



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

DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO  
DE INGENIERO MECATRÓNICO

TEMA: “DISEÑO E IMPLEMENTACIÓN DE UN ROBOT  
ANTROPOMÓRFICO DE SEIS GRADOS DE LIBERTAD OPEN SOURCE  
(OSH-OSS)”

AUTOR:

GALARZA PANIMBOZA, JUAN DANIEL

DIRECTOR:

ING. ESCOBAR CARVAJAL, LUIS FERNANDO

SANGOLQUÍ

2019



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

**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA**

**CARRERA DE INGENIERÍA MECATRÓNICA**

**CERTIFICACIÓN**

Certifico que el trabajo de titulación, **“DISEÑO E IMPLEMENTACIÓN DE UN ROBOT ANTROPOMÓRFICO DE SEIS GRADOS DE LIBERTAD OPEN SOURCE (OSH-OSS)”** fue realizado por el señor **GALARZA PANIMBOZA, JUAN DANIEL** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 01 de febrero de 2019

Una firma manuscrita en tinta azul sobre una línea horizontal.

**ING. LUIS FERNANDO ESCOBAR CARVAJAL**

**C.C.: 1002403200**



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

**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA**

**CARRERA DE INGENIERÍA MECATRÓNICA**

**AUTORÍA DE RESPONSABILIDAD**

Yo, **GALARZA PANIMBOZA, JUAN DANIEL**, declaro que el contenido, ideas y criterios del trabajo de titulación: **“DISEÑO E IMPLEMENTACIÓN DE UN ROBOT ANTROPOMÓRFICO DE SEIS GRADOS DE LIBERTAD OPEN SOURCE (OSH-OSS)”** es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 01 de febrero de 2019

Una firma manuscrita en tinta azul que parece decir 'Galarza J.' sobre una línea horizontal.

**JUAN DANIEL GALARZA PANIMBOZA**

**C.C.: 1804417234**



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

**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA**

**CARRERA DE INGENIERÍA MECATRÓNICA**

**AUTORIZACIÓN**

Yo, **GALARZA PANIMBOZA, JUAN DANIEL**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“DISEÑO E IMPLEMENTACIÓN DE UN ROBOT ANTROPOMÓRFICO DE SEIS GRADOS DE LIBERTAD OPEN SOURCE (OSH-OSS)”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 01 de febrero de 2019

A handwritten signature in blue ink, appearing to read 'Galarza J.D.', written over a horizontal line.

**JUAN DANIEL GALARZA PANIMBOZA**

**C.C.: 1804417234**

## **DEDICATORIA**

A mi familia.

**Juan Daniel Galarza Panimboza**

## **AGRADECIMIENTO**

Agradezco a todo los que colaboraron para que este sueño se convierta en una realidad A mi familia, amigos, docentes, cada uno de los cuales aportaron para alcanzar el primer paso dentro de mi carrera profesional.

Mi más sincero agradecimiento al Ing. Luis Escobar por su guía, consejos y paciencia para alcanzar el éxito del proyecto, y al Ing. David Loza por cultivar en sus alumnos una pasión para alcanzar objetivos que parecen inalcanzables.

**Juan Daniel Galarza Panimboza**

## ÍNDICE DE CONTENIDOS

CERTIFICACIÓN .....	i
AUTORÍA DE RESPONSABILIDAD.....	ii
AUTORIZACIÓN.....	iii
DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
ÍNDICE DE CONTENIDOS .....	vi
ÍNDICE DE TABLAS .....	x
ÍNDICE DE FIGURAS .....	xi
RESUMEN.....	xv
ABSTRACT .....	xvi
CAPÍTULO I.....	1
GENERALIDADES.....	1
1.1 Antecedentes.....	1
1.2 Justificación e Importancia .....	2
1.3 Alcance del proyecto .....	3
1.4 Objetivos.....	5
1.4.1 Objetivo general.....	5
1.4.2 Objetivos específicos.....	5

CAPÍTULO II .....	6
ESTADO DEL ARTE .....	6
2.1 Manipuladores robóticos industriales .....	6
2.1.1 Componentes de un robot .....	6
2.1.2 Características de un robot.....	8
2.1.3 Grados de libertad de un robot .....	9
2.1.4 Espacio de trabajo.....	11
2.1.5 Configuraciones de robots industriales.....	13
2.1.6 Aplicaciones de robots industriales.....	15
2.1.7 Cinemática de robots manipuladores.....	17
2.1.8 Representación de Denavit – Hartenberg para cinemática directa de robots .....	18
2.1.9 Las descripciones de tarea - Enseñanza y Programación .....	18
2.1.10 Integración del sistema.....	20
2.2 Definición de código libre (Open Source) .....	21
2.3 Open Source Software (OSS).....	23
2.4 Open Source Hardware (OSH).....	24
CAPÍTULO III .....	28
DISEÑO MECATRÓNICO .....	28
3.1 Metodología.....	28

3.2	Diseño del subsistema mecánico.....	29
3.2.1	Requerimientos .....	29
3.2.2	Diseño .....	31
3.2.3	Implementación.....	57
3.3	Diseño del subsistema eléctrico y/o electrónico.....	63
3.3.1	Requerimientos .....	63
3.3.2	Diseño .....	65
3.3.3	Implementación.....	69
3.4	Diseño del subsistema TICs.....	70
3.4.1	Requerimientos .....	70
3.4.2	Diseño .....	72
CAPÍTULO IV .....		80
PRUEBAS Y RESULTADOS .....		80
4.1	Desarrollo del protocolo de pruebas .....	80
4.1.1	Prueba de precisión.....	80
4.1.2	Prueba de repetitividad.....	82
4.2	Resultados.....	83
4.3	Análisis financiero .....	86
CAPÍTULO V .....		87

CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO.....	87
5.1 Conclusiones.....	87
5.2 Recomendaciones .....	87
5.3 Trabajo futuro.....	88
BIBLIOGRAFÍA.....	89
ANEXOS.....	94
A. ALGORITMOS DE PROGRAMACIÓN .....	94
B. PLANOS MECÁNICOS.....	103

## ÍNDICE DE TABLAS

<b>Tabla 1</b> <i>Número de grados de libertad para articulaciones comunes</i> .....	12
<b>Tabla 2</b> <i>Diseños de robots específicos de la aplicación y sus principales requisitos funcionales</i>	16
<b>Tabla 3</b> <i>Niveles abstracción en la programación de robots</i> .....	19
<b>Tabla 4</b> <i>Etapas de la integración del sistema</i> .....	20
<b>Tabla 5</b> <i>Subsistema mecánico. Identificación de necesidades</i> .....	29
<b>Tabla 6</b> <i>Lista de métricas para el subsistema mecánico</i> .....	30
<b>Tabla 7</b> <i>Matriz de necesidades – métricas. Subsistema mecánico</i> .....	31
<b>Tabla 8</b> <i>Robot antropomórfico de 6 grados de libertad. Parámetros de Denavit – Hartenberg</i> ..	34
<b>Tabla 9</b> <i>Propiedades físicas y geométricas de los eslabones del manipulador robótico</i> .....	48
<b>Tabla 10</b> <i>Resultados de torque máximo</i> .....	51
<b>Tabla 11</b> <i>Ponderación de criterios de diseño para la selección de actuadores</i> .....	51
<b>Tabla 12</b> <i>Conceptos para la selección de actuadores</i> .....	51
<b>Tabla 13</b> <i>Matriz de decisión para la selección de actuadores</i> .....	52
<b>Tabla 14</b> <i>Evolución de diseño mecánico</i> .....	54
<b>Tabla 15</b> <i>Dimensiones de poleas dentadas</i> .....	57
<b>Tabla 16</b> <i>Prusa I3 X. Especificaciones técnicas</i> .....	58
<b>Tabla 17</b> <i>Parámetros de impresión 3D</i> .....	58
<b>Tabla 18</b> <i>Tiempos de impresión</i> .....	58
<b>Tabla 19</b> <i>Subsistema electrónico. Identificación de necesidades</i> .....	63
<b>Tabla 20</b> <i>Lista de métricas para el subsistema electrónico</i> .....	64
<b>Tabla 21</b> <i>Matriz de necesidades – métricas. Subsistema electrónico</i> .....	64

<b>Tabla 22</b> Selección de servomotores .....	66
<b>Tabla 23</b> Lista de componentes electrónicos.....	67
<b>Tabla 24</b> Subsistema TICs. Identificación de necesidades .....	71
<b>Tabla 25</b> Lista de métricas para el subsistema TICs.....	71
<b>Tabla 26</b> Matriz de necesidades – métricas. Subsistema TICs.....	72
<b>Tabla 27</b> Datos de posición y orientación.....	84
<b>Tabla 28</b> Resultados de precisión.....	85
<b>Tabla 29</b> Resultados de repetitividad .....	85
<b>Tabla 30</b> Análisis financiero.....	86

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Niryo One. Robot open source de 6 grados de libertad .....	1
<b>Figura 2.</b> RBX1. Brazo robótico de 6 grados de libertad impreso en 3D .....	2
<b>Figura 3.</b> Modelo de sistema de un robot.....	8
<b>Figura 4.</b> Manipulador de 6 DOF con desacople cinemático.....	10
<b>Figura 5.</b> Robot SCARA RRRP para operaciones “pick and place” .....	11
<b>Figura 6.</b> Espacios de trabajo típicos para configuraciones comunes .....	13
<b>Figura 7.</b> Configuraciones típicas de brazo y muñeca de robots industriales .....	14
<b>Figura 8.</b> El problema cinemático .....	17
<b>Figura 9.</b> Norma VDI 2206. Modelo V como macro-ciclo.....	28
<b>Figura 10.</b> Proceso de desarrollo de productos .....	29
<b>Figura 11.</b> Componentes del subsistema mecánico.....	32

<b>Figura 12.</b> Robot antropomórfico de 6 grados de libertad. Representación de ejes de coordenadas .....	33
<b>Figura 13.</b> Desacople cinemático .....	37
<b>Figura 14.</b> Representación de los 3 primeros GDL en el espacio .....	37
<b>Figura 15.</b> Proyección del centro de la muñeca en el plano x-y.....	38
<b>Figura 16.</b> Proyección en el plano formado por los eslabones 2 y 3.....	39
<b>Figura 17.</b> Posiciones de las variables articulares para el cálculo de torque máximo.....	49
<b>Figura 18.</b> Torque aplicado para cada variable articular. a) $q_1$ , b) $q_2$ .....	49
<b>Figura 19.</b> Torque aplicado para cada variable articular. a) $q_3$ , b) $q_4$ .....	50
<b>Figura 20.</b> Torque aplicado para cada variable articular. a) $q_5$ , b) $q_6$ .....	50
<b>Figura 21.</b> Método de diseño para manufactura.....	53
<b>Figura 22.</b> Perfiles para correas sincrónicas. a) Perfil trapecoidal b) Perfil curvo.....	55
<b>Figura 23.</b> Términos utilizados en el diseño de perfiles de poleas sincrónicas.....	56
<b>Figura 24.</b> Proceso de implementación de subsistema mecánico.....	60
<b>Figura 25.</b> Eslabón 1. Diseño CAD.....	61
<b>Figura 26.</b> Eslabón 1. Impresión 3D .....	61
<b>Figura 27.</b> Eslabón 1. Simulación CAM para impresión 3D .....	61
<b>Figura 28.</b> Eslabón 1. Ensamblaje de piezas .....	62
<b>Figura 29.</b> Partes impresas en 3D.....	62
<b>Figura 30.</b> Ensamblaje del subsistema mecánico.....	62
<b>Figura 31.</b> Ensamblaje de conjunto .....	63
<b>Figura 32.</b> Componentes del subsistema electrónico .....	65
<b>Figura 33.</b> Esquema electrónico.....	69
<b>Figura 34.</b> Proceso de implementación de subsistema electrónico .....	69

<b>Figura 35.</b> Diagrama de conexión entre Raspberry Pi y módulo Adafruit PCA9685 .....	70
<b>Figura 36.</b> Diseño y construcción de la caja de conexiones.....	70
<b>Figura 37.</b> Componentes del subsistema TICs .....	72
<b>Figura 38.</b> Proceso de configuración para acceso remoto .....	73
<b>Figura 39.</b> Lazo de control .....	73
<b>Figura 40.</b> Interfaz HMI: pantalla principal .....	74
<b>Figura 41.</b> Interfaz HMI: cinemática directa.....	74
<b>Figura 42.</b> Interfaz HMI: cinemática inversa .....	75
<b>Figura 43.</b> Interfaz HMI: generar secuencia.....	75
<b>Figura 44.</b> Interfaz HMI: reproducir secuencia.....	76
<b>Figura 45.</b> Movimientos generados mediante la interfaz HMI .....	76
<b>Figura 46.</b> Jerarquía de nodos desarrollados en ROS .....	77
<b>Figura 47.</b> Pantalla cinemática directa. Diagrama de nodos ROS .....	77
<b>Figura 48.</b> Pantalla cinemática inversa. Diagrama de nodos ROS.....	78
<b>Figura 49.</b> Pantalla generar secuencia. Diagrama de nodos ROS .....	78
<b>Figura 50.</b> Pantalla reproducir secuencia. Diagrama de nodos ROS .....	78
<b>Figura 51.</b> Diagrama de nodos ROS. Árbol general .....	79
<b>Figura 52.</b> Precisión y repetitividad de posición.....	83
<b>Figura 53.</b> Resultados de precisión y repetitividad de posición.....	85
<b>Figura 54.</b> Obtención de matriz homogénea. Diagrama de flujo .....	94
<b>Figura 55.</b> Modelamiento de torque máximo. Diagrama de flujo.....	95
<b>Figura 56.</b> Nodo “interfaz_cd”. Diagrama de flujo.....	96
<b>Figura 57.</b> Nodo “interfaz_ci”. Diagrama de flujo.....	97

<b>Figura 58.</b> Nodo “interfaz_generar_secuencia”. Diagrama de flujo .....	98
<b>Figura 59.</b> Nodo “interfaz_reproducir_secuencia”. Diagrama de flujo .....	99
<b>Figura 60.</b> Nodo “calculo_cd”. Diagrama de flujo.....	100
<b>Figura 61.</b> Nodo “calculo_ci”. Diagrama de flujo .....	100
<b>Figura 62.</b> Nodo “datos_servo”. Diagrama de flujo.....	101
<b>Figura 63.</b> Nodo “control_servos”. Diagrama de flujo .....	102

## RESUMEN

Un importante campo de investigación dentro del área de la Mecatrónica comprende el desarrollo de plataformas robóticas basadas en software libre, enfocado principalmente en el sistema operativo ROS (Robot Operating System). La principal desventaja con respecto a los manipuladores robóticos comerciales se basa en la falta de flexibilidad en el diseño, debido a que el hardware se encuentra protegido por derechos de autor o parcialmente liberado. Existe la necesidad de desarrollar una plataforma que facilite la enseñanza y aprendizaje de la robótica enfocada en el hardware y software libre con aplicaciones industriales. Para el desarrollo del proyecto, se planteó el uso de la metodología de diseño de sistemas mecatrónicos propuesta dentro de la norma VDI 2206. La fabricación del prototipo se realizó principalmente utilizando el proceso de impresión 3D con PLA en una impresora Prusa i3 con un volumen de impresión de 225x225x160 [mm], con un tiempo de fabricación aproximado de 93 horas. El cálculo de la cinemática directa se realizó utilizando la definición de parámetros de Denavit-Hartenberg, mientras que la cinemática inversa se obtuvo mediante la aplicación del concepto de desacople cinemático. Como resultado se obtuvo un robot antropomórfico de seis grados de libertad Open Source (OSH-OSS) con una precisión de 2.03 [mm] y repetitividad de  $\pm 4.89$  [mm], con una carga útil de 0.15 [kg] y una relación de carga útil de 5.21%.

