un peso de 0.5 kg, en donde se generó una comparativa con una trayectoria con carga y sin carga como se muestra en las siguientes ilustraciones.



**Ilustración 73.** Trayectoria con carga y sin carga con un peso de 0,5 kg

**Fuente:** Autores

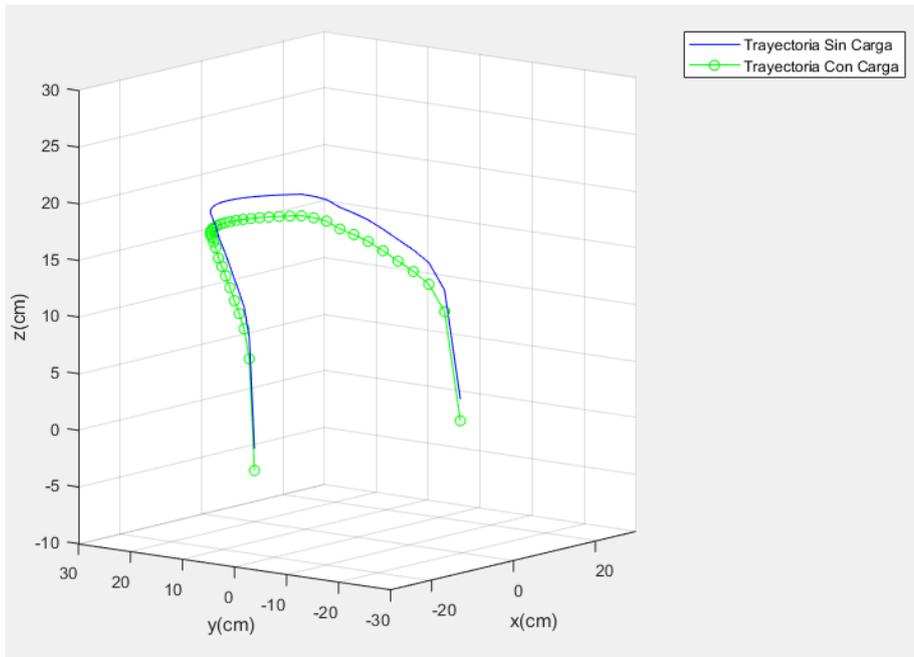


*Ilustración 74. Trayectoria con carga y sin carga representado en cada eje de plano con un peso de 0,5kg*

*Fuente: Autores*

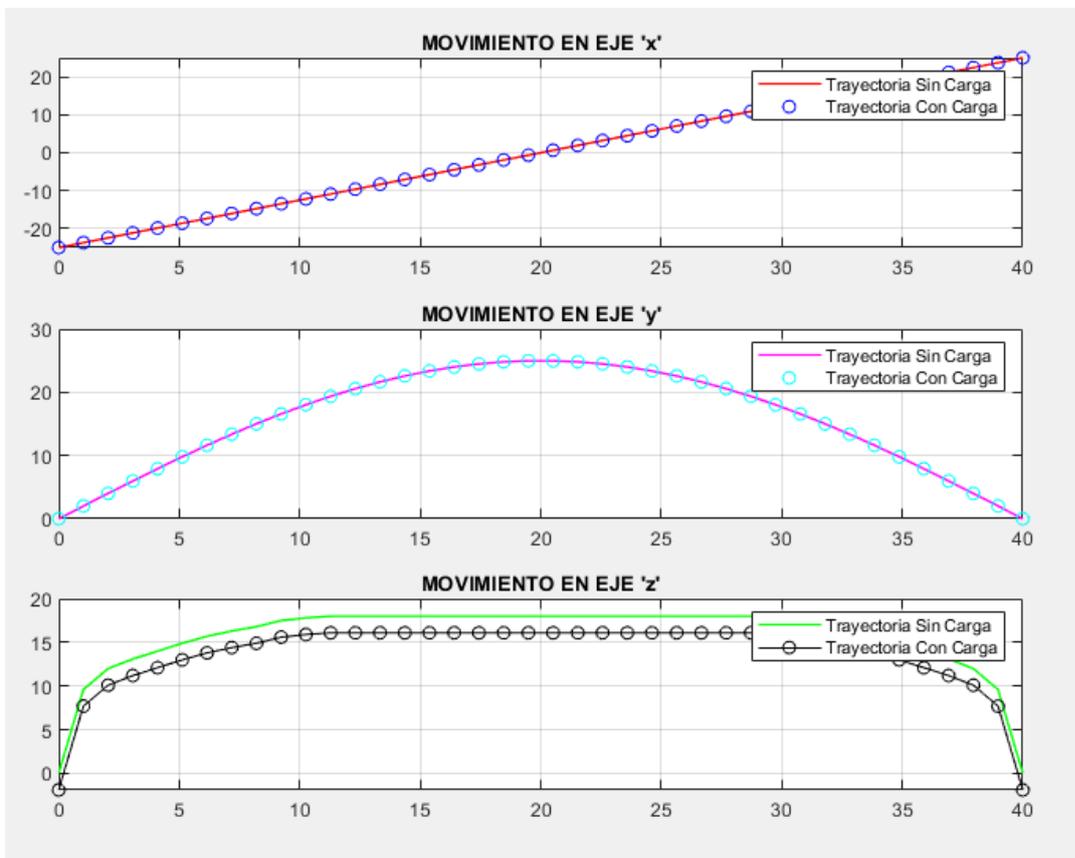
De las pruebas realizadas de la comparativa de la trayectoria con carga y sin carga, se puede obtener que las pruebas realizadas con una carga de 0,5 kg el margen de error presentado es de un máximo de 1.2 cm en el plano z, esto debido a que el efecto de la gravedad sobre el adoquín a escala provoca que las articulaciones por el peso tengan un desfase.

La siguiente prueba realizada se efectuó con un adoquín a escala con un peso de 0.7 kg, en donde se generó la misma comparativa mencionada anteriormente entre una trayectoria con carga y sin carga como se muestra en las siguientes ilustraciones.



**Ilustración 75.** Trayectoria con carga y sin carga con un peso de 0,7kg

**Fuente:** Autores

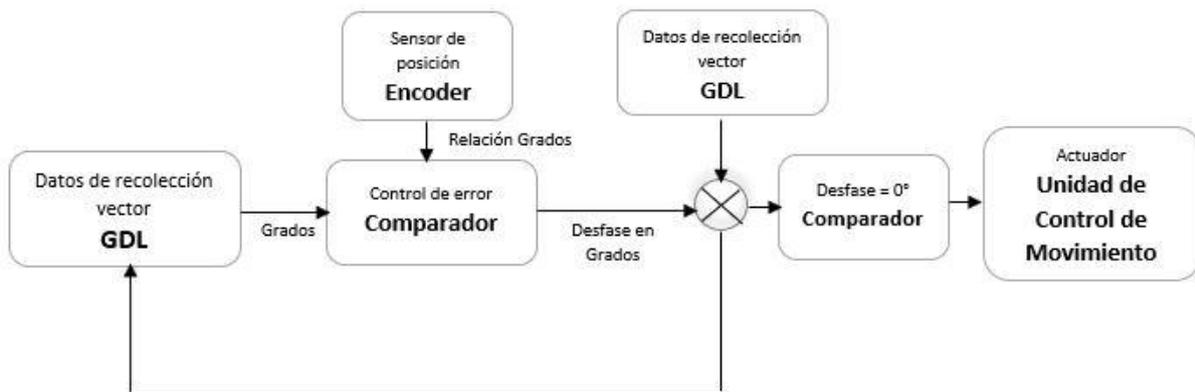


**Ilustración 76.** Trayectoria con carga y si carga con un peso de 0,7kg representada en cada eje del plano

**Fuente:** Autores

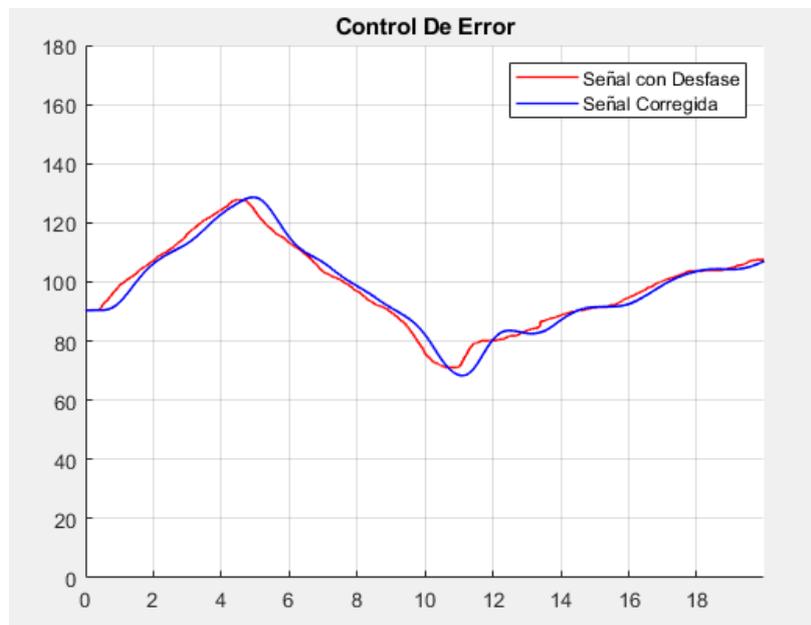
Al realizar el análisis de los resultados de la comparativa de una trayectoria con carga y sin carga para un adoquín de 0,7 kg el margen de error presentado es de un máximo de 1.8 cm en el plano z, el margen de error es superior al anteriormente mencionado debido al mismo efecto de la gravedad sobre el adoquín a escala provoca que las articulaciones por el peso tengan un mayor desfase.

Para poder corregir el inconveniente presentado por los desfases en las trayectorias realizadas con cargas se utilizó un control de error como se muestra en la siguiente ilustración, el cual realiza una comparación entre las posiciones recolectadas y un encoder el cual sensa los valores de posiciones reproducidas, al verificar que tienen un desfase este procede a corregirlas.



**Ilustración 77.** Diagrama en bloques del Control de error

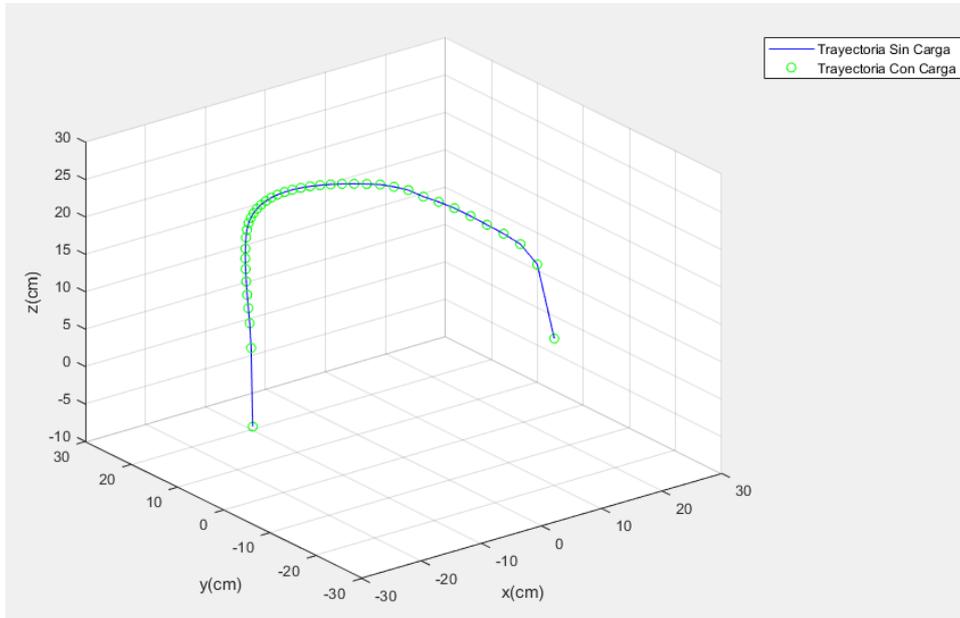
*Fuente: Autores*



**Ilustración 78.** Señal con Desfase corregida por el control de error

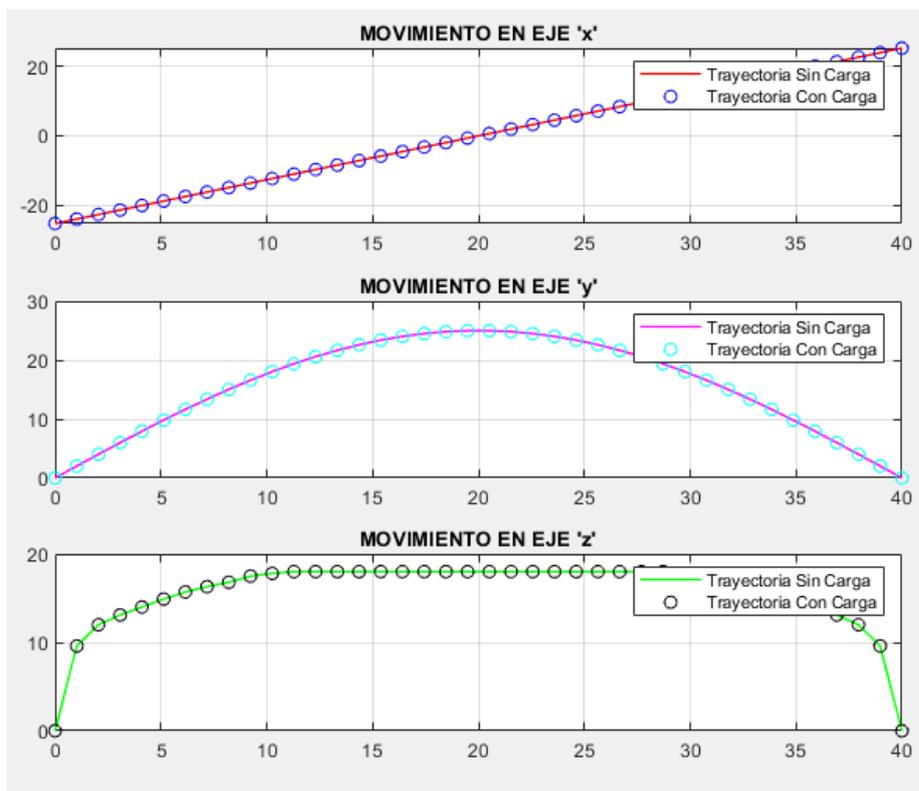
*Fuente: Autores*

De las pruebas generadas se pudo obtener que el control de error utilizado realiza una corrección de desfase, de forma que la reproducción de trayectoria con la carga no presenta ningún desfase como se puede observar en la ilustración siguiente.



**Ilustración 79.** Trayectoria con carga sin desfase

*Fuente:* Autores



**Ilustración 80.** Trayectoria con carga sin desfase presentada en cada eje del plano

*fuentes:* Autores

Una vez verificado el correcto movimiento en trayectoria se procedió a realizar pruebas con diferentes adoquines con el fin de evaluar los resultados que determinan el rango máximo de carga para el brazo robótico, a continuación, se presentan los parámetros de evaluación:

- *Satisfactorio*: Realiza la acción sin ningún sobre esfuerzo en ninguna articulación.
- *Bueno*: Realiza la acción con un sobre esfuerzo mínimo sobre una articulación.
- *Regular*: Realiza la acción con sobre esfuerzo sobre dos o más articulaciones.
- *Malo*: No realiza la acción.

*Tabla II. Cargas de prueba con la pinza*

*Fuente: Autores*

CARGA	PESO	ARTICULACION	INTEGRIDAD PIEZA	RESULTADO
Adoquín a escala 1	0,1kg	Bien	Bien	Satisfactorio
Adoquín a escala 2	0,5kg	Bien	Bien	Satisfactorio
Adoquín a escala 3	0,7kg	Bien	Bien	Satisfactorio
Adoquín a escala 4	0,8kg	Bien	Bien	Satisfactorio
Adoquín a escala 5	1kg	Sobre esfuerzo en la segunda articulación	Bien	Bien

Los resultados obtenidos determinan que el efector final ejecuta la tarea de carga y descarga de forma satisfactoria en los pesos a escalas dispuestos para las pruebas del prototipo de brazo robótico desarrollado en el presente proyecto.

#### **4.3.2 REPETIVIDAD DE TAREAS**

Las pruebas que se realizaron en este apartado con el brazo robótico, sirven para evaluar su funcionalidad y efectividad al momento de ejecutar las tareas de apilamiento: recolectar, trasladar y ubicar cada una de las piezas de adoquín a escala.