### Palabras clave:

- **MANIPULADOR ROBÓTICO**
- **HARDWARE LIBRE**
- **SOFTWARE LIBRE**

## ABSTRACT

An important research field within Mechatronics includes the development of robotic platforms based on free software, mainly focused on ROS (Robot Operating System). The main disadvantage of commercial robotic manipulators is based on the lack of design flexibility, because the hardware is protected by copyright or partially released. There is a need to develop a platform to help the teaching and learning of robotics focused on free hardware and software with industrial applications. For the development of the project, the use of the mechatronic systems design methodology proposed within the VDI 2206 standard as proposed. The manufacturing of the prototype was mainly carried out using the 3D printing process with PLA on a Prusa i3 printer with a print volume of 225x225x160 [mm], with an approximate manufacturing time of 93 hours. The calculation of the direct kinematics was made using the Denavit-Hartenberg parameter definition, while the inverse kinematics was obtained by applying the kinematic decoupling concept to simplify the calculation. As a result we obtained a six degrees of freedom Open Source (OSH-OSS) anthropomorphic robot with an accuracy of 2.03 [mm] and repeatability of  $\pm 4.89$  [mm], with a payload of 0.15 [kg] and a payload ratio of 5.21%.

### **Keywords:**

- **ROBOTIC MANIPULATOR**
- **OPEN HARDWARE**
- **OPEN SOFTWARE**

# CAPÍTULO I

## GENERALIDADES

### 1.1 Antecedentes

Un importante campo de investigación dentro del área de la Mecatrónica comprende el desarrollo de plataformas robóticas basadas en software libre, enfocado principalmente en el sistema operativo ROS (Robot Operating System). ROS fue diseñado para ser lo más distribuido y modular posible, de modo que los usuarios puedan implementar las funcionalidades que requieran de acuerdo a los requerimientos específicos de su aplicación (Open Source Robotics Foundation, s.f.). Algunos ejemplos de manipuladores robóticos de software libre incluyen:

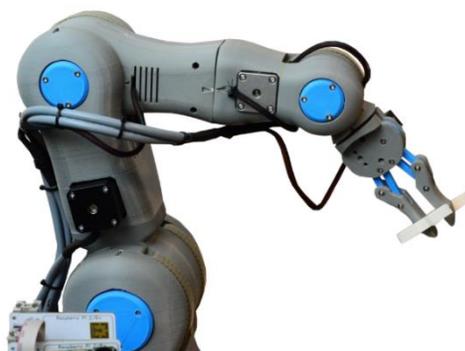
- a) Niryo One: Es un brazo robótico de 6 ejes impreso en 3D, para fabricantes, educadores e investigadores. Está alimentado por Arduino, Raspberry Pi y ROS (Niryo, 2018). Se presenta en la Figura 1.



**Figura 1.** Niryo One. Robot open source de 6 grados de libertad

FUENTE: (Niryo, 2018)

b) RBX1: Es un brazo robótico de 6 grados de libertad impreso en 3D. El robot está construido usando solo una impresora 3D sin componentes mecanizados, y utiliza solo un Raspberry Pi y un controlador de motor a pasos de tipo propietario. No se requiere una computadora externa (Roboteurs, 2018). Se presenta en la Figura 2.



*Figura 2.* RBX1. Brazo robótico de 6 grados de libertad impreso en 3D

FUENTE: (Roboteurs, 2018)

La principal desventaja con los manipuladores comerciales radica en la falta de flexibilidad en el diseño, debido a que el hardware se encuentra protegido por derechos de autor o parcialmente liberado. La Universidad de las Fuerzas Armadas – ESPE, a través de la carrera de Ingeniería Mecatrónica, dedica esfuerzos para el desarrollo de robots basados en software libre. A pesar de esto, no se han realizado proyectos relacionados a la necesidad de universalizar el conocimiento en lo que respecta a hardware libre con fines académicos.

## **1.2 Justificación e Importancia**

En el Campus Matriz de la Universidad de las Fuerzas Armadas – ESPE existen tres manipuladores robóticos industriales KUKA de seis grados de libertad, que regularmente son utilizados para la enseñanza de la robótica dentro de la Carrera de Ingeniería Mecatrónica. A pesar

de ser una alternativa industrialmente aceptada, la misma presenta inconvenientes para la enseñanza de la robótica, entre los que destacan:

- Riesgos de seguridad para usuarios inexpertos.
- Limitado número de equipos en relación al número de estudiantes.
- Elevado costo de mantenimiento.
- Software y Hardware cerrado.

Existe la necesidad de desarrollar una plataforma que facilite la enseñanza y aprendizaje de la robótica enfocada en el software libre con aplicaciones industriales. Se busca desarrollar un manipulador robótico basado en ROS, con enfoque en su desarrollo posterior para aplicaciones de robótica colaborativa.

### **1.3 Alcance del proyecto**

El proyecto abarca el diseño e implementación de un robot antropomórfico de seis grados de libertad open source. Esto incluye el diseño y construcción del manipulador robótico enfocado al concepto de hardware libre, el desarrollo de software basado en ROS (Robot Operating System) y la integración del sistema ciber-físico (CPS).

El desarrollo del proyecto se basa en la consideración de cuatro aspectos fundamentales:

- Diseño e implementación del subsistema mecánico.
- Diseño e implementación del subsistema eléctrico/electrónico.
- Diseño e implementación del subsistema de control.
- Desarrollo de software.

El subsistema mecánico comprende el diseño, simulación y fabricación del manipulador robótico. El diseño desarrollado se utilizará como fundamento para realizar el modelamiento matemático del manipulador para la obtención de los parámetros geométricos, cinemáticos y dinámicos.

El subsistema eléctrico/electrónico consiste en el diseño electrónico de los circuitos de control y potencia. Los componentes electrónicos para el control de los actuadores se implementarán tomando en consideración los requerimientos de procesamiento de datos y las características de control específicas de cada actuador. Se realizarán las adecuaciones necesarias para garantizar la compatibilidad entre la tarjeta de procesamiento y los actuadores seleccionados a través de una tarjeta de control.

El subsistema de control se enfocará en el posicionamiento del manipulador robótico de seis grados de libertad con desacople cinemático. Para la validación del sistema de control, el posicionamiento del robot se desarrollará a partir de un modo de operación basado en coordenadas universales, como mínimo.

Con respecto al apartado de software, se plantea desarrollar una plataforma basada en software libre, enfocada en la integración del robot con el sistema operativo ROS. La ventaja del uso de ROS como plataforma base radica en la modularidad que presenta gracias a su arquitectura basada en nodos intercambiables.

## **1.4 Objetivos**

### **1.4.1 Objetivo general**

Diseñar e implementar un robot antropomórfico de seis grados de libertad Open Source (OSH-OSS) para el Laboratorio de Mecatrónica y Sistemas Dinámicos de la Universidad de las Fuerzas Armadas – ESPE.

### **1.4.2 Objetivos específicos**

- Diseñar y construir un robot antropomórfico de seis grados de libertad enfocado en el concepto de hardware libre.
- Calcular la cinemática y dinámica de un robot antropomórfico de seis grados de libertad (cadena cinemática abierta).
- Diseñar e implementar una interfaz de control y comunicación basada en nodos de ROS.
- Implementar un sistema de interfaz humano-máquina (HMI) para interacción amigable con el usuario.

## CAPÍTULO II

### ESTADO DEL ARTE

#### 2.1 Manipuladores robóticos industriales

Los robots industriales se consideran una piedra angular de la fabricación competitiva, que pretende combinar alta productividad, calidad y adaptabilidad a un costo mínimo. Para poder ser producido en cantidades suficientemente grandes, un diseño de robot debe cumplir con los requisitos para el conjunto más amplio de aplicaciones potenciales. Como esto es difícil de lograr en la práctica, han surgido diversas clases de diseños de robot con respecto a la capacidad de carga útil, el número de ejes del robot y el volumen del espacio de trabajo para categorías de aplicaciones como ensamblaje, paletizado, pintura, soldadura, mecanizado y manipulación general.

Los robots industriales de hoy en día son principalmente el resultado de los requisitos de la fabricación de gran volumen con uso intensivo de capital, principalmente definida por las industrias automotriz, electrónica y de productos eléctricos. (Hägele, Nilsson, & Pires, 2008)

##### 2.1.1 Componentes de un robot

De acuerdo con (Niku, 2001), un robot integra básicamente los siguientes elementos:

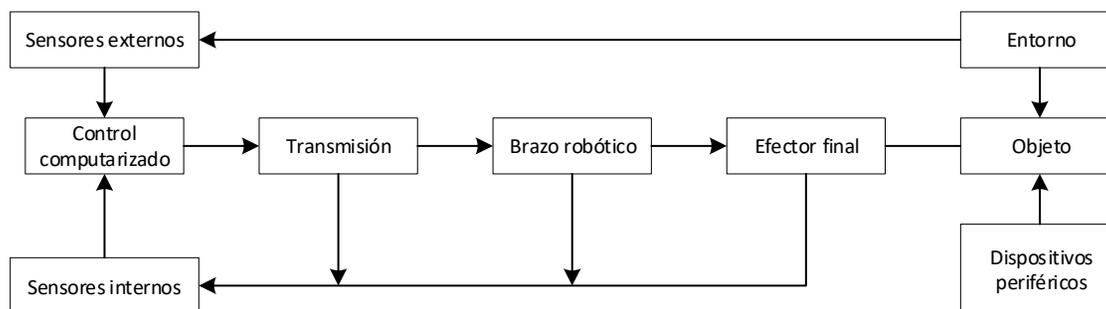
- Manipulador. Constituye el cuerpo del robot y se constituye de eslabones, articulaciones y otros elementos que conforman el robot.
- Efector final. Es el componente conectado al último eslabón de un manipulador, que generalmente manipula objetos, interactúa con otras máquinas o ejecuta las tareas requeridas.

Algunos ejemplos de efectores finales incluyen sopletes de soldadura, pistolas de pintura, ventosas, entre otros.

- Actuadores. Entregan la fuerza requerida por los manipuladores. Los tipos más comunes de actuadores incluyen: servomotores, motores a pasos, cilindros neumáticos y cilindros hidráulicos.
- Sensores. Se utilizan para adquirir información sobre el estado interno del robot, o para comunicarse con el medio externo. Los sensores envían información sobre cada eslabón o articulación hacia el controlador encargado de determinar la posición del robot en el espacio.
- Controlador. Se encarga de recibir datos desde un computador, controlar el movimiento de los actuadores y coordinar los movimientos con la información de retroalimentación de los sensores. En robots más sofisticados, el controlador además se encarga del control de la velocidad y fuerza ejercida por el robot.
- Procesador. Constituye el cerebro del robot, debido a que calcula la posición y velocidad de las articulaciones del manipulador que permitan alcanzar la posición y velocidad deseada, cuyos datos trabajan coordinadamente con los sensores y actuadores. El procesador es generalmente una computadora, por lo que requiere un sistema operativo, programas y periféricos para su adecuado funcionamiento.
- Software. Los componentes de software presentes en un robot pueden agruparse en tres categorías:
  - a) Sistema operativo.
  - b) Software robótico, que calcula los movimientos necesarios de cada articulación basado en las ecuaciones cinemáticas del robot. Estos datos se envían al controlador.

c) Rutinas y aplicaciones para interacción humano-máquina.

La relación funcional de cada uno de los subsistemas componentes de un manipulador robótico se encuentra representada en la Figura 3.



**Figura 3.** Modelo de sistema de un robot

FUENTE: (Warnecke, Schraft, Hägele, Barth, & Schmierer, 1999)

### 2.1.2 Características de un robot

De acuerdo con (Niku, 2001), las siguientes definiciones se utilizan para caracterizar las especificaciones del robot:

- **Carga útil:** es el peso que un robot puede sostener sin alterar sus demás especificaciones. Por ejemplo, la capacidad de carga máxima puede ser mucho mayor a la carga útil especificada, pero a su capacidad máxima, este puede llegar a ser menos preciso o no seguir la trayectoria definida con precisión. La carga útil de los robots en comparación con su propio peso es generalmente muy pequeña.
- **Alcance:** representa la máxima distancia que un robot puede abarcar dentro de su volumen de trabajo, y es una función de la longitud de los eslabones del robot y su configuración.

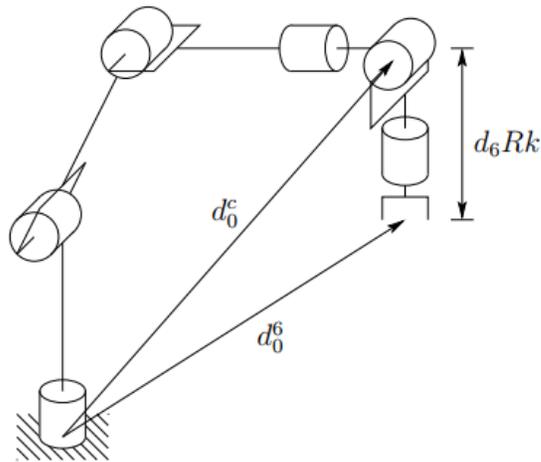
- Precisión (validez): se define en función de la exactitud con la cual el robot puede alcanzar un punto especificado. Esta es una función de la resolución de los actuadores, así como sus dispositivos de retroalimentación.
- Repetibilidad (variabilidad): se refiere a la capacidad del robot para alcanzar la misma posición si el movimiento se repite varias veces, y es mucho más importante que la precisión. Si un robot no es preciso, este presentará un error consistente, que se puede predecir y corregir mediante programación.

### **2.1.3 Grados de libertad de un robot**

Para localizar un punto en el espacio, se requiere especificar tres coordenadas  $x$ ,  $y$ ,  $z$  sobre los ejes cartesianos. Tres coordenadas son necesarias y suficientes para definir la localización del punto. De manera similar, para localizar un cuerpo rígido en el espacio, se necesita especificar la localización de un punto que pertenezca a dicho cuerpo, y por lo tanto requiere tres piezas de información para localizarlo adecuadamente.

Sin embargo, a pesar de que se especifica la localización del objeto, existen infinitas posibilidades de orientar el objeto con respecto al punto seleccionado. Para definir completamente un cuerpo rígido en el espacio, además de especificar la localización de un punto seleccionado, se requiere especificar la orientación del objeto. Esto quiere decir que se requiere un total de seis piezas de información para definir completamente la posición y orientación de un cuerpo rígido.

De la misma forma, se requieren seis grados de libertad para ubicar completamente un objeto en el espacio y orientarlo como se desee (Niku, 2001). Un ejemplo de manipulador robótico de seis grados de libertad se representa en la Figura 4.

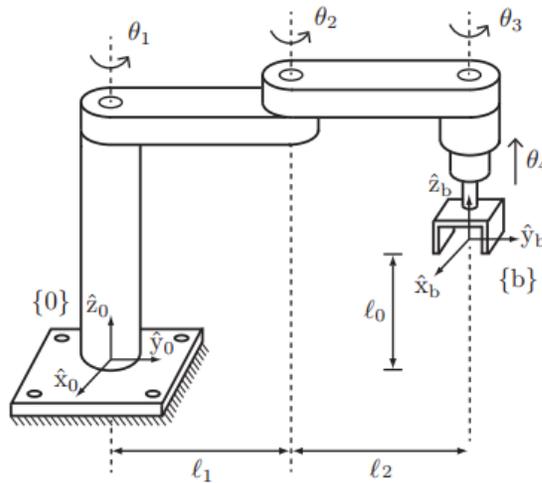


**Figura 4.** Manipulador de 6 DOF con desacople cinemático

FUENTE: (Lynch & Park, 2017)

Si existen menos de seis grados de libertad, las capacidades del robot son limitadas. En un manipulador robótico, el efector final no se considera como un grado de libertad, por lo tanto, ninguno de sus movimientos cuenta como parte de los grados de libertad de un robot.

Existen varios robots en la industria que poseen menos de seis grados de libertad. El robot SCARA de la Figura 5 es una cadena abierta de RRRP que se utiliza ampliamente para tareas de recogida y colocación en mesa. La configuración del efector final está completamente descrita por los cuatro parámetros  $(x; y; z; \phi)$ , donde  $(x; y; z)$  denota la posición cartesiana del punto central del efector final y  $\phi$  denota la orientación del extremo efector en el plano  $xy$ . (Lynch & Park, 2017)



**Figura 5.** Robot SCARA RRRP para operaciones “pick and place”

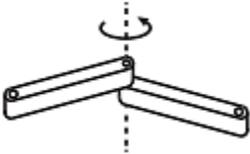
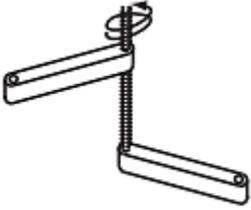
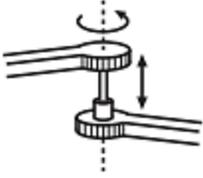
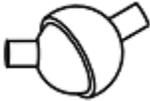
FUENTE: (Lynch & Park, 2017)

### 2.1.4 Espacio de trabajo

El espacio de trabajo de un manipulador es el volumen total barrido por el efector final mientras el manipulador ejecuta todos los movimientos posibles. El espacio de trabajo está limitado por la geometría del manipulador, así como por las restricciones mecánicas en las uniones. Por ejemplo, una junta de revolución puede estar limitada a menos de  $360^\circ$  de movimiento completo. El espacio de trabajo a menudo se divide en un espacio de trabajo accesible y un espacio de trabajo diestro.