El resultado de esta prueba se presenta a continuación en la tabla 10, para esto se realizaron 5 intentos, los cuales son de ayuda para obtener la efectividad del brazo robótico mediante la utilización de la siguiente fórmula:

$$\% \text{ Efectividad} = \frac{\text{Intentos exitosos}}{\text{Número de intentos}} \times 100$$

$$\text{Intentos Fallidos} = \text{Intentos exitosos} - \text{Número de intentos}$$

**Tabla 12.** Pruebas realizadas con el brazo robótico

*Fuente:* Autores

<b>N.º</b>	<b>Descripción de las pruebas</b>	<b>Intentos exitosos</b>	<b>Intentos fallidos</b>	<b>Número de intentos</b>	<b>Efectividad [%]</b>	<b>Repetitividad de tarea</b>
1	Recolectar, trasladar y ubicar adoquín 1 (Trayectoria 1)	4	1	5	80	Infinita
2	Recolectar, trasladar y ubicar adoquín 2 (Trayectoria 1)	4	1	5	80	Infinita
3	Recolectar, trasladar y ubicar adoquín 3 (Trayectoria 1)	5	0	5	100	Infinita
4	Recolectar, trasladar y ubicar adoquín 4 (Trayectoria 1)	5	0	5	100	Infinita
5	Recolectar, trasladar y ubicar adoquín 5 (Trayectoria 1)	5	0	5	100	Infinita
6	Recolectar, trasladar y ubicar adoquín 1 (Trayectoria 2)	4	1	5	80	Infinita
7	Recolectar, trasladar y ubicar adoquín 2 (Trayectoria 2)	5	0	5	100	Infinita

<b>8</b>	Recolectar, trasladar y ubicar adoquín 3 (Trayectoria 2)	5	0	5	100	Infinita
<b>9</b>	Recolectar, trasladar y ubicar adoquín 4 (Trayectoria 2)	5	0	5	100	Infinita
<b>10</b>	Recolectar, trasladar y ubicar adoquín 5 (Trayectoria 2)	4	1	5	80	Infinita

$$\% \text{ Efectividad} = \frac{920}{10}$$

$$\% \text{ Efectividad} = 92\%$$

Mostrados los resultados en la tabla anterior, se puede determinar que el brazo robótico tiene una gran efectividad en las pruebas realizadas con un porcentaje total del 92%. Además, cabe recalcar que mientras los operarios del brazo robótico adquieren más práctica los resultados son cada vez mejores. Así mismo por método de observación se pudo definir qué número de repetitividad de las tareas es infinito hasta cuando el operario así lo decida.

## CAPITULO V

### 5.1 CONCLUSIONES

- El sistema de teleoperación implementado permitió el correcto intercambio de paquetes de información con tiempos de retardo mínimos en la comunicación de la estación local y la estación remota.
- En las aplicaciones del sector industrial de hormigón, un prototipo de brazo robótico con las características que incorpora el desarrollado en esta tesis, será una buena herramienta de apoyo, ya que cuenta con ventajas, tanto en la estabilidad como en su moderada velocidad, además, que el aprendizaje directo puede ser reprogramado, para que cumpla su función de agarrar y soltar objetos o cargas en diferentes trayectorias.
- En el sistema electrónico, el Arduino Mega 2560 procesa correctamente las señales enviadas manualmente y trabaja de una manera correcta con los límites establecidos, lo cual nos permite concluir que, para un sistema de control óptimo, es una muy buena opción al momento de realizar este tipo de proyectos, ya que cumple con lo requerido de una manera eficaz.
- Para la creación del prototipo de brazo robótico, se basó en un diseño perteneciente a un autor en la web, el cual fue punto de partida para llegar al diseño final que cuenta con 4 grados de libertad, el mismo que desempeña la tarea enseñada de manera satisfactoria, a pesar de los cambios realizados.
- La ayuda de un sistema conformado por cámaras IP e interface de visualización para el usuario, permite observar en tiempo real, la manipulación teleoperada del operador en la estación local sobre la estación remota.

## 5.2 RECOMENDACIONES

- Verificar la habilitación de los puertos necesarios con el proveedor de servicio de internet o IPS para el servidor MQTT, para que todos los usuarios que se encuentren ya sean dentro o fuera de la red del servidor, puedan establecer una correcta comunicación.
- Se sugiere utilizar servomotores que tengan un torque superior a 20 kg, para evitar golpes en el brazo robótico, ya que la fuerza que necesita los servomotores en cada grado de libertad es considerable al momento de la reproducción de los movimientos enseñados.
- Para utilizar adecuadamente el brazo robótico y poder visualizar la reproducción de los movimientos enseñados, se debe revisar detenidamente las conexiones antes de energizar el sistema, ya que al tener contacto unos cables con otros, pueden generar cortos y por ende se produce daños a los dispositivos, los mismos que ocasionarían un funcionamiento incorrecto del sistema.
- Se deben colocar cables más flexibles de una manera correcta, ya que, al realizar los movimientos enseñados, tienden a estirarse y pueden ocasionar daños en los dispositivos utilizados para el aprendizaje directo, además que, de acuerdo con los requerimientos previos para la teleoperación se recomienda que el internet tenga un gran ancho de banda y una buena velocidad de transmisión, para una rápida recepción de datos.
- Se recomienda un estudio de diseño de brazos robóticos en caso de que este prototipo sea desarrollado ya para el ambiente real de trabajo.
- El presente proyecto es una base para el desarrollo de aprendizaje por demostración en diferentes campos industriales, con un estudio más detallado que ayude a que el brazo robótico cuente con más estabilidad y precisión en los movimientos, este proyecto sería de ayuda para el desarrollo de brazos robóticos en la industria nuclear y química.
- El presente proyecto es una base para que, en un futuro se desarrolle un sistema más complejo capaz de integrar inteligencia artificial partiendo de la tesis realizada, ya que en el transcurso de la investigación se encontró un ejemplo el cual, concretaba que, el sistema robótico capaz de jugar al tenis de mesa, al final de su brazo sujeta una raqueta y consta de cuatro cámaras, que permiten a la maquina observar su entorno y la pelota que se dirige a ella, dicha máquina ha aprendido

directamente de su creadora Muelling (Velasco, 2012), la cual ha entrenado al brazo robótico moviéndolo con las manos, tomando alrededor de 25 patrones de movimientos básicos y mediante su combinación, el robot es capaz de devolver de manera correcta las pelotas que se lanzan en varios ensayos, todos los movimientos realizados quedaban grabados en una librería de posiciones llamadas “primitivas” en el cerebro del sistema y mediante el dicho popular “la práctica lleva a la perfección”, el robot ha sido capaz de devolver el 88% de las pelotas con éxito después de jugar una hora contra un humano.

## **BIBLIOGRAFÍA**

- Atkeson, C. G., & Schaal, S. (1997). Robot Learning from Demonstration. Proceedings of the Fourteenth International Conference on Machine Learning, (1994), 12–20.
- Calinon, S., & Billard, A. (2007). Active teaching in robot programming by demonstration. Proceedings - IEEE International Workshop on Robot and Human Interactive Communication, 702–707. <https://doi.org/10.1109/ROMAN.2007.4415177>
- Hewitt, A., Yang, C., Li, Y., & Cui, R. (2017). DMP and GMR based teaching by demonstration for a KUKA LBR robot. ICAC 2017 - 2017 23rd IEEE International Conference on Automation and Computing: Addressing Global Challenges through Automation and Computing. <https://doi.org/10.23919/IConAC.2017.8081982>
- Kyranini, M., Haseeb, M. A., Ristić-Durrant, D., & Gräser, A. (2018). Robot learning of industrial assembly task via human demonstrations. *Autonomous Robots*, 1–19. <https://doi.org/10.1007/s10514-018-9725-6>
- Julio Astete Rupay, D. T. (2016). SISTEMA TELEOPERADO APLICADO A LA MANIPULACIÓN DE RESIDUOS HOSPITALARIOS CON RIESGO BIOLÓGICO. *Revista Ingenium* vol. 1 .
- López, G. L., & D. A. (2017). IDEAS DEL ALUMNADO SOBRE ROBÓTICA Y PROGRAMACIÓN EN 3.º DE LA ESO.
- Li, P., & Chu, C. (2016). Augmented Reality based Robot Path Planning for Programming by Demonstration Augmented Reality based Robot Path Planning for Programming by Demonstration. *Design Engineering Workshop*, (December), 1–5.
- Neto, P., Pires, J. N., & Moreira, A. P. (2009). Accelerometer-based control of an industrial robotic arm. Proceedings - IEEE International Workshop on Robot and Human Interactive Communication, 1192–1197. <https://doi.org/10.1109/ROMAN.2009.5326285>
- Road, K., & Synergy, H. H. (2018). Programming of an Industrial Robot Using Demonstrations and Soft Computing Techniques, *77*(March), 156–163.
- Kramberger, A. (2014). A comparison of learning-by-demonstration. 23rd International Conference on Robotics in Alpe-Adria-Danube Region.