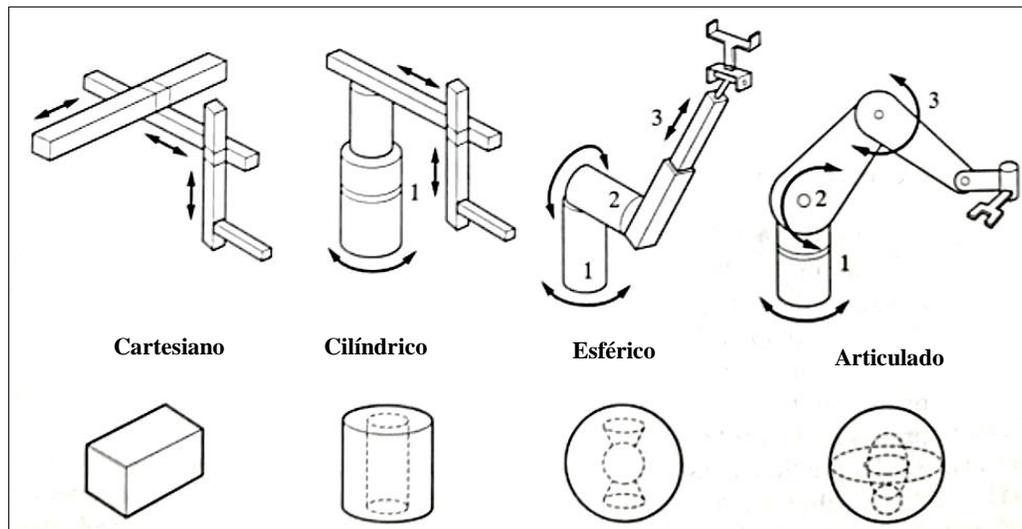
El espacio de trabajo alcanzable es el conjunto completo de puntos alcanzables por el manipulador, mientras que el espacio de trabajo diestro consiste en aquellos puntos que el manipulador puede alcanzar con una orientación arbitraria del efector final. Obviamente, el espacio de trabajo diestro es un subconjunto del espacio de trabajo accesible. (Spong, Hutchinson, & Vidyasagar, 2005). La Tabla 1 describe los tipos de articulaciones más comunes y sus características funcionales.

**Tabla 1**  
*Número de grados de libertad para articulaciones comunes*

TIPO DE ARTICULACIÓN	REPRESENTACIÓN	DEFINICIÓN	DOF
Revolución (R)		También llamada articulación de bisagra, permite un movimiento de rotación alrededor del eje de la articulación.	1
Prismática (P)		También llamada junta deslizante o lineal, permite el movimiento de traslación (o rectilíneo) a lo largo de la dirección del eje de la articulación.	1
Helicoidal (H)		También llamado junta de tornillo, permite la rotación y traslación simultáneas de un eje de tornillo.	1
Cilíndrica (C)		Permite traslaciones y rotaciones independientes sobre un solo eje de articulación fijo.	2
Universal (U)		Consiste en un par de juntas de revolución dispuestas de modo que sus ejes articulares sean ortogonales	2
Esférica (S)		Tiene funciones muy parecidas a nuestra articulación del hombro.	3

FUENTE: (Lynch & Park, 2017)

Dependiendo de su configuración y el tamaño de los eslabones, los robots pueden alcanzar una colección de puntos denominado espacio de trabajo. La forma del espacio de trabajo para cada robot está relacionado a sus características. La Figura 6 muestra el espacio de trabajo aproximado para configuraciones comunes.

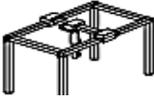
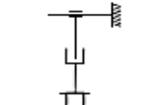
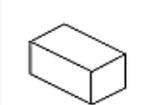
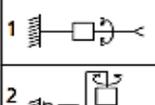
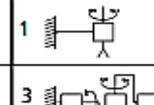
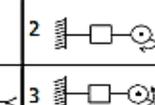
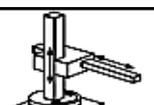
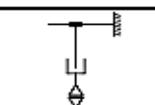
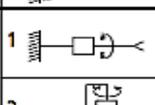
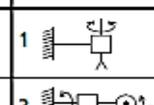
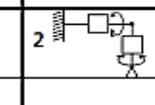
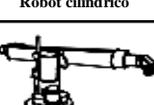
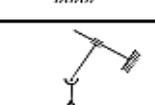
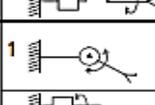
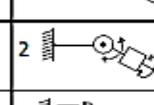
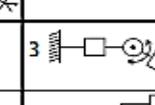
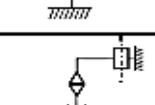
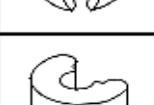
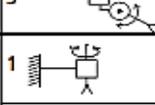
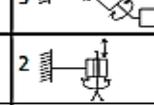
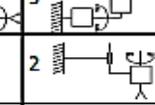
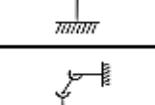
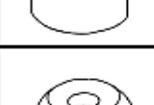
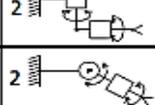
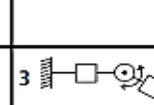
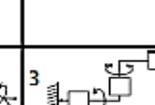
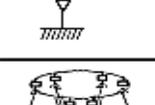


**Figura 6.** Espacios de trabajo típicos para configuraciones comunes

FUENTE: (Niku, 2001)

### 2.1.5 Configuraciones de robots industriales

La elección del mecanismo, sus propiedades cinemáticas, los métodos de cálculo utilizados para determinar los movimientos de las articulaciones y la aplicación prevista de un manipulador de robots están estrechamente relacionados. Los diagramas en la Figura 7 muestran varios tipos comunes de configuraciones para robots industriales.

Robot	Ejes		Muñeca (DOF)		
	Principio	Cadena cinemática			
 Robot cartesiano			 1	 1	 2
 Robot cilíndrico			 1	 1	 2
 Robot esférico			 1	 2	 3
 Robot SCARA			 1	 2	 2
 Robot articulado			 2	 3	 3
 Robot paralelo					

**Figura 7.** Configuraciones típicas de brazo y muñeca de robots industriales

FUENTE: (Warnecke, Schraft, Hägele, Barth, & Schmierer, 1999)

La elección de la estructura mecánica del robot depende principalmente de requisitos mecánicos fundamentales, como la carga útil y el tamaño del espacio de trabajo. Teniendo en cuenta un determinado nivel de costo, generalmente hay una compensación entre el tamaño del espacio de trabajo y la rigidez. Para permitir que el robot alcance dentro o alrededor de obstáculos, es claramente ventajoso utilizar un diseño mecánico articulado. (Hägele, Nilsson, & Pires, 2008)

### 2.1.6 Aplicaciones de robots industriales

Los robots son los más adecuados para trabajar en entornos donde los humanos no pueden realizar las tareas. Los robots ya han sido utilizados en muchas industrias y para diferentes propósitos. A menudo pueden desempeñarse mejor que los humanos y a un costo menor. (Niku, 2001) presenta una lista de aplicaciones de los robots industriales, entre las que destacan:

- Carga de maquinaria: los robots abastecen o remueven partes de otras máquinas. En este tipo de trabajo, el robot solo es un medio para maniobrar partes dentro de un conjunto de operaciones.
- Operaciones “pick and place”: el robot recoge partes y las ubica en otra parte. Esto puede incluir paletización, ensamblaje simple, colocar piezas dentro de un horno y otras rutinas similares.
- Soldadura: el robot, con la configuración adecuada y un efector final de soldadura, se utiliza para soldar partes. Este es una de las aplicaciones más comunes de los robots en la industria automotriz.
- Pintura: es otra aplicación muy común de los robots, especialmente en la industria automotriz, debido a que el entorno de trabajo es nocivo para el ser humano.
- Manufactura: incluye diferentes operaciones tales como remoción de material, taladrado, colocación de pegamento, corte, etc.

Con el paso de los años, se han producido muchas aplicaciones con la frecuencia suficiente para que puedan surgir diseños de robots que satisfagan los requisitos específicos del proceso. Ejemplos de los diferentes diseños y sus requisitos específicos se dan en la Tabla 2.

**Tabla 2**

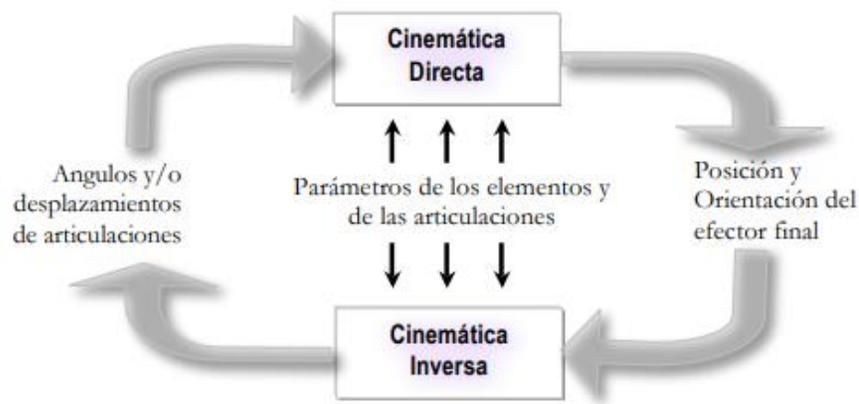
*Diseños de robots específicos de la aplicación y sus principales requisitos funcionales*

			
<b>Paletizado</b>	<b>Soldadura</b>	<b>Ensamblaje</b>	<b>Pintura en spray</b>
<ul style="list-style-type: none"> <li>• Amplia área de trabajo.</li> <li>• Altas velocidades.</li> <li>• Altas aceleraciones.</li> <li>• Ocupa poco espacio.</li> <li>• 4 o 5 DOF.</li> </ul>	<ul style="list-style-type: none"> <li>• Interfaces para equipo de soldadura.</li> <li>• Integración de sensores.</li> <li>• Baja capacidad de carga.</li> <li>• 6 DOF.</li> </ul>	<ul style="list-style-type: none"> <li>• Movimientos horizontales/verticales rápidos.</li> <li>• Alta precisión.</li> <li>• Integración de sensores (detección de partes, control de calidad).</li> <li>• 4 – 6 DOF.</li> </ul>	<ul style="list-style-type: none"> <li>• Movimientos espaciales rápidos.</li> <li>• Alta destreza.</li> <li>• Protegido contra explosiones.</li> <li>• 5,6 DOF.</li> <li>• Programación off-line.</li> <li>• Funciones de control de proceso.</li> </ul>
			
<b>Medición, control de calidad</b>	<b>Automatización de laboratorio</b>	<b>Prensado</b>	<b>Maquinado</b>
<ul style="list-style-type: none"> <li>• Alta precisión.</li> <li>• Integración de sensores (táctiles, visión).</li> <li>• Alta destreza.</li> <li>• 5 – 6 DOF.</li> </ul>	<ul style="list-style-type: none"> <li>• Barato.</li> <li>• Fácil de programar.</li> <li>• 3 – 5 DOF.</li> <li>• Capacidad de carga limitada.</li> </ul>	<ul style="list-style-type: none"> <li>• Alta carga útil.</li> <li>• Trabajo pesado.</li> <li>• Rápido.</li> <li>• 4 – 6 DOF.</li> <li>• Equilibrado.</li> </ul>	<ul style="list-style-type: none"> <li>• Alta rigidez.</li> <li>• Precisión de posicionamiento.</li> <li>• Integración de sensores (táctiles, visión).</li> <li>• 6 DOF.</li> </ul>

FUENTE: (Warnecke, Schraft, Hägele, Barth, & Schmierer, 1999)

### 2.1.7 Cinemática de robots manipuladores

La cinemática de un manipulador robótico implica el estudio de la geometría de su movimiento en función del tiempo (posiciones, velocidades y aceleraciones), sin considerar las fuerzas que originan dicho movimiento. Tal como se describe en la Figura 8, el problema cinemático puede dividirse en dos:



**Figura 8.** El problema cinemático

FUENTE: (Jaramillo Botero, s.f.)

- Cinemática directa: A partir de un conjunto de parámetros físicos, que definen la geometría de un manipulador dado, y de los ángulos (asumiendo articulaciones de tipo rotacional) y/o desplazamientos (asumiendo articulaciones de tipo traslacional) articulares se halla la posición y orientación del efector final en el espacio tridimensional.
- Cinemática inversa: A partir de un conjunto de parámetros físicos, que definen la geometría de un manipulador dado, y de una posición y orientación específica del efector final se halla el conjunto de ángulos y/o desplazamientos articulares que dan como resultado dicha posición y orientación del efector final.

### **2.1.8 Representación de Denavit – Hartenberg para cinemática directa de robots**

El modelo de Denavit-Hartenberg (D-H) es una forma sencilla de modelar eslabones y articulaciones que pueden utilizarse dentro de cualquier configuración de robots, independientemente de su secuencia o complejidad. Puede utilizarse para representar transformaciones geométricas en cualquiera de los sistemas de coordenadas existentes (cartesiano, cilíndrico, esférico, Euler, entre otros). Una de las ventajas que presenta Denavit-Hartenberg con respecto a otros métodos similares es que varias formulaciones matemáticas se han desarrollado tomando en consideración su estructura y resultados, tales como el cálculo del Jacobiano, análisis de fuerzas, etc. (Niku, 2001)

A pesar de que la representación D-H es extensamente utilizada en el modelado y análisis del movimiento de robots y se ha convertido en un estándar, el método presenta una imperfección que muchos investigadores han intentado resolver mediante modificaciones del proceso. El problema fundamental radica en que, debido a que todos los movimientos se realizan sobre los ejes x-z, el método no es capaz de representar ningún movimiento sobre el eje y; esto implica que, si existe algún movimiento sobre el eje y a causa de imperfecciones de ensamblaje, dichas irregularidades no pueden ser modeladas con D-H.

### **2.1.9 Las descripciones de tarea - Enseñanza y Programación**

Los lenguajes y entornos de programación de robots se han separado tradicionalmente en programación en línea (utilizando el robot real in situ) y programación fuera de línea (usando herramientas de software sin necesidad del robot). Con el poder creciente de las herramientas de programación fuera de línea, su capacidad emergente para conectarse al robot físico y el creciente nivel de funciones de software que están integradas en el sistema de control del robot, la

programación en línea ahora es inusual, excepto para verificar y ajustar manualmente los programas generados fuera de línea.

Durante la última década y en los desarrollos actuales, la tendencia ha sido centrarse en la herramienta (efector final del robot) y en el conocimiento del proceso necesario para el proceso de fabricación, y para permitir que el operador exprese la tarea del robot en esos términos. Este desarrollo da como resultado la necesidad de un mayor nivel de abstracción para simplificar la programación, lo que refleja el hecho de que el llamado programador de robots conoce muy bien el proceso de producción, pero tiene habilidades de programación bastante limitadas (Hägele, Nilsson, & Pires, 2008). La Tabla 3 describe las características de diferentes niveles de programación en base a su complejidad.

**Tabla 3**

*Niveles abstracción en la programación de robots*

<b>Producto</b>	Descripción de la forma y ensamblaje final de las piezas de trabajo, en términos de dicho producto; el sistema del robot planifica las operaciones.
<b>Proceso</b>	Basado en secuencias conocidas u operaciones de manufactura específicas, cada una de las cuales se expresa en términos de parámetros del proceso.
<b>Herramienta</b>	El movimiento de la herramienta acoplada al robot está especificado en función de programas o guía manual; el usuario conoce el proceso que se ejecuta.
<b>Manipulador</b>	El manipulador robótico en su placa para el montaje de herramientas se programa para moverse en el espacio cartesiano; el usuario conoce la herramienta.
<b>Articulación</b>	Para cada posición requerida se especifican los ángulos de cada articulación, por lo cual resulta complicado ejecutar movimientos en línea recta.