- Kuderer, M., Gulati, S., & Burgard, W. (2016). Learning Driving Styles for Autonomous Vehicles from Demonstration. Department of Computer Science at University of Freiburg.
- Lauretti, C., Cordella, F., Guglielmelli, E., & Zollo, L. (2017). Learning by Demonstration for planning activities. IEEE ROBOTICS AND AUTOMATION.
- WANG, H., CHEN, J., Y K LAU, H., & REN, H. (2016). Motion Planning based on Learning from Demonstration. IEEE ROBOTICS AND AUTOMATION.
- Tirado Ruiz, J. J., & Higinio Rubio , A. (Julio de 2012). ESTUDIO DE LA INFLUENCIA DE LOS PARAMETROS DE CONTACTO EN ADAMS: APLICACION A UN MODELO RESTRINGIDO DEL BIPEDO PASIBOT. Leganès, Madrid, España: Universidad Carlos III de Madrid.
- A. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, E. Liang, Inverted autonomous helicopter flight via reinforcement learning, in: International Symposium on Experimental Robotics, 2004
- B. Browning, L. Xu, M. Veloso, Skill acquisition and use for a dynamicallybalancing soccer robot, in: Proceeding of the 19th National Conference on Artificial Intelligence, AAAI'04, 2004.
- P.K. Pook, D.H. Ballard, Recognizing teleoperated manipulations, in: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'93, 1993.
- J.D. Sweeney, R.A. Grupen, A model of shared grasp affordances from demonstration, in: Proceedings of the IEEE-RAS International Conference on Humanoids Robots, 2007.
- J.M. Sabater (2006). Desarrollo de una Interfaz Kinestésica Paralela y Experimentación en Control de Sistemas Hápticos y Teleoperados. PhD thesis, Universidad Miguel Hernández, Elche, Alicante España, Junio 2003. Disponible desde: [Acceso 12 de enero de 2012].
- J. Tafur, C. Peña, C. Garcia, R. Aracil, “Implementación de una Plataforma Experimental para un Sistema de Teleoperación Robótica en Tiempo Real”, Revista Iberoamericana de Sistemas, Cibernética e Informática Sistemas, Cibernética e Informática, 7(1), 69-74, 2010.

- M. Palankar, K. J. De Laurentis, R. Alqasemi, E. Veras, R. Dubey, Y. Arbel, and E. Donchin, "Control of a 9-DoF wheelchair-mounted robotic arm system using a P300 brain computer interface: Initial experiments," 2008 IEEE Int.Conf. Robot. Biomimetics, ROBIO 2008, pp. 348–353, 2008.
- J. Chen, A. Zelinsky, Programming by demonstration: Coping with suboptimal teaching actions, *The International Journal of Robotics Research* 22 (5) (2003) 299–319.
- T. Inamura, M. Inaba, H. Inoue, Acquisition of probabilistic behavior decision model based on the interactive teaching method, in: *Proceedings of the Ninth International Conference on Advanced Robotics, ICAR'99*, 1999.
- W.D. Smart, Making Reinforcement Learning Work on Real Robots. Ph.D. Thesis, Department of Computer Science, Brown University, Providence, RI, 2002.
- J.A. Clouse, On integrating apprentice learning and reinforcement learning. Ph.D. Thesis, University of Massachusetts, Department of Computer Science, 1996.
- R.P.N. Rao, A.P. Shon, A.N. Meltzoff, A bayesian model of imitation in infants and robots, in: K. Dautenhahn, C. Nehaniv (Eds.), *Imitation and Social Learning in Robots, Humans, and Animals: Behavioural, Social and Communicative Dim*
- P. Abbeel, A.Y. Ng, Apprenticeship learning via inverse reinforcement learning, in: *Proceedings of the 21st International Conference on Machine Learning, ICML'04*, 2004.
- S. Chernova, M. Veloso, Multi-thresholded approach to demonstration selection for interactive robot learning, in: *Proceedings of the 3rd ACM/IEEE International Conference on Human–Robot Interaction, HRI'08*, 2008.
- R. Aler, O. Garcia, J.M. Valls, Correcting and improving imitation models of humans for robosoccer agents, *Evolutionary Computation* 3 (2–5) (2005) 2402–2409.
- A.G. Billard, S. Calinon, F. Guenter, Discriminative and adaptive imitation in uni-manual and bi-manual tasks, in: *The Social Mechanisms of Robot Programming by Demonstration*, *Robotics and Autonomous Systems* 54 (5) (2006) 370–384 (special issue).
- C. Breazeal, M. Berlin, A. Brooks, J. Gray, A.L. Thomaz, Using perspective taking to learn from ambiguous demonstrations, in: *The Social Mechanisms of Robot Programming by*

Demonstration, Robotics and Autonomous Systems 54 (5) (2006) 385–393 (special issue).

P.E. Rybski, K. Yoon, J. Stolarz, M.M. Veloso, Interactive robot task training through dialog and demonstration, in: Proceedings of the 2nd ACM/IEEE International Conference on Human–Robot Interactions, HRI'07, 2007.

B. Argall, B. Browning, M. Veloso, Learning from demonstration with the critique of a human teacher, in: Proceedings of the 2nd ACM/IEEE International Conference on Human–Robot Interactions, HRI'07, 2007.

W.D. Smart, Making Reinforcement Learning Work on Real Robots. Ph.D. Thesis, Department of Computer Science, Brown University, Providence, RI, 2002.

Tirado Ruiz, J. J., & Higinio Rubio , A. (Julio de 2012). ESTUDIO DE LA INFLUENCIA DE LOS PARAMETROS DE CONTACTO EN ADAMS: APLICACION A UN MODELO RESTRINGIDO DEL BIPEDO PASIBOT. Leganès, Madrid, España: Universidad Carlos III de Madrid.

Almeida, B. M., & Cabrera, L. C. (2015). Evaluación de los Parámetros que Afectan la Calidad de Servicio.

Augé, A. C. (2018). Demostrador arquitectura publish/subscribe con MQTT.

Breñosa, J. (2016). ESARROLLO DE TECNOLOGIAS DE TELEMANIPULACION CON AL GRADO DE ESCALADO OREINTADAS A LA INTERACCION HOMBRE. ROBOT EN ENTORNORNOS NUCLEARES.

Camacho, R. (n.d.). Infraestructura de eventos para la Internet de las cosas. Universidad de Castilla.

Campero, J. (2016). Infraestructura de telecomunicaciones y TIC.

Carrión, A. (2009). Solución Integral de Conectividad y Telecomunicaciones SICT.

Colado, U. Z. (2003). Robótica Asistida por Teleoperación y Realidad Virtual .

Parra, C. A. (2018). Análisis de rendimiento de protocolos de.

Percy, L. (2019). Desarrollo de una Plataforma en Android para Teleoperación de un Sistema Robótico en Seguridad de Empresas y Almacenes.

Ramos, V. M. (2014). Análisis de latencia de handover en WLANs para aplicaciones en tiempo real.

ADSL zone. (2019, Enero 3). Retrieved from <https://www.testdevelocidad.es/2019/01/03/cual-es-la-latencia-minima-o-adecuada-para-jugar-online-o-hacer-una-video-llamada/>

ABCELECTRONICA. (s.f.). ABCELECTRONICA. Obtenido de <https://www.abcelectronica.net/productos/relés/modulos/>

ALIBABA. (s.f.). ALIBABA. Obtenido de [https://www.alibaba.com/product-detail/TD-8320MG-20kg-Waterproof-Digital-Metal\\_50042767385.html](https://www.alibaba.com/product-detail/TD-8320MG-20kg-Waterproof-Digital-Metal_50042767385.html)

ARDUINO. (07 de Septiembre de 2018). ARDUINO. Obtenido de <https://www.arduino.cc/en/Guide/ArduinoEthernetShield>

ARDUINO. (2020). ARDUINO. Obtenido de <https://www.arduino.cc/en/Guide/Environment>

ARDUINO. (2020). ARDUINO. Obtenido de <https://www.arduino.cc/en/Guide/Introduction>

ARDUINO. (2020). ARDUINO. Obtenido de <https://store.arduino.cc/usa/mega-2560-r3>

ARDUINO. (2020). ARDUINO. Obtenido de <https://store.arduino.cc/usa/arduino-nano>

ARMZTEC. (05 de Enero de 2018). ARMZTEC. Obtenido de <http://www.armztec.com/2016/06/controladorservomotores-pwm-16ch.html>

BARNES, R. (2018). THE MAGPI. Obtenido de <https://magpi.raspberrypi.org/articles/thonny>

BIGTRONICA. (s.f.). BIGTRONICA. Obtenido de <https://www.bigtronica.com/poblado/motores/servomotor/193-servomotor-estandar-de-alto-torque-mg996r.html>

CIM (Manufactura Integrada por computadora). (s.f.). ROBOTICA. Obtenido de ROBOTICA: <https://www.coursehero.com/file/51832749/2-Roboticapdf/>

Digital Guide IONOS. (30 de Julio de 2019). Protocolos de red, la base de la transmisión electrónica de datos. Obtenido de Protocolos de red, la base de la transmisión electrónica de datos: <https://www.ionos.es/digitalguide/servidores/know-how/los-protocolos-de-red-en-la-transmision-de-datos/>

fortzero. (2019). instructables circuits . Obtenido de <https://www.instructables.com/id/4-Axis-Robot-Arm-DIY/#discuss>

Mecafenix, F. (21 de Abril de 2017). Ingeniería Mecafenix . Obtenido de <https://www.ingmecafenix.com/electronica/potenciometro/>

MICROJMP. (s.f.). MICROJMP. Obtenido de <https://www.microjpm.com/products/encoder-giratorio-ky-040/>

NAYLAMPMECHATRONICS. (s.f.). NAYLAMPMECHATRONICS. Obtenido de <https://naylampmechatronics.com/sensores-proximidad/55-modulo-de-deteccion-pir-hc-sr501.html>

RASPBERRY PI. (2020). RASPBERRY PI. Obtenido de <https://www.raspberrypi.org/about/>

RASPBERRY PI. (2020). RASPBERRY PI. Obtenido de <https://www.raspberrypi.org/downloads/>

Web-Robotica.com. (s.f.). módulo sensor de Infrarrojos IR FC-51 para evitar obstáculos. Obtenido de <https://www.web-robotica.com/arduino/conceptos-basicos-arduino/como-usar-el-modulo-sensor-de-infrarrojos-ir-fc-51-para-evitar-obstaculos-con-robot-arduinogenuino>

UNET , «Dep. de Sistemas Eelectrónicos y de Control,» 2015. [En línea]. Available: <http://www.unet.edu.ve/~ielectro/adc3.pdf>.

Futurlec, «ELECTRONIC COMPONENTS,» 2018. [En línea]. Available: <http://www.futurlec.com/ADConv/ADC0809.shtml>.

J. Caballero, «Scribd,» 25 Mayo 2018. [En línea]. Available: <https://es.scribd.com/document/167210496/Funcionamiento-del-convertidor-analogico-digital-ADC0809>.

DIGILENT , «Cmod S6,» Junio 2016. [En línea]. Available: <https://reference.digilentinc.com/reference/programmable-logic/cmod-s6/reference-manual>.

E. Rojas, «Nodo Electrónico,» 1 Enero 2016. [En línea]. Available: <https://nodoelectronico.com/2016/01/01/que-son-los-fpgas-como-funcionan-para-que-sirven-quien-deberia-utilizarlos/>.

Kevinvilla, «Scribd,» 25 Mayo 2018. [En línea]. Available:

<https://es.scribd.com/document/237677507/Aplicaciones-de-La-Modulacion-PWM>.

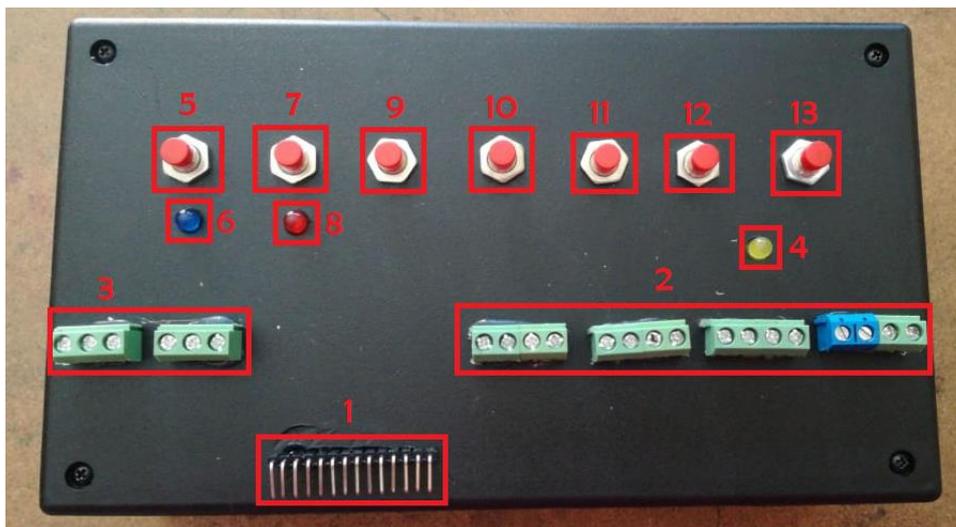
R. F. Coughlin, Julio 2016. [En línea]. Available:

**ANEXOS**

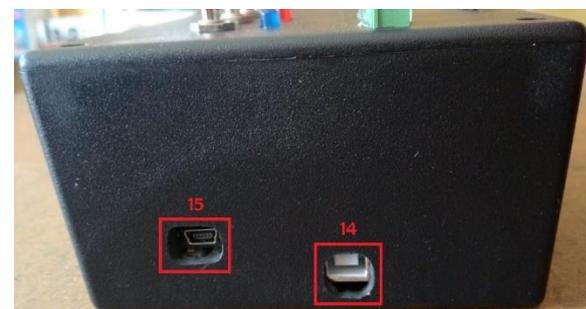
# **ANEXOS**

# MANUAL DE MANEJO PARA EL BRAZO ROBÓTICO

## ANEXO 1 – MANUAL DE MANEJO PARA EL BRAZO ROBÓTICO



1. Pines de conexión para los servomotores.
2. Pines de conexión para los encoders.
3. Pines de conexión para el módulo de detección PIR.
4. Led indicador para detección de obstáculos.
5. Pulsador para funcionamiento de los encoders.
6. Led indicador de encoders.
7. Pulsador para funcionamiento de los servomotores.
8. Led indicador de servomotores.
9. Pulsador para la enseñanza 1.
10. Pulsador para la enseñanza 2.



11. Pulsador para borrar los datos recolectados.
12. Pulsador para dar el tiempo de reproducción.
13. Pulsador para la reproducción del brazo robótico.
14. Conector del Arduino Mega.
15. Conector del Arduino Nano.
16. Conectores para la fuente de PC.



## ANEXO 2 – INTRUCCIONES DEL BRAZO ROBÓTICO

- Se realizan las conexiones de los terminales que posee el brazo robótico de VCC y GND correspondientes a los encoders, servomotores y módulos PIR, después los terminales CLK y DT que corresponde al encoder, y el terminal de SEÑAL que corresponde al servomotor y módulos PIR.
- Para realizar las conexiones tomar en cuenta que, en el caso de los encoders, los conectores van de derecha a izquierda, primero los terminales VCC, segundo los terminales GND, tercero los terminales CLK y cuarto los terminales DT, en el caso de los servomotores, los conectores van de izquierda a derecha, en este apartado primero se conecta la BASE con los pines SEÑAL, VCC y GND, segundo se conecta el HOMBRO, tercero el CODO y por último la PINZA.

- Para el accionamiento del brazo robótico se conectan todos los dispositivos de alimentación.
- Presionar el pulsador 7 para que el brazo robótico se ponga en su posición inicial.
- Desactivar el pulsador 7 (des energiza los servomotores).
- Presionar el pulsador 5 para energizar los encoders y empezar la enseñanza.
- Presionar el pulsador 9 de enseñanza 1 para darle el aprendizaje directo al brazo robótico de forma secuencial, es decir, primero la base, luego el hombro, siguiendo con el codo y por último la pinza.
- Presionar el pulsador 10 de enseñanza 2 para darle el aprendizaje directo al brazo robótico de forma secuencial, es decir, primero el hombro, luego el codo, siguiendo con la base y por último la pinza.
- Presionar el pulsador 12 de tiempo de reproducción (total número de adoquines), que da el número de veces que el brazo robótico repite los movimientos enseñados.
- Desactivar el pulsador 5 (des energiza los encoders).
- Presionar el pulsador 13 de reproducción para observar los movimientos enseñados al brazo robótico.
- Si los valores de la recolección de datos no son los requeridos por el usuario presionar el pulsador 11 para borrar la enseñanza guardada, luego seguir los anteriores pasos.