FUENTE: (Hägele, Nilsson, & Pires, 2008)

La cinemática inversa incorporada permite que el movimiento de la herramienta se especifique en coordenadas cartesianas, lo que es claramente una gran simplificación en muchas aplicaciones. Es decir, el usuario del robot podría centrarse más en el trabajo que realizará el robot y menos en el propio robot. Sin embargo, las propiedades del robot, como los límites de las juntas, no se pueden descuidar. (Hägele, Nilsson, & Pires, 2008)

### 2.1.10 Integración del sistema

El escenario de automatización incluye todos los problemas de integración de computadoras y sus periféricos, además de problemas adicionales que tienen que ver con la variedad de dispositivos (eléctrica y mecánicamente incompatibles) y su interacción con el entorno físico (incluyendo sus inexactitudes, tolerancias y efectos físicos no modelados). El número de variaciones es enorme, por lo que a menudo no es posible crear soluciones reutilizables. En total, esto da como resultado una necesidad de ingeniería extensa para poner en funcionamiento un robot. Esta ingeniería se denomina integración de sistemas (Hägele, Nilsson, & Pires, 2008). Las etapas de la integración de sistemas se presenta en la Tabla 4.

**Tabla 4**

*Etapas de la integración del sistema*

<b>Sistema físico</b>	<ul style="list-style-type: none"> <li>• Selección de equipo basado en dimensionamiento para tamaño, carga y esfuerzos.</li> <li>• Interfaces mecánicas (ubicaciones, placas de adaptadores, etc.).</li> <li>• Suministro de energía eléctrica (voltajes y corrientes para robots, efectores, alimentadores, etc.).</li> <li>• Conexiones para señales analógicas (blindaje, escalado, corrientes, niveles binarios, etc.).</li> </ul>
-----------------------	--

**CONTINÚA**

<b>Comunicación</b>	<ul style="list-style-type: none"> <li>• Interconexiones para E / S digitales de un bit.</li> <li>• Comunicación de datos por bytes, incluidas latencias y tasas de bits.</li> <li>• Transferencia de secuencias de bytes.</li> </ul>
<b>Configuración</b>	<ul style="list-style-type: none"> <li>• Configuración de mensajes entre dispositivos que interactúan.</li> <li>• Establecimiento de servicios.</li> <li>• Ajuste para el rendimiento y la utilización de los recursos.</li> </ul>
<b>Aplicación</b>	<ul style="list-style-type: none"> <li>• Definición de funciones / servicios a nivel de aplicación.</li> </ul>
<b>Tarea</b>	<ul style="list-style-type: none"> <li>• Programación de aplicaciones, utilizando los servicios de nivel de aplicación.</li> </ul>

FUENTE: (Hägele, Nilsson, & Pires, 2008)

## 2.2 Definición de código libre (Open Source)

El sitio web de la OSI (Open Source Initiative) define un conjunto de criterios que un paquete de software libre debe cumplir para poder denominarse como tal. Estos son:

- Redistribución gratuita. La licencia no debe restringir a ninguna parte de vender o regalar el software como un componente de una distribución agregada de software que contiene programas de varias fuentes diferentes. La licencia no debe requerir una regalía u otra tarifa para esta venta.
- Código fuente. El programa debe incluir el código fuente, y debe permitir la distribución del código fuente, así como en forma compilada. Cuando alguna forma de producto no se distribuye con el código fuente, debe haber un medio bien publicitado para obtener el código fuente por un costo de reproducción no superior a lo razonable, preferiblemente descargando a través de Internet sin cargo. El código fuente debe ser la forma preferida en que un programador modificará el programa. No se permite código fuente deliberadamente ininteligible.

- Trabajos Derivados. La licencia debe permitir modificaciones y trabajos derivados, y debe permitir que se distribuyan bajo los mismos términos que la licencia del software original.
- Integridad del código fuente del autor. La licencia debe permitir explícitamente la distribución de software creado a partir de código fuente modificado. La licencia puede requerir trabajos derivados para llevar un nombre o número de versión diferente del software original.
- No discriminación contra personas o grupos. La licencia no debe discriminar a ninguna persona o grupo de personas.
- No discriminación en función de la finalidad perseguida. La licencia no debe restringir el uso del programa en un campo específico de actividad.
- Distribución de licencia. Los derechos adjuntos al programa deben aplicarse a todos aquellos a los que se redistribuye el programa sin la necesidad de la ejecución de una licencia adicional por esas partes.
- La licencia no debe ser específica de un producto. Los derechos adjuntos al programa no deben depender de que el programa sea parte de una distribución de software en particular. Si el programa se extrae de esa distribución y se usa o distribuye dentro de los términos de la licencia del programa, todas las partes a quienes se redistribuye el programa deben tener los mismos derechos que los otorgados junto con la distribución de software original.
- La licencia no debe restringir otro software. La licencia no debe imponer restricciones sobre otro software que se distribuye junto con el software licenciado. Por ejemplo, la licencia no debe insistir en que todos los demás programas distribuidos en el mismo medio deben ser software de código abierto.

- La licencia debe ser neutra desde el punto de vista tecnológico. Ninguna disposición de la licencia puede basarse en ninguna tecnología individual o estilo de interfaz.

(Open Source Initiative, 2007)

### 2.3 Open Source Software (OSS)

El movimiento del software de código abierto ha tenido un enorme impacto en la tecnología actual. Ha ayudado a la investigación académica, ha cambiado la forma en que muchas compañías tecnológicas hacen negocios. (Rubow, 2008)

El software de código abierto se basa en las llamadas “4 libertades”, que de acuerdo con (Free Software Foundation Europe, 2018) permite “usar, estudiar, compartir y mejorar el software”. Esta ideología representa ventajas de desarrollo tales como:

- Fomentar la reutilización
- Permitir la innovación, flexibilidad, integración más fácil.
- Reducir el precio del software a cero.

Algunas ventajas del software libre mencionadas por (Government of Prince Edward Island) incluyen:

- Costos más bajos: el software de código abierto generalmente no requiere una tarifa de licencia, por lo que su costo es generalmente más bajo.
- Flexibilidad: un programador puede tomar un paquete de software estándar y modificarlo para adaptarse mejor a las necesidades específicas de su aplicación.

- Fiabilidad y calidad: es imposible decir que el software de fuente abierta es mejor que el software propietario en términos de confiabilidad y calidad. Sin embargo, el software maduro de código abierto generalmente se considera de buena calidad y confiabilidad.
- Disponibilidad de soporte externo: el soporte técnico externo está disponible para muchos de los paquetes de software de código abierto. Algunos proveedores ofrecen contratos de soporte y hay proveedores de servicios que instalan, configuran y mantienen un sistema OSS. Muchos productos de código abierto también tienen un activo soporte a través de blogs en línea.

## 2.4 Open Source Hardware (OSH)

No existe una definición clara y ampliamente aceptada de "hardware de código abierto". Una definición de las primeras definiciones existentes proviene de la licencia de código abierto de TAPR (Tucson Amateur Packet Radio Corp, 2007):

Open Hardware es una cosa, un artefacto físico, ya sea eléctrico o mecánico, cuya información de diseño está disponible para el público y puede ser utilizada por el público de una manera que le permite a cualquiera hacer, modificar, distribuir y usar esa cosa.

La definición ampliamente aceptada proviene de la OSHWA (Open Source Hardware Association), que brinda la siguiente definición:

El hardware de código abierto es un hardware cuyo diseño se pone a disposición del público para que cualquiera pueda estudiar, modificar, distribuir, fabricar y vender el diseño o el hardware en función de ese diseño. La fuente del hardware, el diseño del que está hecho, está disponible en el formato preferido para realizar modificaciones. Idealmente, el hardware de código abierto utiliza componentes y materiales fácilmente disponibles, procesos estándar, infraestructura abierta, contenido no restringido y herramientas de diseño de código abierto para

maximizar la capacidad de las personas para hacer y usar hardware. (Open Source Hardware Association, 2018)

El hardware es diferente del software en el hecho de que los recursos físicos siempre deben estar comprometidos para la creación de bienes físicos. En consecuencia, las personas o empresas que producen productos con una licencia de OSH tienen la obligación de dejar en claro que dichos productos no son fabricados, vendidos, garantizados o aprobados por el diseñador original y tampoco deben utilizar marcas comerciales. propiedad del diseñador original.

Los criterios para considerar un producto como hardware libre se basan en la definición de código abierto para el software libre. Los términos de distribución de Open Source Hardware deben cumplir, además de lo expuesto para componentes de software, con los siguientes criterios:

- Documentación. El hardware debe ser publicado con documentación que incluya archivos de diseño, y debe permitir la modificación y distribución de los archivos de diseño. Cuando la documentación no está provista con el producto físico, debe haber un medio bien publicitado para obtener esta documentación por un costo de reproducción no superior a lo razonable, preferiblemente descargando a través de Internet sin cargo. La documentación debe incluir archivos de diseño en el formato preferido para realizar cambios, por ejemplo, el formato de archivo nativo de un programa CAD. Los archivos de diseño deliberadamente ininteligibles no están permitidos. Las formas intermedias análogas al código compilado de la computadora, como los planos listos para imprimir de un programa de CAD, no están permitidas como sustitutos. La licencia puede requerir que los archivos de diseño se proporcionen en formato (s) abierto (s) y completamente documentado.

- Alcance. La documentación del hardware debe especificar claramente qué parte del diseño, si no es todo, se está liberando bajo la licencia.
- Software necesario. Si el diseño con licencia requiere software, incrustado o no, para operar correctamente y cumplir con sus funciones esenciales, entonces la licencia puede requerir que se cumpla una de las siguientes condiciones: a) Las interfaces están lo suficientemente documentadas como para que se pueda considerar razonablemente sencillo escribir software de código abierto que permita que el dispositivo funcione correctamente y cumpla con sus funciones esenciales. Por ejemplo, esto puede incluir el uso de diagramas detallados de temporización de señal o pseudocódigo para ilustrar claramente la interfaz en funcionamiento. b) El software necesario se libera bajo una licencia de código abierto aprobada por OSI.
- Trabajos Derivados. La licencia debe permitir modificaciones y trabajos derivados, y debe permitir que se distribuyan bajo los mismos términos que la licencia del trabajo original. La licencia debe permitir la fabricación, venta, distribución y uso de productos creados a partir de los archivos de diseño, los propios archivos de diseño y sus derivados.
- Redistribución gratuita. La licencia no debe restringir a un tercero el vender o entregar la documentación del proyecto. La licencia no debe requerir un canon u otra tarifa por tal venta. La licencia no requerirá ninguna regalía o tarifa relacionada con la venta de obras derivadas.
- Atribución. La licencia puede requerir documentos derivados, y avisos de derechos de autor asociados con dispositivos, para otorgar atribución a los licenciantes al distribuir archivos de diseño, productos manufacturados y / o derivados de los mismos. La licencia puede requerir que esta información sea accesible para el usuario final que usa el dispositivo normalmente, pero no

debe especificar un formato específico de visualización. La licencia puede requerir trabajos derivados para llevar un nombre o número de versión diferente del diseño original.

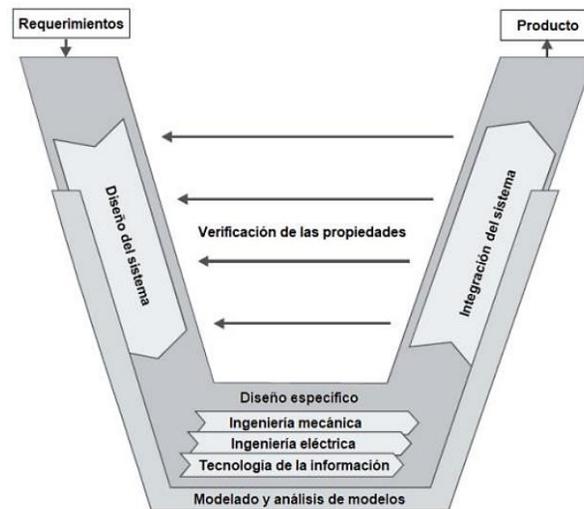
(Open Source Hardware Association, 2018)

## CAPÍTULO III

### DISEÑO MECATRÓNICO

#### 3.1 Metodología

Para el desarrollo del proyecto se plantea el uso de la metodología de diseño de sistemas mecatrónicos propuesta dentro de la norma VDI 2206 (Association of German Engineers, 2004). Dicha metodología se describe como un modelo flexible que puede ser aplicado para el manejo de sistemas tanto a nivel general como específico. La metodología para el diseño del sistema mecatrónico como un modelo integral se valida mediante un procedimiento genérico conocido como modelo V, tal como se describe en la Figura 9.

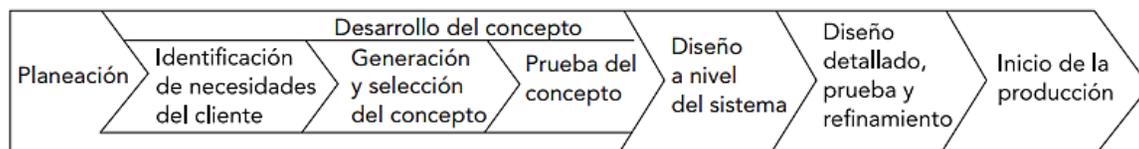


**Figura 9.** Norma VDI 2206. Modelo V como macro-ciclo

FUENTE: (Association of German Engineers, 2004)

Para simplificar el proceso de diseño, se propone dividir las diferentes etapas del proyecto en módulos que pueden ser desarrollados por separado para luego integrarse dentro del producto final. Para el diseño de cada uno de los subsistemas, se plantea la aplicación del proceso de desarrollo de

productos recomendado por (Ulrich, Karl & Eppinger, 2013). La metodología de desarrollo de productos a nivel general se presenta en la Figura 10.



**Figura 10.** Proceso de desarrollo de productos

FUENTE: (Ulrich, Karl & Eppinger, 2013)

## 3.2 Diseño del subsistema mecánico

### 3.2.1 Requerimientos

#### A. Identificación de necesidades

Los requerimientos generales que debe cumplir el subsistema mecánico se presentan en la Tabla 5, y se basan en la necesidad de obtener un producto de fácil ensamblaje y fabricación a un bajo costo.

**Tabla 5**

*Subsistema mecánico. Identificación de necesidades*

	<b>Requerimiento</b>	<b>Prioridad</b>
<b>1</b>	El robot debe ser fácil de fabricar	1
<b>2</b>	El robot debe ser fácil de ensamblar	1
<b>3</b>	El robot debe tener un diseño flexible	3
<b>4</b>	El costo del robot debe ser reducido	2
<b>5</b>	Los materiales utilizados deben ser de fácil acceso	2
<b>6</b>	El robot debe ser compacto	3
<b>7</b>	El robot debe ser ligero	4
<b>8</b>	El robot debe funcionar de manera similar a un manipulador industrial	2

## B. Lista de métricas

Los requerimientos se asocian con una variable que luego pueda ser evaluada para determinar el grado de cumplimiento de los objetivos planteados para el desarrollo del subsistema mecánico. Las métricas establecidas para cada necesidad planteada se presentan en la Tabla 6.

**Tabla 6**

*Lista de métricas para el subsistema mecánico*

	<b>Necesidades</b>	<b>Métrica</b>	<b>Importancia</b>	<b>Unidades</b>
<b>1</b>	1	Tiempo de fabricación	3	horas
<b>2</b>	2	Tiempo de ensamblaje	3	horas
<b>3</b>	6,7	Masa total	4	Kg
<b>4</b>	6	Alcance máximo	2	mm
<b>5</b>	7	Carga útil	1	Kg
<b>6</b>	1,4,5	Costo de fabricación	2	Dólares
<b>7</b>	8	Precisión	2	mm
<b>8</b>	8	Repetitividad	2	%

## C. Matriz de necesidades – métricas

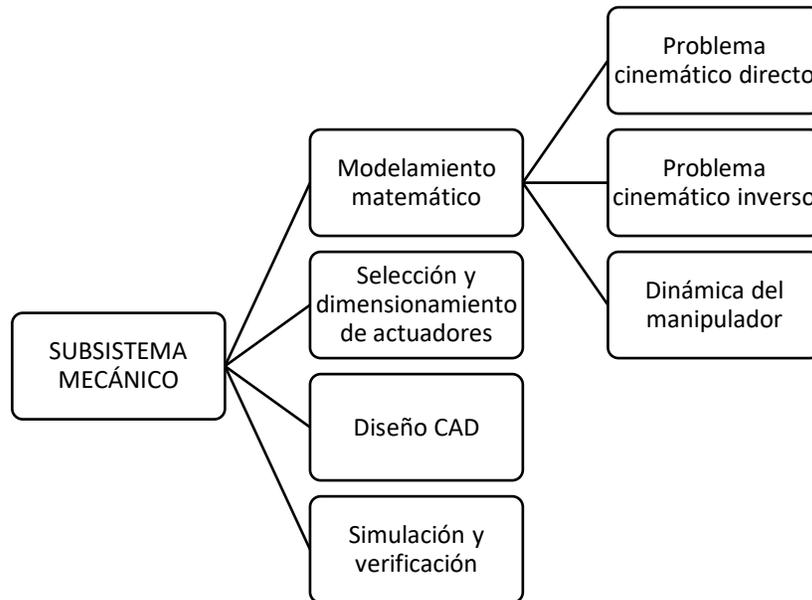
Una matriz de necesidades-métricas representa la relación entre las necesidades del proyecto (representadas en las filas) y las métricas (representadas en las columnas) que permiten evaluar el grado en que el producto satisface los requerimientos establecidos. La matriz para el subsistema mecánico se presenta en la Tabla 7.

**Tabla 7***Matriz de necesidades – métricas. Subsistema mecánico*

		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
<b>Métrica</b>		Tiempo de fabricación	Tiempo de ensamble	Masa total	Alcance máximo	Carga útil	Costo de fabricación	Precisión	Repetitividad
<b>Necesidad</b>									
<b>1</b>	El robot debe ser fácil de fabricar	*					*		
<b>2</b>	El robot debe ser fácil de ensamblar		*						
<b>3</b>	El robot debe tener un diseño flexible								
<b>4</b>	El costo del robot debe ser reducido						*		
<b>5</b>	Los materiales utilizados deben ser de fácil acceso						*		
<b>6</b>	El robot debe ser compacto			*	*				
<b>7</b>	El robot debe ser ligero			*		*			
<b>8</b>	El robot debe funcionar de manera similar a un manipulador industrial							*	*

### 3.2.2 Diseño

El diseño del subsistema mecánico abarca una serie de etapas que se describen en la Figura 11. Se plantea la ejecución del diseño CAD de cada uno de los componentes mecánicos del prototipo, que servirá como fundamento para realizar el modelamiento matemático del manipulador robótico para la obtención de los parámetros geométricos, cinemáticos y dinámicos. Los resultados obtenidos en el modelamiento se utilizarán para el dimensionamiento adecuado de los actuadores en función de los requerimientos propuestos. Los resultados obtenidos se verificarán a través de simulaciones para su posterior fabricación.



**Figura 11.** Componentes del subsistema mecánico

## A. Modelamiento matemático

### a. Cinemática directa

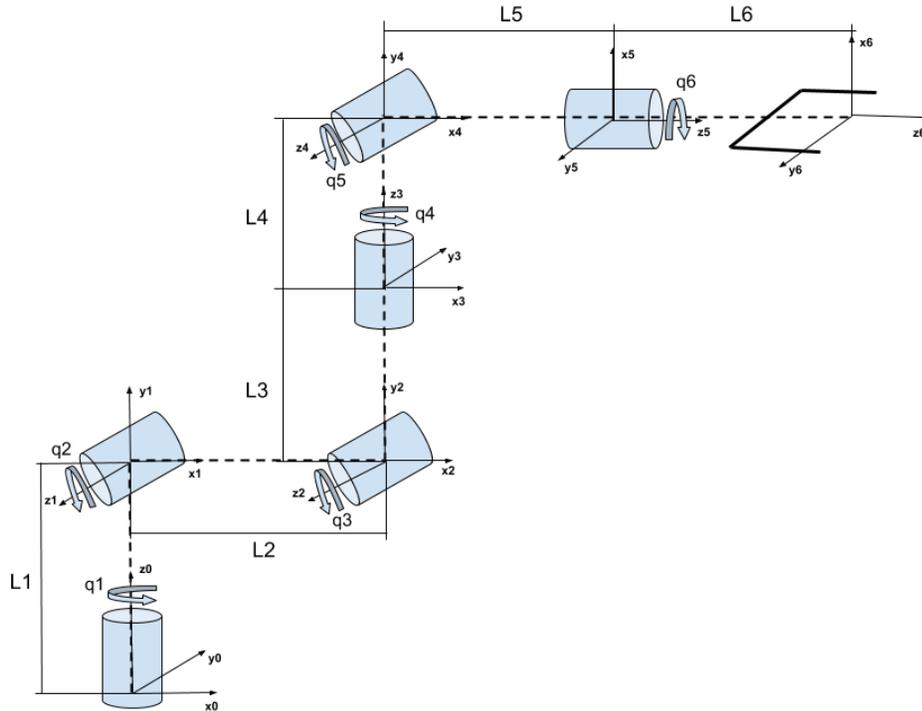
El problema cinemático directo determina la posición y orientación del efector final dados los valores para las variables de sus articulaciones. El método utilizado para obtener las ecuaciones cinemáticas se basa en la convención de Denavit-Hartenberg, que se presenta explicada a detalle en (Spong, Hutchinson, & Vidyasagar, 2005). En esta convención, cada matriz de transformación homogénea  $A_i$  se representa como el producto de cuatro transformaciones básicas, tal como se aprecia en la Ecuación (1).

$$A_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \quad (1)$$

La Ecuación (2) representa la forma general de una matriz homogénea en función de los parámetros de Denavit-Hartenberg, donde los cuatro parámetros  $\theta_i, d_i, a_i, \alpha_i$ , son parámetros asociados con el eslabón y la articulación  $i$  dentro de la cadena cinemática.

$$A_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Para definir la cinemática directa del manipulador robótico representado en la Figura 12, que describe la configuración deseada del prototipo que se desea desarrollar, se requiere obtener la matriz homogénea característica a partir de los parámetros de Denavit-Hartenberg que se presentan en la Tabla 8.



**Figura 12.** Robot antropomórfico de 6 grados de libertad. Representación de ejes de coordenadas

**Tabla 8***Robot antropomórfico de 6 grados de libertad. Parámetros de Denavit – Hartenberg*

	$\theta$	$d$	$a$	$\alpha$
${}^0H_1$	$q_1$	$L_1$	0	$90^\circ$
${}^1H_2$	$q_2$	0	$L_2$	0
${}^2H_3$	$q_3$	0	0	$-90^\circ$
${}^3H_4$	$q_4$	$L_3 + L_4$	0	$90^\circ$
${}^4H_5$	$q_5 + 90^\circ$	0	0	$90^\circ$
${}^5H_6$	$q_6$	$L_5 + L_6$	0	0

En base a los parámetros definidos en la Tabla 8, y aplicando la ecuación general para la obtención de la matriz homogénea para cada una de las variables articulares (Ecuación (2)), se obtienen los resultados que se presentan en las Ecuaciones (3)-(8).

$${}^0H_1 = A_1 = \begin{bmatrix} cq_1 & 0 & sq_1 & 0 \\ sq_1 & 0 & -cq_1 & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$${}^1H_2 = A_2 = \begin{bmatrix} cq_2 & -sq_2 & 0 & L_2cq_2 \\ sq_2 & cq_2 & 0 & L_2sq_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$${}^2H_3 = A_3 = \begin{bmatrix} cq_3 & 0 & -sq_3 & 0 \\ sq_3 & 0 & cq_3 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$${}^3H_4 = A_4 = \begin{bmatrix} cq_4 & 0 & sq_4 & 0 \\ sq_4 & 0 & -cq_4 & 0 \\ 0 & 1 & 0 & L_3 + L_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$${}^4H_5 = A_5 = \begin{bmatrix} -sq_5 & 0 & cq_5 & 0 \\ cq_5 & 0 & sq_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$${}^5H_6 = A_6 = \begin{bmatrix} cq_6 & -sq_6 & 0 & 0 \\ sq_6 & cq_6 & 0 & 0 \\ 0 & 0 & 1 & L_5 + L_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

La matriz homogénea total del manipulador robótico se obtiene multiplicando las matrices obtenidas para cada articulación (Ecuaciones (3)-(8)), obteniendo la matriz de rotación y coordenadas del extremo del efector final, tal como se expresa en la Ecuación (10):

$${}^0H_6 = A_1 * A_2 * A_3 * A_4 * A_5 * A_6 \quad (9)$$

$${}^0H_6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & o_x \\ r_{21} & r_{22} & r_{23} & o_y \\ r_{31} & r_{32} & r_{33} & o_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

A partir de la matriz de rotación representada de forma general en la Ecuación (10), la orientación del efector final se obtiene considerando que su representación se basa ángulos de Euler en secuencia ZYX, cuya secuencia matricial se desarrolla en la Ecuación (11).

$$\begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\alpha & -s\alpha \\ 0 & s\alpha & c\alpha \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$\begin{bmatrix} c\gamma c\beta & c\gamma s\beta s\alpha - s\gamma c\alpha & c\gamma s\beta c\alpha + s\gamma s\alpha \\ s\gamma c\beta & s\gamma s\beta s\alpha + c\gamma c\alpha & s\gamma s\beta c\alpha - c\gamma s\alpha \\ -s\beta & c\beta s\alpha & c\beta c\alpha \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (11)$$

A partir de las expresiones obtenidas de la última fila y la primera columna de la Ecuación (11), y despejando las variables correspondientes se obtienen los tres parámetros requeridos para definir la orientación del efector final con relación al sistema de referencia de la base, que se presentan en las Ecuaciones (12), (13) y (14).

$$\alpha = \text{atan2}\left(\frac{r_{32}}{r_{33}}\right) \quad (12)$$

$$\beta = \text{atan2}\left(\pm \frac{r_{31}}{\sqrt{r_{32}^2 + r_{33}^2}}\right) \quad (13)$$

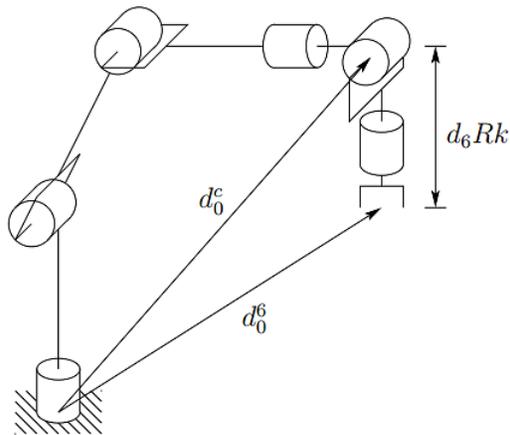
$$\gamma = \text{atan2}\left(\frac{r_{21}}{r_{11}}\right) \quad (14)$$

## b. Cinemática inversa

El problema cinemático inverso determina los valores de las variables de las articulaciones a partir de la posición y orientación del efector final. Para manipuladores con seis grados de libertad, con las tres últimas juntas cruzando en un punto común, es posible separar el problema cinemático inverso en dos problemas más simples, conocidos como cinemática inversa de posición y cinemática inversa de orientación (Spong, Hutchinson, & Vidyasagar, 2005).

En otras palabras, para un manipulador robótico de seis grados de libertad con muñeca esférica, primero se encuentra la posición del punto de intersección de los ejes de la muñeca, conocido como

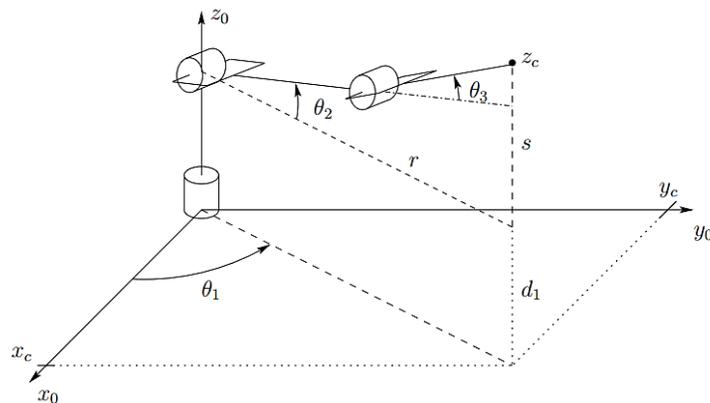
centro de muñeca, y luego se calcula la orientación de la muñeca. La descripción del desacople cinemático se ilustra en la Figura 13.



**Figura 13.** Desacople cinemático

FUENTE: (Spong, Hutchinson, & Vidyasagar, 2005)

Conocida la posición y orientación deseadas dadas por una matriz homogénea similar a la presentada en la Ecuación (10), las coordenadas del centro de la muñeca se obtienen a partir de la relación presentada por la Ecuación (15). La Figura 14 presenta la proyección del centro de muñeca  $(x_c, y_c, z_c)$  en el espacio.



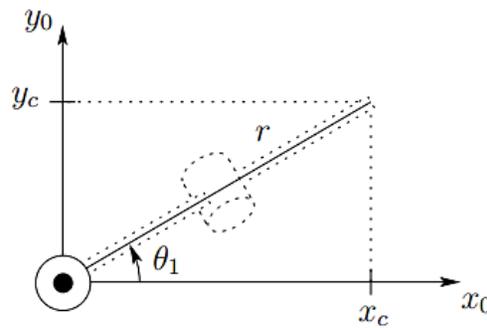
**Figura 14.** Representación de los 3 primeros GDL en el espacio

FUENTE: (Spong, Hutchinson, & Vidyasagar, 2005)

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} o_x - d_6 r_{13} \\ o_y - d_6 r_{23} \\ o_z - d_6 r_{33} \end{bmatrix} \quad (15)$$

A partir de la representación espacial de la Figura 14, el centro de muñeca se proyecta en el plano  $x-y$  como se muestra en la Figura 15. De la proyección se deriva la Ecuación (16), que representa la posición del primer grado de libertad del manipulador robótico.

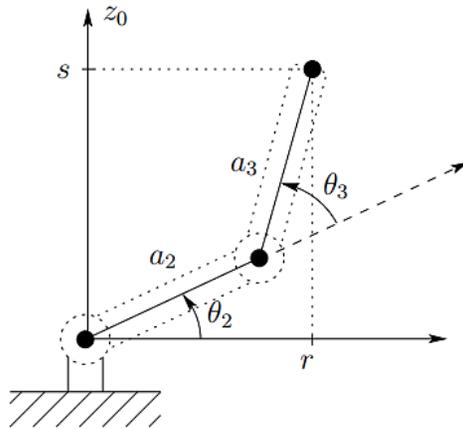
$$\theta_1 = \text{atan2} \left( \frac{y_c}{x_c} \right) \quad (16)$$



**Figura 15.** Proyección del centro de la muñeca en el plano  $x-y$

FUENTE: (Spong, Hutchinson, & Vidyasagar, 2005)

Para encontrar los valores de los ángulos  $\theta_2$ ,  $\theta_3$  que representan la segunda y tercera variable articular del robot, respectivamente, se considera el plano formado por los eslabones 2 y 3 como se describe en la Figura 16.