# INSTALACIÓN EN LA ESTACIÓN REMOTA

## ANEXO 3 – INSTALACIÓN EN LA ESTACIÓN REMOTA

En la estación remota será necesario instalar un complemento llamado Mosquitto en la Raspberry Pi.

- Como primer paso debemos abrir el terminal de comandos en Raspberry.
- En el terminal de comandos utilizamos la siguiente instrucción “apt-get update”

```
root@raspberrypi:~# apt-get update
Get:1 http://mirrordirector.raspbian.org/jessie InRelease [14.9 kB]
Get:2 http://archive.raspberrypi.org/jessie InRelease [22.9 kB]
Get:3 http://repo.mosquitto.org/wheezy InRelease [3.893 B]
Get:4 http://mirrordirector.raspbian.org/jessie/main armhf Packages [9.536 kB]
Get:5 http://repo.mosquitto.org/wheezy/main armhf Packages [2.457 B]
Get:6 http://archive.raspberrypi.org/jessie/main armhf Packages [171 kB]
Ign http://repo.mosquitto.org/wheezy/main Translation-en_GB
Ign http://repo.mosquitto.org/wheezy/main Translation-en
Hit http://archive.raspberrypi.org/jessie/ui armhf Packages
Get:7 http://mirrordirector.raspbian.org/jessie/contrib armhf Packages [43.3 kB]
Ign http://archive.raspberrypi.org/jessie/main Translation-en_GB
Get:8 http://mirrordirector.raspbian.org/jessie/non-free armhf Packages [84.2 kB]
]
```

*Ilustración 81. Función “Update”*

*Fuente: Autores*

- Una vez actualizado la lista de paquetes y versiones procedemos a instalar el bróker Mosquitto

```
root@raspberrypi:~# apt-get install mosquitto
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  apparmor
The following NEW packages will be installed:
  mosquitto
0 upgraded, 1 newly installed, 0 to remove and 267 not upgraded.
Need to get 139 kB of archives.
After this operation, 339 kB of additional disk space will be used.
Get:1 http://repo.mosquitto.org/debian/wheezy/main mosquitto armhf 1.4.14-0mosq
uitto1-nowsl [139 kB]
Fetched 139 kB in 0s (299 kB/s)
Selecting previously unselected package mosquitto.
(Reading database ... 122495 files and directories currently installed.)
Preparing to unpack .../mosquitto_1.4.14-0mosquitto1-nowsl_armhf.deb ...
Unpacking mosquitto (1.4.14-0mosquitto1-nowsl)...
```

*Ilustración 82. Instalación del bróker Mosquitto*

*Fuente: Autores*

- Ahora debemos instalar client MQTT en Raspberry Pi con la siguiente instrucción  
“apt-get install mosquitto-clients”

```
root@raspberrypi:~# apt-get install mosquitto-clients
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libmosquitto1
The following NEW packages will be installed:
  libmosquitto1 mosquitto-clients
0 upgraded, 2 newly installed, 0 to remove and 267 not upgraded.
Need to get 110 kB of archives.
After this operation, 276 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

*Ilustración 83. Instalación del cliente Mosquitto*

*Fuente: Autores*

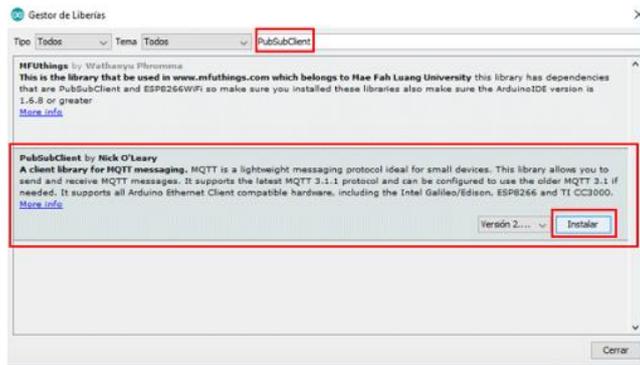
Al realizar estos procesos la estación remota constaría con lo necesario para poder realizar la comunicación MQTT.

## **INSTALACIÓN EN LA ESTACIÓN LOCAL**

### **ANEXO 4 – INSTALACIÓN EN LA ESTACIÓN LOCAL**

Para el funcionamiento de las funciones del servidor es necesario la instalación de la librería PubSubClient la cual es compatible con la placa Arduino con la cual trabaja en el brazo robótico master.

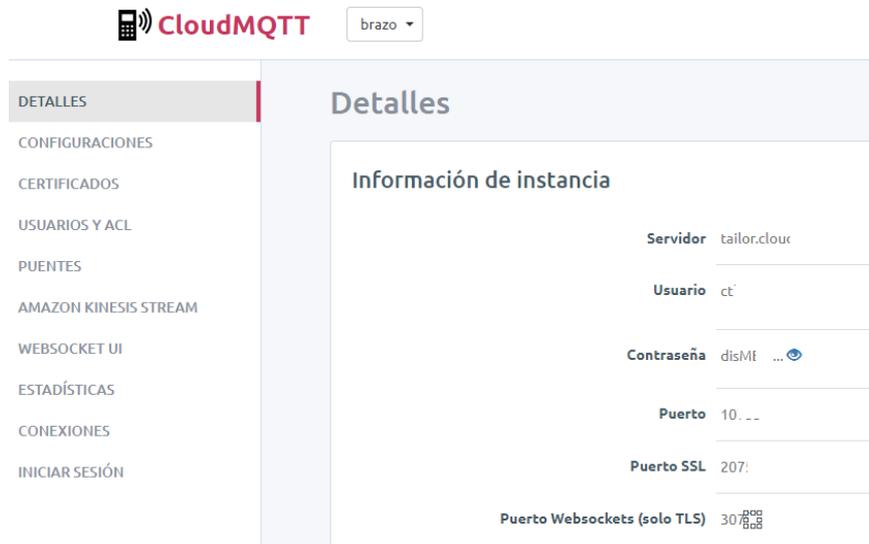
- Instalar se debe abrir el IDLE de Arduino e instalar PubSubClient. Abrir el menú Programa>Incluir Librería>Gestor de librerías y busca PubSubClient.



**Ilustración 84.** Instalación de la librería “PubSubClient”

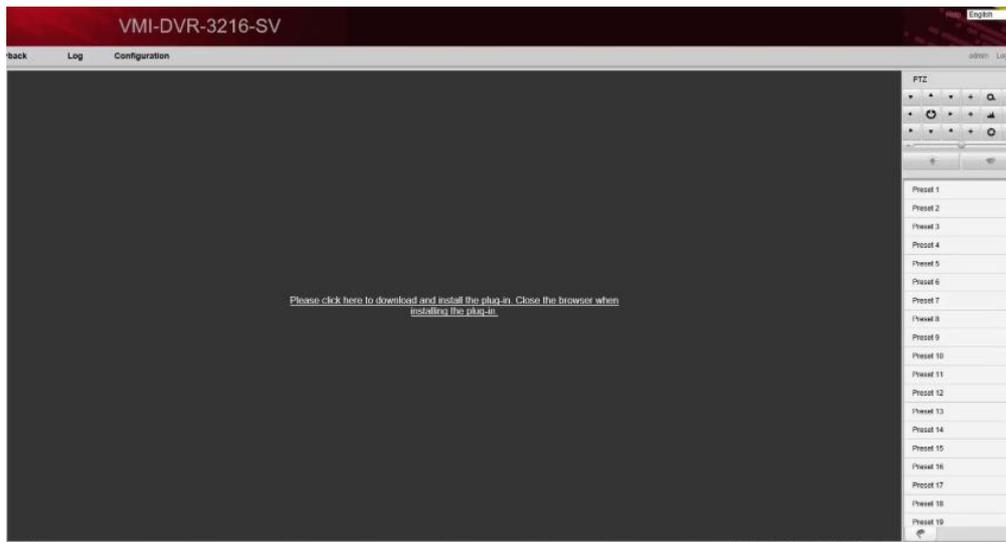
**Fuente:** Autores

Es importante en par en control y supervisión realizar el proceso de instalación del complemento de MQTT Lens en cualquier versión y la IDLE de transmisión de video en tiempo real de las cámaras IP el cual se encuentra en la web oficial de Hikvision, ambas plataformas de supervisión utilizadas en este proyecto son de instalación intuitiva



**Ilustración 85.** Software de supervisión del Bróker MQTT

**Fuente:** Autores



*Ilustración 86. Software de supervisión de video*



## ANEXO 5 – INSTALACIÓN DE RASPBIAN

Para poder realizar todas estas tareas es necesario que esta tarjeta cuente con un sistema operativo, el proceso se detalla a continuación.

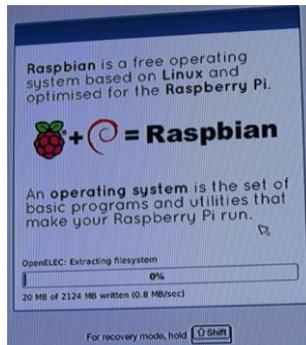
- En una tarjeta SD transferir los archivos del sistema operativo RASPBIAN
- Insertamos la SD en la ranura de la Raspberry Pi y la encendemos seguidamente



*Ilustración 87. Iniciación del sistema operativo sobre Raspberry Pi.*

*Fuente: Autores*

- Al iniciar se ejecutará de forma automática el asistente de instalación del sistema operativo, seguido se escoge la partición de almacenamiento y el lenguaje y se presiona en instalar y hasta que se complete la instalación.



*Ilustración 88. Proceso de instalación del Raspbian.*

*Fuente: Autores*

- Al finalizar la instalación se mostrará un mensaje de éxito en la instalación, se presiona “OK” y se reiniciará de forma automática ya con el sistema operativo listo para ejecutarse.



*Ilustración 89. Finalización de la instalación de Raspbian.*

*Fuente: Autores*



## ANEXO 6 – INSTALACIÓN DEL PROTOCOLO I2C

La instalación de Mosquitto se detalla en el apartado anterior (anexo 5.1), como siguiente punto se explica la instalación y habilitación de Adafruit el componente necesario para la habilitación del protocolo de comunicación I2C entre la Raspberry y el controlador de servomotores PCA 9685.

- Abrir un terminal de comandos en nuestra Raspberry e ingresamos la siguiente línea de código con el fin de agregar e instalar las herramientas necesarias para configuración del protocolo I2C “`sudo apt-get install build-essential python-dev python-smbus i2c-tools python-pip --yes`”
- Después insertamos la siguiente instrucción. “`sudo raspi-config`”

- Finalmente activamos el protocolo con la siguiente instrucción. “`i2cdetect -y 1`” y verificamos que este activado el puerto verificando la salida como se muestra a continuación.