**Figura 16.** Proyección en el plano formado por los eslabones 2 y 3

FUENTE: (Spong, Hutchinson, & Vidyasagar, 2005)

Tomando en consideración que el movimiento de los eslabones 2 y 3 es planar, la solución es análoga a la propuesta para un manipulador de dos grados de libertad. Aplicando ley de cosenos sobre la proyección presentada en la Figura 16 se obtiene:

$$\cos(\theta_3) = \frac{r^2 + s^2 - a_2^2 - a_3^2}{2a_2a_3} \quad (17)$$

Donde:

$$r^2 = x_c^2 + y_c^2$$

$$s = z_c - d_1$$

Desarrollando la Ecuación (17), y despejando para la variable  $\theta_3$ , se obtiene el resultado presentado en la Ecuación (19).

$$\cos(\theta_3) = \frac{x_c^2 + y_c^2 + (z_c - d_1)^2 - a_2^2 - a_3^2}{2a_2a_3} := D \quad (18)$$

$$\theta_3 = \operatorname{atan2} \left( \pm \frac{\sqrt{1 - D^2}}{D} \right) \quad (19)$$

Las dos soluciones para  $\theta_3$  corresponden a las posiciones de hombro arriba y hombro abajo, respectivamente. De forma similar, la Ecuación (20) describe la solución para la variable  $\theta_2$  a partir de la geometría descrita en la Figura 16.

$$\theta_2 = \operatorname{atan2} \left( \frac{z_c - d_1}{\sqrt{x_c^2 + y_c^2}} \right) - \operatorname{atan2} \left( \frac{a_3 \sin(\theta_3)}{a_2 + a_3 \cos(\theta_3)} \right) \quad (20)$$

Una vez obtenidos los valores para las tres primeras variables articulares, correspondientes con el posicionamiento del robot, cuya matriz de rotación se presenta en la Ecuación 21, se procede con el cálculo de las tres variables articulares correspondientes con la orientación de la muñeca. Para el cálculo de los ángulos de rotación de la muñeca, se toma en cuenta el procedimiento utilizado para la resolución de configuraciones basadas en la muñeca de Euler.

$${}^0R_3 = A_1 A_2 A_3$$

$${}^0R_3 = \begin{bmatrix} c_1 c_{23} & -s_1 & -c_1 s_{23} \\ s_1 c_{23} & c_1 & -s_1 s_{23} \\ s_{23} & 0 & c_{23} \end{bmatrix} \quad (21)$$

La Ecuación (22) representa la ecuación general de una muñeca de Euler para una configuración Z-X-Z.

$${}^3R_6 = \begin{bmatrix} c_4 s_5 c_6 - s_4 s_6 & -c_4 s_5 s_6 - s_4 c_6 & c_4 s_5 \\ s_4 s_5 c_6 + c_4 s_6 & -s_4 c_5 c_6 + c_4 c_6 & s_4 s_5 \\ -s_5 c_6 & s_5 s_6 & c_5 \end{bmatrix} \quad (22)$$

Los valores para las tres variables articulares que componen la muñeca se obtienen despejando las matrices de rotación de acuerdo con la Ecuación (23).

$${}^3R_6 = ({}^0R_3)^T R \quad (23)$$

$$\begin{bmatrix} c_4 s_5 c_6 - s_4 s_6 & -c_4 s_5 s_6 - s_4 c_6 & c_4 s_5 \\ s_4 s_5 c_6 + c_4 s_6 & -s_4 c_5 c_6 + c_4 c_6 & s_4 s_5 \\ -s_5 c_6 & s_5 s_6 & c_5 \end{bmatrix} = \begin{bmatrix} c_1 c_{23} & -s_1 & -c_1 s_{23} \\ s_1 c_{23} & c_1 & -s_1 s_{23} \\ s_{23} & 0 & c_{23} \end{bmatrix}^T \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (24)$$

Las ecuaciones derivadas de la tercera columna de la Ecuación (24) se presentan a continuación:

$$c_4 s_5 = c_1 c_{23} r_{13} + s_1 c_{23} r_{23} + s_{23} r_{33} \quad (25)$$

$$s_4 s_5 = -s_1 r_{13} + c_1 r_{23} \quad (26)$$

$$c_5 = -c_1 s_{23} r_{13} - s_1 s_{23} r_{23} + c_{23} r_{33} \quad (27)$$

A partir de las Ecuaciones (25), (26) y (27) se obtienen los valores para las variables  $\theta_5$  y  $\theta_4$ , presentados en las Ecuaciones (28) y (29), respectivamente.

$$s_5 = \pm \sqrt{1 - (c_5)^2}$$

$$\theta_5 = \text{atan2} \left( \pm \frac{\sqrt{1 - (-c_1 s_{23} r_{13} - s_1 s_{23} r_{23} + c_{23} r_{33})^2}}{-c_1 s_{23} r_{13} - s_1 s_{23} r_{23} + c_{23} r_{33}} \right) \quad (28)$$

$$\theta_4 = \text{atan2} \left( \frac{-s_1 r_{13} + c_1 r_{23}}{c_1 c_{23} r_{13} + s_1 c_{23} r_{23} + s_{23} r_{33}} \right) \quad (29)$$

De la tercera fila de la Ecuación (24) se obtienen las siguientes ecuaciones:

$$-s_5c_6 = -c_1s_{23}r_{11} - s_1s_{23}r_{21} + c_{23}r_{31} \quad (30)$$

$$s_5s_6 = -c_1s_{23}r_{12} - s_1s_{23}r_{22} + c_{23}r_{32} \quad (31)$$

Resolviendo el sistema compuesto por las Ecuaciones (30) y (31), se obtiene el valor de la variable articular  $\theta_6$ :

$$\theta_6 = \text{atan2} \left( \frac{-c_1s_{23}r_{12} - s_1s_{23}r_{22} + c_{23}r_{32}}{c_1s_{23}r_{11} + s_1s_{23}r_{21} - c_{23}r_{31}} \right) \quad (32)$$

### c. Dinámica del manipulador

La mecánica Lagrangiana se basa en la diferenciación de los términos de la energía con respecto a las variables del sistema y el tiempo. Conforme la complejidad de un sistema incrementa, el método del Lagrangiano resulta más simple de usar que los métodos de la mecánica Newtoniana.

El Lagrangiano se define como:

$$L = K - P \quad (33)$$

Donde:

*L: Lagrangiano*

*K: Energía cinética del sistema*

*P: Energía potencial del sistema*

A partir de la definición dada por la Ecuación (33), la mecánica Lagrangiana se basa en dos ecuaciones generales, una para movimientos lineales y otra para movimientos rotacionales. Las

Ecuaciones (34) y (35) permiten obtener los parámetros de fuerza o torque para sistemas mecánicos que presenten movimiento lineal o rotacional, respectivamente.

$$F_i = \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{x}_i} \right) - \frac{\partial L}{\partial x_i} \quad (34)$$

$$T_i = \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\theta}_i} \right) - \frac{\partial L}{\partial \theta_i} \quad (35)$$

Donde:

*F*: Suma de todas las fuerzas externas para un movimiento lineal

*T*: Suma de todos los torques en un movimiento rotacional

*θ, x*: Variables del sistema

De las Ecuaciones (34) y (35), resulta importante notar cómo la complejidad de los términos incrementa de acuerdo al número de grados de libertad del sistema. Tomando en consideración que las ecuaciones dinámicas para robots con múltiples grados de libertad son extensas y complicadas, la obtención de sus ecuaciones a partir del Lagrangiano resulta un proceso complejo que requiere la generación de un algoritmo computacional para facilitar la obtención de resultados. El proceso planteado por (Niku, 2001) presenta una generalización para la obtención del torque en cada junta para robots con n grados de libertad. La ecuación dinámica para un robot general multi-ejes puede resumirse como:

$$T_i = \sum_{j=i}^n D_{ij} \ddot{q}_j + I_{i(act)} \ddot{q}_i + \sum_{j=1}^n \sum_{k=1}^n D_{ijk} \dot{q}_j \dot{q}_k + D_i \quad (36)$$

La primera parte de la ecuación representa los términos de aceleración-inercia, la segunda parte expresa la inercia del actuador, la tercera parte abarca los términos de aceleración centrífuga y Coriolis, y el último término describe los efectos de la gravedad sobre el sistema.

Donde:

$$D_{ij} = \sum_{p=\max(i,j)}^n \text{tr}(U_{pj} J_p U_{pi}^T)$$

$$D_{ijk} = \sum_{p=\max(i,j,k)}^n \text{tr}(U_{pj} J_p U_{pi}^T)$$

$$D_i = \sum_{p=i}^n -m_p g^T U_{pi} \bar{r}_p$$

$$U_{ij} = \frac{\partial^0 T_i}{\partial q_j} = \frac{\partial(A_1 A_2 \dots A_j \dots A_i)}{\partial q_j} = A_1 A_2 \dots Q_j A_j \dots A_i \quad j \leq i$$

$$U_{ijk} = \frac{\partial U_{ij}}{\partial q_k}$$

$$Q_i(\text{revolución}) = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$J_i = \begin{bmatrix} \frac{(-I_{xx} + I_{yy} + I_{zz})}{2} & I_{xy} & I_{xz} & m_i \bar{x}_i \\ I_{xy} & \frac{(I_{xx} - I_{yy} + I_{zz})}{2} & I_{yz} & m_i \bar{y}_i \\ I_{xz} & I_{yz} & \frac{(I_{xx} + I_{yy} - I_{zz})}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix}$$

Para el caso particular del proyecto, la ecuación general de movimiento que describe el torque para un robot de seis grados de libertad con juntas de revolución es la siguiente:

$$\begin{aligned}
T_i = & D_{i1}\ddot{\theta}_1 + D_{i2}\ddot{\theta}_2 + D_{i3}\ddot{\theta}_3 + D_{i4}\ddot{\theta}_4 + D_{i5}\ddot{\theta}_5 + D_{i6}\ddot{\theta}_6 + I_{i(act)}\ddot{\theta}_i \\
& + D_{i11}\dot{\theta}_1^2 + D_{i22}\dot{\theta}_2^2 + D_{i33}\dot{\theta}_3^2 + D_{i44}\dot{\theta}_4^2 + D_{i55}\dot{\theta}_5^2 + D_{i66}\dot{\theta}_6^2 \\
& + D_{i12}\dot{\theta}_1\dot{\theta}_2 + D_{i13}\dot{\theta}_1\dot{\theta}_3 + D_{i14}\dot{\theta}_1\dot{\theta}_4 + D_{i15}\dot{\theta}_1\dot{\theta}_5 + D_{i16}\dot{\theta}_1\dot{\theta}_6 \\
& + D_{i21}\dot{\theta}_2\dot{\theta}_1 + D_{i23}\dot{\theta}_2\dot{\theta}_3 + D_{i24}\dot{\theta}_2\dot{\theta}_4 + D_{i25}\dot{\theta}_2\dot{\theta}_5 + D_{i26}\dot{\theta}_2\dot{\theta}_6 \\
& + D_{i31}\dot{\theta}_3\dot{\theta}_1 + D_{i32}\dot{\theta}_3\dot{\theta}_2 + D_{i34}\dot{\theta}_3\dot{\theta}_4 + D_{i35}\dot{\theta}_3\dot{\theta}_5 + D_{i36}\dot{\theta}_3\dot{\theta}_6 \\
& + D_{i41}\dot{\theta}_4\dot{\theta}_1 + D_{i42}\dot{\theta}_4\dot{\theta}_2 + D_{i43}\dot{\theta}_4\dot{\theta}_3 + D_{i45}\dot{\theta}_4\dot{\theta}_5 + D_{i46}\dot{\theta}_4\dot{\theta}_6 \\
& + D_{i51}\dot{\theta}_5\dot{\theta}_1 + D_{i52}\dot{\theta}_5\dot{\theta}_2 + D_{i53}\dot{\theta}_5\dot{\theta}_3 + D_{i54}\dot{\theta}_5\dot{\theta}_4 + D_{i56}\dot{\theta}_5\dot{\theta}_6 \\
& + D_{i61}\dot{\theta}_6\dot{\theta}_1 + D_{i62}\dot{\theta}_6\dot{\theta}_2 + D_{i63}\dot{\theta}_6\dot{\theta}_3 + D_{i64}\dot{\theta}_6\dot{\theta}_4 + D_{i65}\dot{\theta}_6\dot{\theta}_5 + D_i
\end{aligned} \tag{37}$$

A partir de la Ecuación (37), tomando en consideración los términos de la forma  $D_{ijk}$  y  $D_{ikj}$ , para un robot con todas sus juntas de revolución, se cumple que:

$$D_{ijk} = D_{ikj} \tag{38}$$

Esto permite simplificar la Ecuación (37) para cada una de las articulaciones que componen el robot planteado. Desarrollando las ecuaciones dinámicas para cada una de las variables articulares del sistema, se obtiene un conjunto de seis ecuaciones (Ecuaciones (39)-(44)) que describen el torque en cada una de las articulaciones de la plataforma robótica para un valor determinado de sus variables articulares.

$$\begin{aligned}
T_1 = & D_{11}\ddot{\theta}_1 + D_{12}\ddot{\theta}_2 + D_{13}\ddot{\theta}_3 + D_{14}\ddot{\theta}_4 + D_{15}\ddot{\theta}_5 + D_{16}\ddot{\theta}_6 + I_{1(act)}\ddot{\theta}_1 \\
& + D_{111}\dot{\theta}_1^2 + D_{122}\dot{\theta}_2^2 + D_{133}\dot{\theta}_3^2 + D_{144}\dot{\theta}_4^2 + D_{155}\dot{\theta}_5^2 + D_{166}\dot{\theta}_6^2 \\
& + 2D_{112}\dot{\theta}_1\dot{\theta}_2 + 2D_{113}\dot{\theta}_1\dot{\theta}_3 + 2D_{114}\dot{\theta}_1\dot{\theta}_4 + 2D_{115}\dot{\theta}_1\dot{\theta}_5 + 2D_{116}\dot{\theta}_1\dot{\theta}_6 \\
& + 2D_{123}\dot{\theta}_2\dot{\theta}_3 + 2D_{124}\dot{\theta}_2\dot{\theta}_4 + 2D_{125}\dot{\theta}_2\dot{\theta}_5 + 2D_{126}\dot{\theta}_2\dot{\theta}_6 + 2D_{134}\dot{\theta}_3\dot{\theta}_4 \\
& + 2D_{135}\dot{\theta}_3\dot{\theta}_5 + 2D_{136}\dot{\theta}_3\dot{\theta}_6 + 2D_{145}\dot{\theta}_4\dot{\theta}_5 + 2D_{146}\dot{\theta}_4\dot{\theta}_6 + 2D_{156}\dot{\theta}_5\dot{\theta}_6 \\
& + D_1
\end{aligned} \tag{39}$$

$$\begin{aligned}
T_2 = & D_{21}\ddot{\theta}_1 + D_{22}\ddot{\theta}_2 + D_{23}\ddot{\theta}_3 + D_{24}\ddot{\theta}_4 + D_{25}\ddot{\theta}_5 + D_{26}\ddot{\theta}_6 + I_{2(act)}\ddot{\theta}_2 \\
& + D_{211}\dot{\theta}_1^2 + D_{222}\dot{\theta}_2^2 + D_{233}\dot{\theta}_3^2 + D_{244}\dot{\theta}_4^2 + D_{255}\dot{\theta}_5^2 + D_{266}\dot{\theta}_6^2 \\
& + 2D_{212}\dot{\theta}_1\dot{\theta}_2 + 2D_{213}\dot{\theta}_1\dot{\theta}_3 + 2D_{214}\dot{\theta}_1\dot{\theta}_4 + 2D_{215}\dot{\theta}_1\dot{\theta}_5 + 2D_{216}\dot{\theta}_1\dot{\theta}_6 \\
& + 2D_{223}\dot{\theta}_2\dot{\theta}_3 + 2D_{224}\dot{\theta}_2\dot{\theta}_4 + 2D_{225}\dot{\theta}_2\dot{\theta}_5 + 2D_{226}\dot{\theta}_2\dot{\theta}_6 + 2D_{234}\dot{\theta}_3\dot{\theta}_4 \\
& + 2D_{235}\dot{\theta}_3\dot{\theta}_5 + 2D_{236}\dot{\theta}_3\dot{\theta}_6 + 2D_{245}\dot{\theta}_4\dot{\theta}_5 + 2D_{246}\dot{\theta}_4\dot{\theta}_6 + 2D_{256}\dot{\theta}_5\dot{\theta}_6 \\
& + D_2
\end{aligned} \tag{40}$$

$$\begin{aligned}
T_3 = & D_{31}\ddot{\theta}_1 + D_{32}\ddot{\theta}_2 + D_{33}\ddot{\theta}_3 + D_{34}\ddot{\theta}_4 + D_{35}\ddot{\theta}_5 + D_{36}\ddot{\theta}_6 + I_{3(act)}\ddot{\theta}_3 \\
& + D_{311}\dot{\theta}_1^2 + D_{322}\dot{\theta}_2^2 + D_{333}\dot{\theta}_3^2 + D_{344}\dot{\theta}_4^2 + D_{355}\dot{\theta}_5^2 + D_{366}\dot{\theta}_6^2 \\
& + 2D_{312}\dot{\theta}_1\dot{\theta}_2 + 2D_{313}\dot{\theta}_1\dot{\theta}_3 + 2D_{314}\dot{\theta}_1\dot{\theta}_4 + 2D_{315}\dot{\theta}_1\dot{\theta}_5 + 2D_{316}\dot{\theta}_1\dot{\theta}_6 \\
& + 2D_{323}\dot{\theta}_2\dot{\theta}_3 + 2D_{324}\dot{\theta}_2\dot{\theta}_4 + 2D_{325}\dot{\theta}_2\dot{\theta}_5 + 2D_{326}\dot{\theta}_2\dot{\theta}_6 + 2D_{334}\dot{\theta}_3\dot{\theta}_4 \\
& + 2D_{335}\dot{\theta}_3\dot{\theta}_5 + 2D_{336}\dot{\theta}_3\dot{\theta}_6 + 2D_{345}\dot{\theta}_4\dot{\theta}_5 + 2D_{346}\dot{\theta}_4\dot{\theta}_6 + 2D_{356}\dot{\theta}_5\dot{\theta}_6 \\
& + D_3
\end{aligned} \tag{41}$$

$$\begin{aligned}
T_4 = & D_{41}\ddot{\theta}_1 + D_{42}\ddot{\theta}_2 + D_{43}\ddot{\theta}_3 + D_{44}\ddot{\theta}_4 + D_{45}\ddot{\theta}_5 + D_{46}\ddot{\theta}_6 + I_{4(act)}\ddot{\theta}_4 \\
& + D_{411}\dot{\theta}_1^2 + D_{422}\dot{\theta}_2^2 + D_{433}\dot{\theta}_3^2 + D_{444}\dot{\theta}_4^2 + D_{455}\dot{\theta}_5^2 + D_{466}\dot{\theta}_6^2 \\
& + 2D_{412}\dot{\theta}_1\dot{\theta}_2 + 2D_{413}\dot{\theta}_1\dot{\theta}_3 + 2D_{414}\dot{\theta}_1\dot{\theta}_4 + 2D_{415}\dot{\theta}_1\dot{\theta}_5 + 2D_{416}\dot{\theta}_1\dot{\theta}_6 \\
& + 2D_{423}\dot{\theta}_2\dot{\theta}_3 + 2D_{424}\dot{\theta}_2\dot{\theta}_4 + 2D_{425}\dot{\theta}_2\dot{\theta}_5 + 2D_{426}\dot{\theta}_2\dot{\theta}_6 + 2D_{434}\dot{\theta}_3\dot{\theta}_4 \\
& + 2D_{435}\dot{\theta}_3\dot{\theta}_5 + 2D_{436}\dot{\theta}_3\dot{\theta}_6 + 2D_{445}\dot{\theta}_4\dot{\theta}_5 + 2D_{446}\dot{\theta}_4\dot{\theta}_6 + 2D_{456}\dot{\theta}_5\dot{\theta}_6 \\
& + D_4
\end{aligned} \tag{42}$$

$$\begin{aligned}
T_5 = & D_{51}\ddot{\theta}_1 + D_{52}\ddot{\theta}_2 + D_{53}\ddot{\theta}_3 + D_{54}\ddot{\theta}_4 + D_{55}\ddot{\theta}_5 + D_{56}\ddot{\theta}_6 + I_{5(act)}\ddot{\theta}_5 \\
& + D_{511}\dot{\theta}_1^2 + D_{522}\dot{\theta}_2^2 + D_{533}\dot{\theta}_3^2 + D_{544}\dot{\theta}_4^2 + D_{555}\dot{\theta}_5^2 + D_{566}\dot{\theta}_6^2 \\
& + 2D_{512}\dot{\theta}_1\dot{\theta}_2 + 2D_{513}\dot{\theta}_1\dot{\theta}_3 + 2D_{514}\dot{\theta}_1\dot{\theta}_4 + 2D_{515}\dot{\theta}_1\dot{\theta}_5 + 2D_{516}\dot{\theta}_1\dot{\theta}_6 \\
& + 2D_{523}\dot{\theta}_2\dot{\theta}_3 + 2D_{524}\dot{\theta}_2\dot{\theta}_4 + 2D_{525}\dot{\theta}_2\dot{\theta}_5 + 2D_{526}\dot{\theta}_2\dot{\theta}_6 + 2D_{534}\dot{\theta}_3\dot{\theta}_4 \\
& + 2D_{535}\dot{\theta}_3\dot{\theta}_5 + 2D_{536}\dot{\theta}_3\dot{\theta}_6 + 2D_{545}\dot{\theta}_4\dot{\theta}_5 + 2D_{546}\dot{\theta}_4\dot{\theta}_6 + 2D_{556}\dot{\theta}_5\dot{\theta}_6 \\
& + D_5
\end{aligned} \tag{43}$$

$$\begin{aligned}
T_6 = & D_{61}\ddot{\theta}_1 + D_{62}\ddot{\theta}_2 + D_{63}\ddot{\theta}_3 + D_{64}\ddot{\theta}_4 + D_{65}\ddot{\theta}_5 + D_{66}\ddot{\theta}_6 + I_{6(act)}\ddot{\theta}_6 \\
& + D_{611}\dot{\theta}_1^2 + D_{622}\dot{\theta}_2^2 + D_{633}\dot{\theta}_3^2 + D_{644}\dot{\theta}_4^2 + D_{655}\dot{\theta}_5^2 + D_{666}\dot{\theta}_6^2 \\
& + 2D_{612}\dot{\theta}_1\dot{\theta}_2 + 2D_{613}\dot{\theta}_1\dot{\theta}_3 + 2D_{614}\dot{\theta}_1\dot{\theta}_4 + 2D_{615}\dot{\theta}_1\dot{\theta}_5 + 2D_{616}\dot{\theta}_1\dot{\theta}_6 \\
& + 2D_{623}\dot{\theta}_2\dot{\theta}_3 + 2D_{624}\dot{\theta}_2\dot{\theta}_4 + 2D_{625}\dot{\theta}_2\dot{\theta}_5 + 2D_{626}\dot{\theta}_2\dot{\theta}_6 + 2D_{634}\dot{\theta}_3\dot{\theta}_4 \\
& + 2D_{635}\dot{\theta}_3\dot{\theta}_5 + 2D_{636}\dot{\theta}_3\dot{\theta}_6 + 2D_{645}\dot{\theta}_4\dot{\theta}_5 + 2D_{646}\dot{\theta}_4\dot{\theta}_6 + 2D_{656}\dot{\theta}_5\dot{\theta}_6 \\
& + D_6
\end{aligned} \tag{44}$$

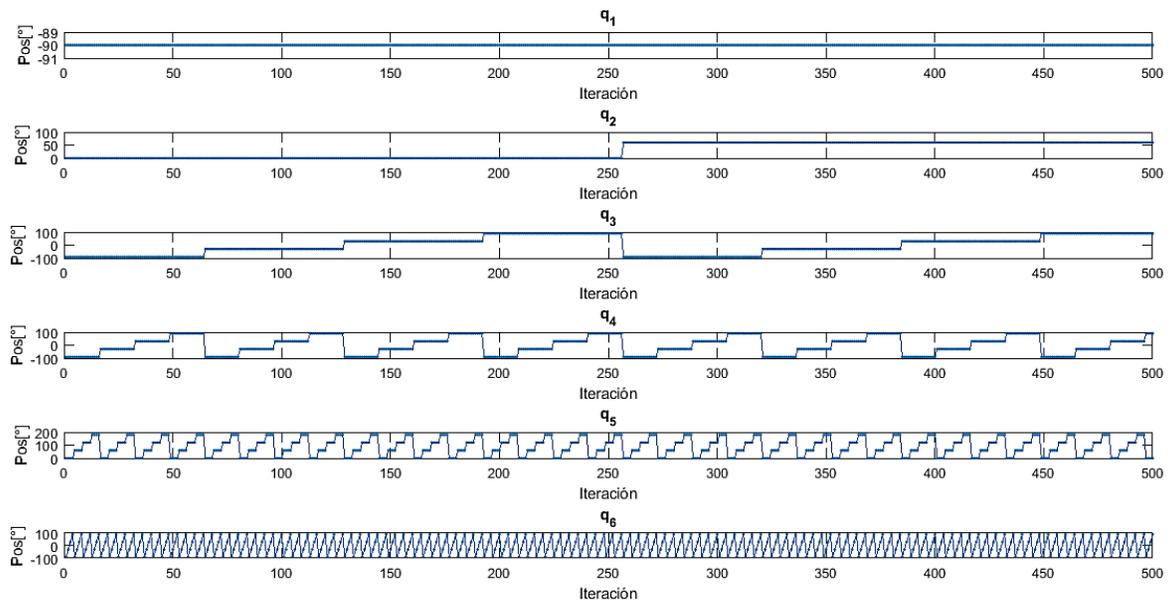
A partir de las Ecuaciones (39)-(44) es posible obtener los valores máximos de torque en cada articulación, que posteriormente se utilizarán para la selección de actuadores. Las propiedades físicas y geométricas de todos los eslabones, parámetros necesarios para los cálculos dinámicos posteriores, se describen en la Tabla 9.

**Tabla 9**

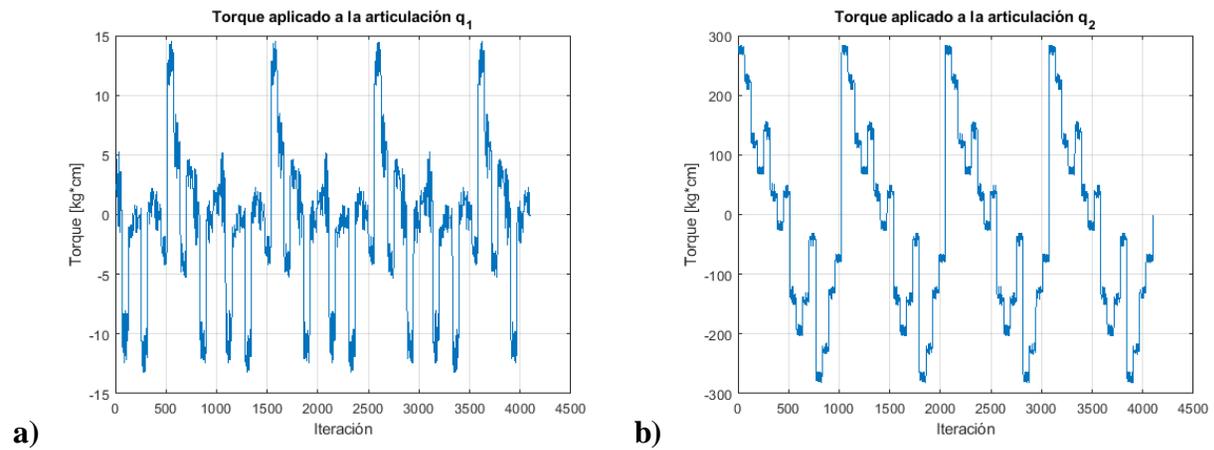
Propiedades físicas y geométricas de los eslabones del manipulador robótico

Propiedades	Eslabones					
	L <sub>1</sub>	L <sub>2</sub>	L <sub>3</sub>	L <sub>4</sub>	L <sub>5</sub>	L <sub>6</sub>
longitud [mm]	219.5	175	155	75	26	135
masa [g]	1494	295.44	208.78	213.86	92.89	150
$I_{xx}$ [ $g * mm^2$ ]	21864331.37	303621.85	1010177.71	1051341.50	25714.96	980000
$I_{yx}$ [ $g * mm^2$ ]	-387841.60	30043.66	39993.95	2.26	14121.45	0
$I_{zx}$ [ $g * mm^2$ ]	-516082.03	-21844.16	910.58	13.59	0.08	0
$I_{xy}$ [ $g * mm^2$ ]	-387841.60	30043.66	39993.95	2.26	14121.45	0
$I_{yy}$ [ $g * mm^2$ ]	19593824.45	3638148.85	90289.94	775157.29	46078.05	980000
$I_{zy}$ [ $g * mm^2$ ]	-555641.31	-7578.78	-3131.31	119.41	0.06	0
$I_{xz}$ [ $g * mm^2$ ]	-516082.03	-21844.16	910.58	13.59	0.08	0
$I_{yz}$ [ $g * mm^2$ ]	-555641.31	-757887	-3131.31	119.41	0.06	0
$I_{zz}$ [ $g * mm^2$ ]	4308289.72	3635816.38	1034663.04	303477.89	60476.09	1250000
$\bar{x}$ [mm]	-3.16	-88.84	5.29	0.04	-7.61	0
$\bar{y}$ [mm]	-92.22	1.69	0.31	-29.49	0	0
$\bar{z}$ [mm]	3.77	-0.47	-100.99	0.01	-37.78	-65

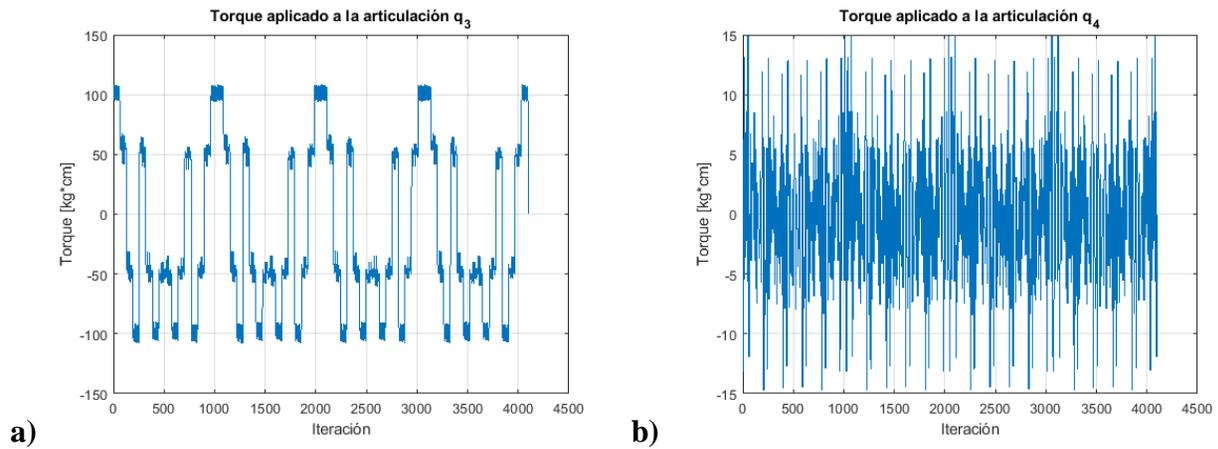
Se desarrolló un algoritmo en Matlab (Anexo A) que permite obtener de manera gráfica el torque aplicado en cada articulación a través de movimientos cíclicos dentro del volumen de trabajo, tal como se describe en la Figura 17. A continuación se presentan los resultados del torque producido para las variables articulares  $q_1$  (Figura 18a),  $q_2$  (Figura 18b),  $q_3$  (Figura 19a),  $q_4$  (Figura 19b),  $q_5$  (Figura 20a) y  $q_6$  (Figura 20b), respectivamente.



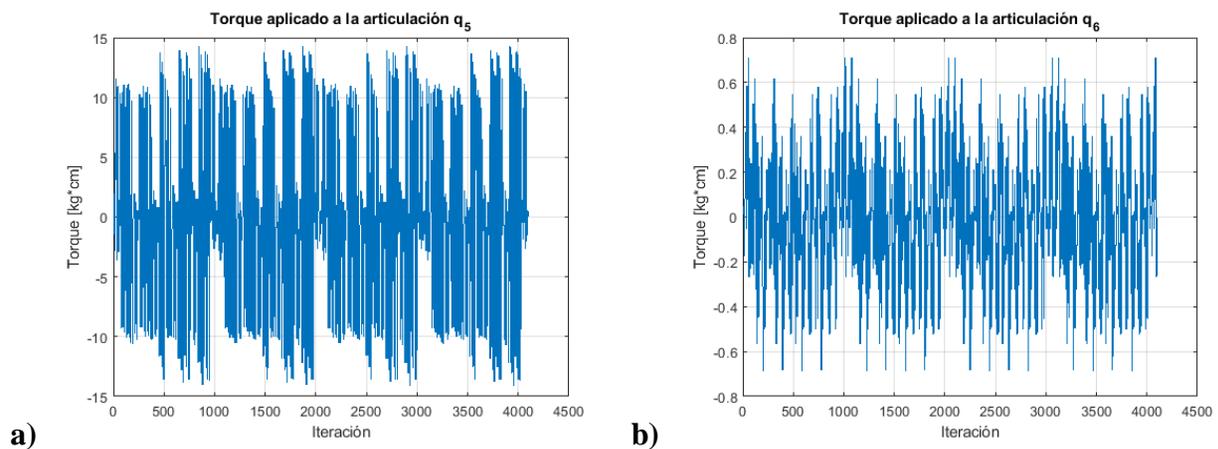
**Figura 17.** Posiciones de las variables articulares para el cálculo de torque máximo



**Figura 18.** Torque aplicado para cada variable articular. a)  $q_1$ , b)  $q_2$



**Figura 19.** Torque aplicado para cada variable articular. a)  $q_3$ , b)  $q_4$



**Figura 20.** Torque aplicado para cada variable articular. a)  $q_5$ , b)  $q_6$

Los resultados de la simulación desarrollada se obtiene a partir de los picos presentes en las gráficas de torque para cada articulación (Figura 18-Figura 20), y se presentan resumidos en la Tabla 10.