```

pi@raspberrypi (~) - ssh - bash - 4.3 - | sudo i2cdetect -y 1
00:  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  40  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  70  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --

```

*Ilustración 90. Comprobación del protocolo I2C sobre Raspberry.*

*Fuente: Autores*



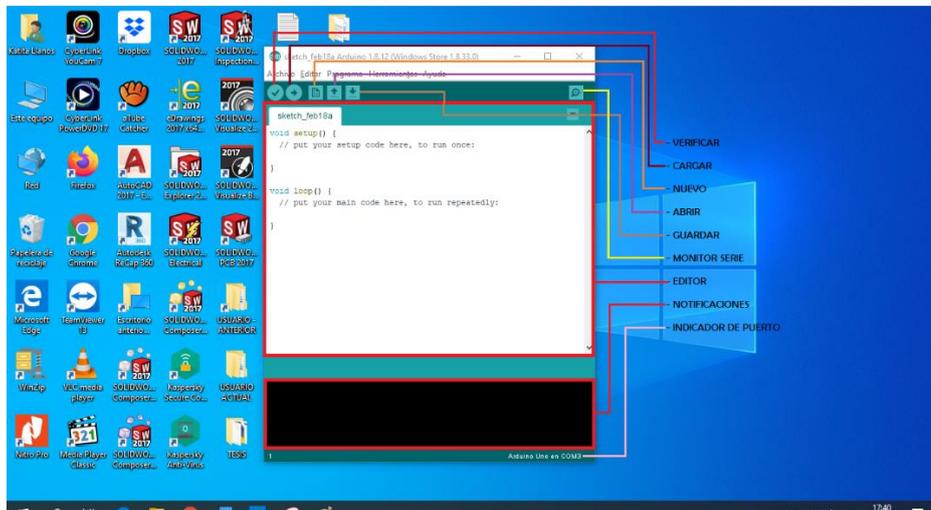
## ANEXO 7 – HERRAMIENTAS Y COMPONENTES

- ARDUINO**

Arduino es una plataforma electrónica de código abierto basada en hardware y software fácil de usar. Las placas Arduino pueden leer entradas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida: activar un motor, encender un LED, publicar algo en línea. Puede decirle a su placa qué hacer enviando un conjunto de instrucciones al microcontrolador de la placa. Para hacerlo, utiliza el lenguaje de programación Arduino (basado en el cableado) y el software Arduino (IDE), basado en el procesamiento. (ARDUINO, 2020)

## ▪ INTERFAZ IDE DE ARDUINO

El Arduino Integrated Development Environment, o Arduino Software (IDE), contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para funciones comunes y una serie de menús. Se conecta al hardware Arduino y Genuino para cargar programas y comunicarse con ellos. (ARDUINO, 2020)



*Ilustración 91. IDE Arduino*

*Fuente: Autores*

## ▪ ARDUINO UNO

Arduino UNO es una placa de microcontrolador de código abierto basado en el microchip Atmega328P, es una plataforma open-hardware de bajo coste, diseñado para un fácil desarrollo de proyectos multidisciplinarios, el cual permite tomar los valores lineales obtenidos por los potenciómetros, procesarlos y adecuarlos para ser enviados al servidor web a través de un enlace de red.



*Ilustración 92. Arduino UNO*

*Fuente: (ARDUINO, 2019)*

- **ARDUINO MEGA 2560**

El MEGA 2560 está diseñado para proyectos más complejos. Con 54 pines de E / S digitales, 16 entradas analógicas y un espacio más grande para su boceto, es la placa recomendada para impresoras 3D y proyectos de robótica. (ARDUINO, 2020)

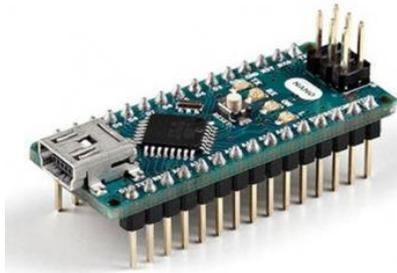


*Ilustración 93. Arduino MEGA 2560*

*Fuente: (ARDUINO, 2020)*

- **ARDUINO NANO**

El Arduino Nano es un tablero pequeño, completo y amigable basado en el ATmega328P. Tiene más o menos la misma funcionalidad que Arduino UNO, pero en un paquete diferente. Carece solo de un conector de alimentación de CC y funciona con un cable USB Mini-B en lugar de uno estándar. (ARDUINO, 2020)



*Ilustración 94. Arduino NANO*

*Fuente: (ARDUINO, 2020)*

- **RASPBERRY PI**

Raspberry Pi es una placa integrada de bajo coste que incluye en su totalidad los elementos que contiene una computadora. En la actualidad este tipo de tarjetas son de gran importancia por la diversidad de proyectos que es capaz de realizar en áreas tecnológicas como redes informáticas, circuitos electrónicos robótica, domótica, programación, entre otras. (RASPBERRY PI, 2020)

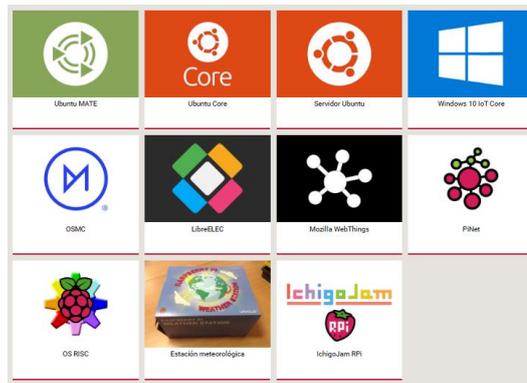


*Ilustración 95. RASPBERRY PI 3*

*Fuente: (RASPBERRY PI, 2020)*

- **SISTEMA OPERATIVO DE RASPBERRY**

La fundación Raspberry Pi recomienda utilizar el sistema operativo Raspbian, ya que esta optimizado para el hardware de la placa, sin embargo, existen diferentes plataformas de terceros.

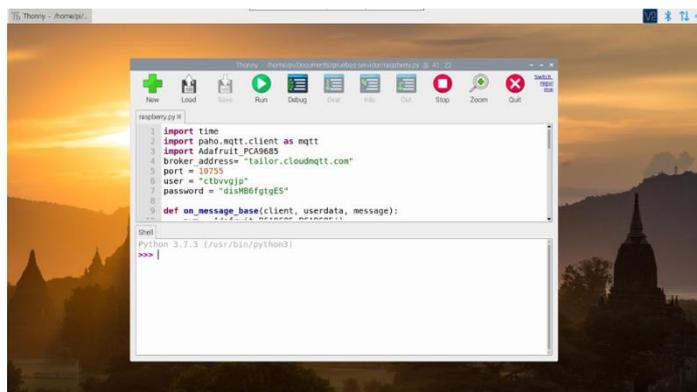


*Ilustración 96. Sistemas Operativos de terceros*

*Fuente: (RASPBerry PI, 2020)*

## ▪ THONNY PYTHON

Thonny es un nuevo IDE (entorno de desarrollo integrado) incluido con la última versión del sistema operativo Raspbian con PIXEL. Usando Thonny, ahora es mucho más fácil aprender a codificar. Thonny viene con Python 3.6 integrado, por lo que no necesita instalar nada. (BARNES, 2018)



*Ilustración 97. IDE Thonny de Python*

*Fuente: Autores*

## MÓDULOS O COMPONENTES ELECTRÓNICOS

### ▪ POTENCIOMETRO

Un potenciómetro es un dispositivo que nos permite modificar el valor de resistencia deslizando una perilla rotatoria, esta función permitirá que, al variar el valor óhmico, cambien los valores de PWM de los servomotores que conforman el prototipo de brazo robótico master, permitiendo así el movimiento del mismo. (Mecafenix, 2017)

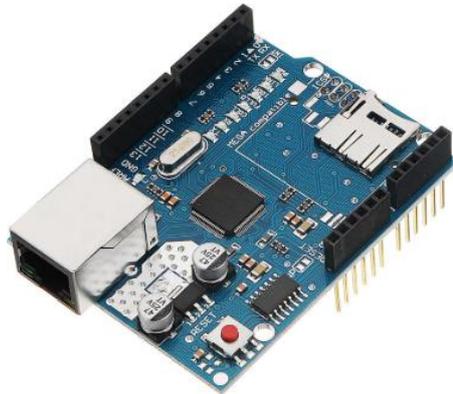


*Ilustración 98. Potenciómetro*

*Fuente: (Mecafenix, 2017)*

#### ▪ SHIELD ETHERNET

El Arduino Ethernet Shield permite a una placa Arduino conectarse a internet, el shield provee un conector ethernet estándar RJ45 y un conector lector de tarjeta Micro SD, este dispositivo nos permite crear un enlace de red entre el Arduino y el servidor web, para poder transmitir o enviar los datos obtenidos por el potenciómetro. (ARDUINO, 2018)



*Ilustración 99. Arduino Shield Ethernet*

*Fuente: (ARDUINO, 2018)*

#### ▪ SERVOMOTOR

Este elemento nos permitirá el control del movimiento del prototipo de brazo robótico, para esto se ha utilizado dos tipos de servomotores el MG996R y el TD-8320MG este último fue utilizado en la articulación del hombro debido a que esta, soporta el peso tanto de articulación del codo como de la muñeca.



**Ilustración 100.** Servomotores a) MG996R y b) TD-8320MG

**Fuente:** a) (BIGTRONICA, s.f.) b) (ALIBABA, s.f.)

**Tabla 13.** Especificaciones técnicas del servomotor 996R

**Fuente:** Datasheet

<b>Peso</b>	55 g
<b>Dimensiones</b>	40.7 x 19.7 x 42.9 mm aprox.
<b>Par de parada</b>	9.4 kgf·cm (4.8 V ), 11 kgf·cm (6 V)
<b>Velocidad de operación</b>	0.17 s/60° (4.8 V), 0.14 s/60° (6 V)
<b>Voltaje de operación</b>	4.8 V a 7.2 V
<b>Corriente de operación</b>	500 mA - 900 mA (6V)
<b>Corriente de parada</b>	2.5 A (6V)
<b>Periodo de trabajo</b>	5 μs

**Tabla 14.** Especificaciones técnicas del servomotor TD-8320

**Fuente:** Datasheet

<b>Peso</b>	60 g
<b>Dimensiones</b>	40.6 x 20.4 x 39 mm aprox.
<b>Par de parada</b>	17.2 kgf·cm (4.8 V ), 20.5 kgf·cm (6 V)
<b>Velocidad de operación</b>	0.19 s/60° (4.8 V), 0.17 s/60° (6 V)
<b>Voltaje de operación</b>	4.8 V a 6.6 V
<b>Periodo de trabajo</b>	1520 μs / 333Hz

▪ **ENCODER**

El módulo KY – 040, es un encoder rotatorio que tiene su perilla giratoria, la misma que proporciona información de cuanto ha sido girada y en qué dirección se está girando. La cual se utiliza para controlar dispositivos como motores paso a paso y servomotores. (MICROJMP, s.f.)

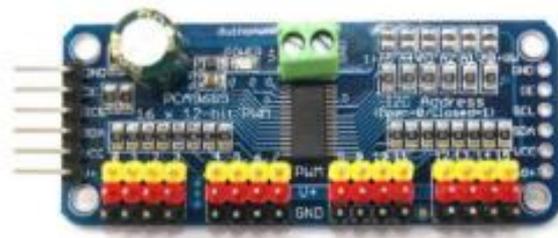


*Ilustración 101. Módulo KY -040*

*Fuente: (MICROJMP, s.f.)*

- **MÓDULO PCA 9685**

El módulo PCA9685 es una placa electrónica que permiten generar 16 canales de salidas PWM con resolución de 12 bits y únicamente un canal de comunicación mediante I2C a un micro-controlador. (ARMZTEC, 2018)



*Ilustración 102. Módulo PCA 9685*

*Fuente: (ARMZTEC, 2018)*

- **SENSOR DE MOVIMIENTO PIR**

Los sensores PIR tienen como función detectar movimiento o cambios de radiación infrarroja en todo objeto (o cuerpo humano), además no se desgastan. Normalmente los podemos encontrar en electrodomésticos y gadgets para la oficina o el hogar. Son conocidos como PIR, “Sensores Infrarrojos” o “Sensores de movimiento”. (NAYLAMPMECHATRONICS, s.f.)

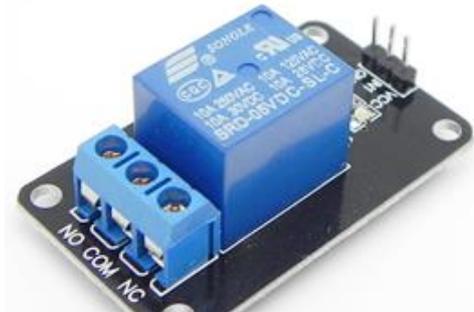


*Ilustración 103. Sensor de movimiento PIR*

*Fuente: (NAYLAMPMECHATRONICS, s.f.)*

#### ▪ **MÓDULO RELÉ**

El módulo relé cuenta con 3 pines que son positivo, señal y negativo, este se conecta directamente a un microcontrolador o también a un circuito integrado, su voltaje de alimentación puede ser de 5VDC o de 12VDC, soporta una corriente de hasta 10A y con este tipo de relé se activa una señal de 0VDC, es decir, un 0 lógico. (ABCELECTRONICA, s.f.)

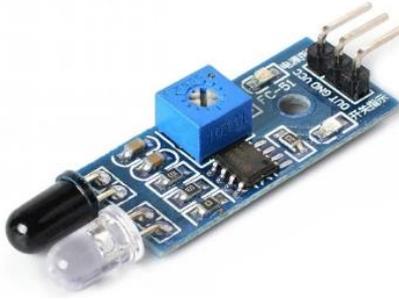


*Ilustración 104. Módulo Relé*

*Fuente: (ABCELECTRONICA, s.f.)*

#### ▪ **SENSOR INFRARROJO**

El sensor infrarrojo está compuesto por un transmisor que emite energía de infrarrojos y un receptor que detecta la energía IR reflejada por la presencia de algún obstáculo, se lo utiliza principalmente para evitar obstáculos en los robots. Su activación depende de la configuración del usuario, el mismo que podrá establecer niveles altos (1 lógico) y bajos (0 lógico). (Web-Robotica.com, s.f.)



*Ilustración 105. Sensor Infrarrojo de detección de obstáculos*

*Fuente: (Web-Robotica.com, s.f.)*