**Tabla 10***Resultados de torque máximo*

Articulación	Torque máximo [ $kg * cm$ ]
$q_1$	16.7
$q_2$	302.8
$q_3$	123.3
$q_4$	20.4
$q_5$	19.6
$q_6$	0.7

**B. Selección y dimensionamiento de actuadores**

A partir de los requerimientos establecidos para el subsistema mecánico, en la Tabla 11 se presentan los criterios de diseño a tomar en consideración para la selección de actuadores con sus respectivas ponderaciones. Los distintos conceptos desarrollados se presentan en la Tabla 12.

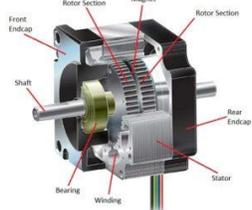
**Tabla 11***Ponderación de criterios de diseño para la selección de actuadores*

	Costo	Disponibilidad	Precisión	$\Sigma+1$	Ponderación
<b>Costo</b>		0	0.5	1.5	0.25
<b>Disponibilidad</b>	1		0.5	2.5	0.42
<b>Precisión</b>	0.5	0.5		2	0.33
<b>TOTAL</b>				6	1

**Tabla 12***Conceptos para la selección de actuadores*

Concepto	Descripción	Características	Ilustración
1	Servomotor	<ul style="list-style-type: none"> <li>Alimentación en el rango de 4 – 6V DC.</li> <li>Resolución en el rango de 1 a 10 grados.</li> <li>Peso en el rango entre 15 y 200 gramos.</li> <li>Control en lazo cerrado.</li> <li>El consumo de energía es proporcional al torque de carga aplicado al motor.</li> <li>El torque no se reduce en función de la velocidad.</li> </ul>	

**CONTINÚA**

2	Motor a pasos	<ul style="list-style-type: none"> <li>• Motor eléctrico síncrono sin escobillas que convierte los pulsos digitales en rotación mecánica del eje.</li> <li>• Se puede alcanzar una amplia gama de velocidades de rotación, ya que la velocidad es proporcional a la frecuencia de los pulsos de entrada.</li> <li>• Requiere altas corrientes aplicadas en todo momento, incluso cuando están estacionados con poca o ninguna carga.</li> <li>• El torque se reduce con respecto al incremento de velocidad.</li> <li>• Resolución: 1.8°.</li> <li>• Precisión: 3 -5%.</li> <li>• Control en lazo abierto.</li> </ul>	
3	Motor DC con caja reductora	<ul style="list-style-type: none"> <li>• La adición de una caja de engranes a un motor reduce la velocidad al tiempo que aumenta el torque a la salida.</li> <li>• El torque se reduce con respecto al incremento de velocidad.</li> </ul>	

Luego del análisis de cada concepto con respecto a los criterios de diseño seleccionados, en la Tabla 13 se presentan las ponderaciones obtenidas para cada solución. Se concluye que el tipo de actuador que mejor se adapta a los requerimientos del proyecto corresponde al uso de servomotores en todas las articulaciones.

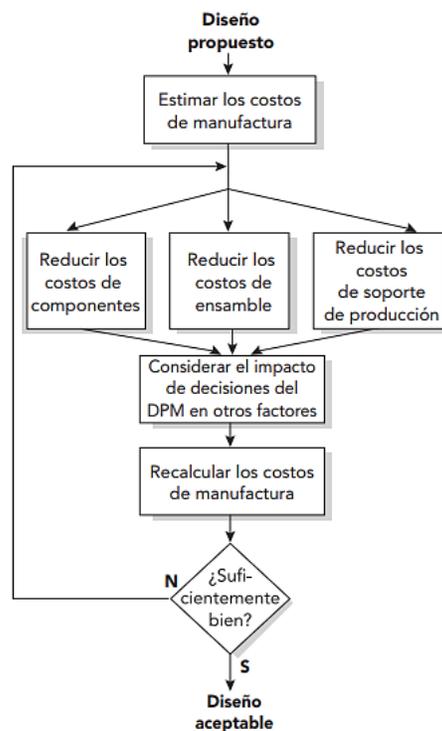
**Tabla 13**

*Matriz de decisión para la selección de actuadores*

	Criterios de diseño									Total
	Costo			Disponibilidad			Precisión			
Concepto	W	S	T	W	S	T	W	S	T	
Concepto 1	0.25	5	1.25	0.42	9	3.75	0.33	8	2.67	7.67
Concepto 2	0.25	7	1.75	0.42	6	2.50	0.33	7	2.33	6.58
Concepto 3	0.25	8	2.00	0.42	5	2.08	0.33	5	1.67	5.75

### C. Diseño CAD

Para el diseño mecánico del prototipo, se tomó en consideración el proceso de diseño para manufactura desarrollado en (Ulrich, Karl & Eppinger, 2013). El ciclo de diseño se presenta en la Figura 21. Uno de los requerimientos importantes radica en el hecho de que la mayor parte de los componentes deben ser estándar y de fácil acceso. Como producto de las interacciones propias del proceso de diseño se obtuvo un diseño cuya evolución puede apreciarse en la Tabla 14.



**Figura 21.** Método de diseño para manufactura

FUENTE: (Ulrich, Karl & Eppinger, 2013)

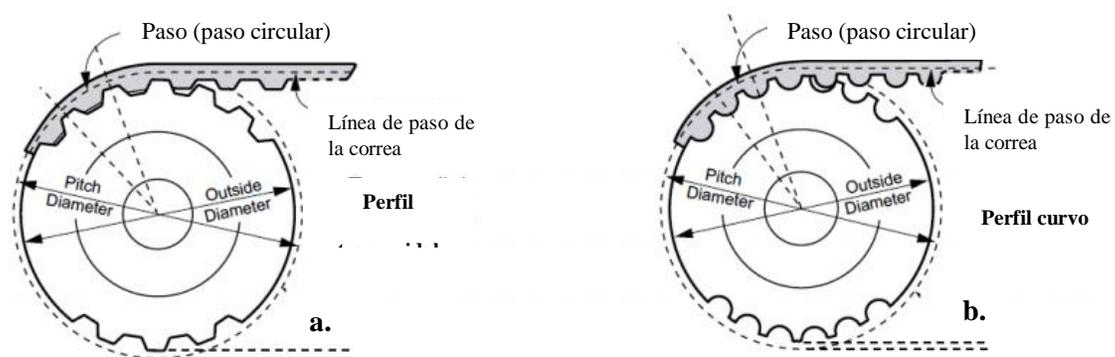
**Tabla 14***Evolución de diseño mecánico*

Versión	Descripción	Ilustración
1	<ul style="list-style-type: none"> <li>• Enfocado en reducción del peso para maximizar relación de carga útil.</li> <li>• Transmisión directa.</li> <li>• Asimétrico.</li> </ul>	
2	<ul style="list-style-type: none"> <li>• Se realizaron mejoras en la estructura para obtener un modelo simétrico, mejorando la estabilidad general de la estructura.</li> <li>• Se reubicaron los motores para reducir su influencia en la dinámica del sistema.</li> <li>• Transmisión mediante correas dentadas.</li> </ul>	
3	<ul style="list-style-type: none"> <li>• Muñeca rediseñada para adaptarse a componentes comerciales.</li> <li>• Estructura adaptada para alojar los motores seleccionados.</li> <li>• Mejoras estéticas y de optimización de espacio.</li> </ul>	
4	<ul style="list-style-type: none"> <li>• Diseño reformado basado en diseño para la manufactura.</li> <li>• Piezas grandes divididas para facilitar su fabricación.</li> <li>• La base se adaptó para alojar los componentes electrónicos que formarán parte del robot.</li> <li>• El diseño toma en consideración el espacio necesario para ocultar el cableado internamente.</li> </ul>	

## D. Diseño de poleas dentadas

El tipo de correas dentadas que se obtienen comercialmente son del tipo GT2 con perfil circular. Algunas de las ventajas del perfil circular (Figura 22b) sobre el perfil trapezoidal (Figura 22a), de acuerdo con (Designatronics Inc., 2013), incluyen:

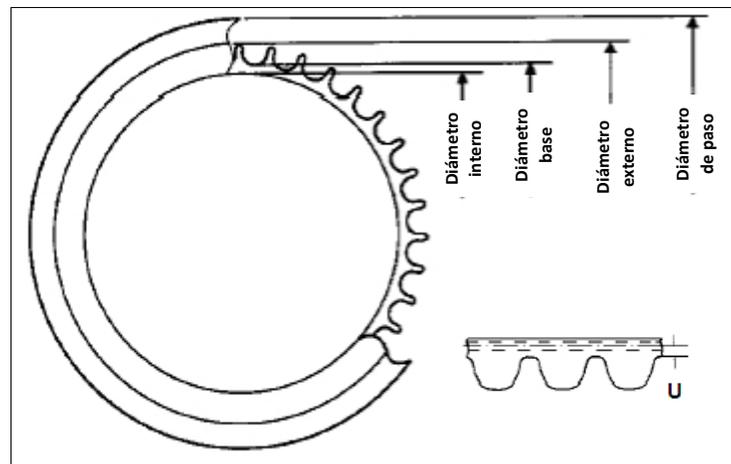
- Los dientes proporcionalmente más profundos ayudan a prevenir que los dientes salten mientras operan.
- Construcción más liviana, con menor pérdida centrífuga.
- Menor presión por unidad sobre el diente, ya que el área de contacto es mayor.
- Mayor resistencia en cortante debido a una mayor sección transversal del diente.
- Más barato, ya que una correa más estrecha soportará la misma carga.
- Más eficiente energéticamente, particularmente si se reemplaza una transmisión de correa en "V", que incurre en pérdidas de energía debido al deslizamiento.
- Tensión de instalación pequeña, lo que reduce las cargas en los rodamientos.



**Figura 22.** Perfiles para correas sincrónicas. a) Perfil trapezoidal b) Perfil curvo

FUENTE: (Designatronics Inc., 2013)

Comercialmente puede encontrarse un tipo de correa con perfil circular de bajo costo, conocida como GT2. Para obtener las relaciones de transmisión requeridas y facilitar la fabricación y ensamblaje, las poleas serán también diseñadas e impresas en 3D. Para tal efecto, se requiere diseñar poleas de 20, 40 y 60 dientes, de acuerdo con la recomendaciones obtenidas de (Designatronics Inc., 2018). La interpretación geométrica de los elementos desarrollados en las fórmulas se describen en la Figura 23. Los parámetros obtenidos para el diseño de las poleas se resume en la Tabla 15.



**Figura 23.** Términos utilizados en el diseño de perfiles de poleas sincrónicas

FUENTE: (Designatronics Inc., 2018)

$$PD = \frac{\text{paso} * \#dientes}{\pi} \quad (45)$$

$$OD = PD - 2U \quad (46)$$

$$RD = OD - 2GD \quad (47)$$

$$ID = RD - PF \quad (48)$$

Donde:

*PD: Diámetro de paso.*

*OD: Diámetro externo.*

*U: Distancia desde la línea de paso hasta el fondo del diente de la banda.*

*ID: Diámetro interno máximo.*

*RD: Diámetro base.*

*GD: Profundidad de diente.*

*PF: Factor de paso.*

**Tabla 15**

*Dimensiones de poleas dentadas*

Número de dientes	Paso [mm]	Factor de paso [mm]	Profundidad de diente [mm]	U [mm]	Diámetro de paso [mm]	Diámetro externo [mm]	Diámetro base [mm]	Diámetro interno máximo [mm]
<b>20</b>	2	0.41	0.76	0.25	12.73	12.22	10.70	10.29
<b>40</b>	2	0.41	0.76	0.25	25.46	24.96	23.44	23.03
<b>60</b>	2	0.41	0.76	0.25	38.20	37.69	36.17	35.76

### 3.2.3 Implementación

El proceso de impresión 3D de todos los componentes mecánicos se realizó en una impresora Prusa I3 (Geetech, 2016) con el software Ultimaker Cura 3.5.1 (Ultimaker B.V., 2018). Las especificaciones técnicas de la impresora utilizada se muestran en la Tabla 16. Los parámetros de impresión utilizados se presentan en la Tabla 17. Los tiempos de impresión requeridos para cada una de las piezas se muestran en la Tabla 18.

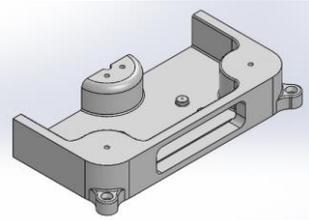
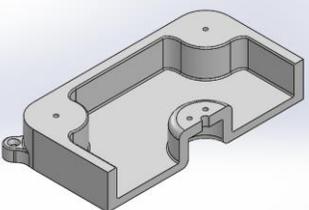
**Tabla 16***Prusa I3 X. Especificaciones técnicas*

Parámetro	Valor
Tecnología de impresión	Fused filament fabrication (FFF)
Diámetro de filamento	1.75 mm
Diámetro de extrusor	0.4 mm
Resolución	0.1 mm
Volumen de impresión	200*200*200 mm

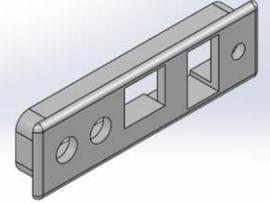
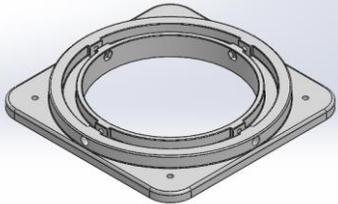
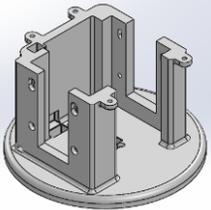
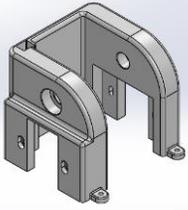
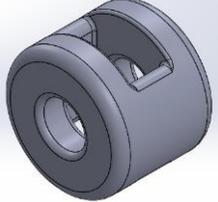
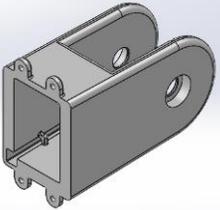
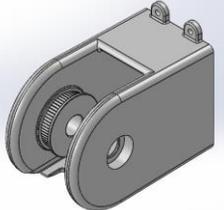
**Tabla 17***Parámetros de impresión 3D*

Parámetro	Valor
Altura de capa	0.2 mm
Grosor de la pared	0.15 mm
Densidad de relleno	20 %
Velocidad de impresión	60 mm/s
Material	PLA

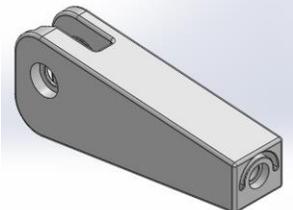
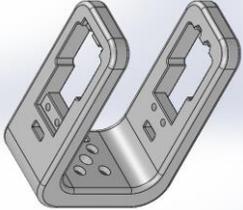
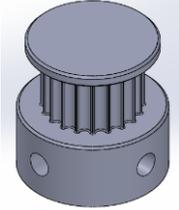
**Tabla 18***Tiempos de impresión*

Pieza	Tiempo de impresión	Material [g]	Ilustración
Base 1/2	6h 56min	115	
Base 2/2	7h 12min	116	

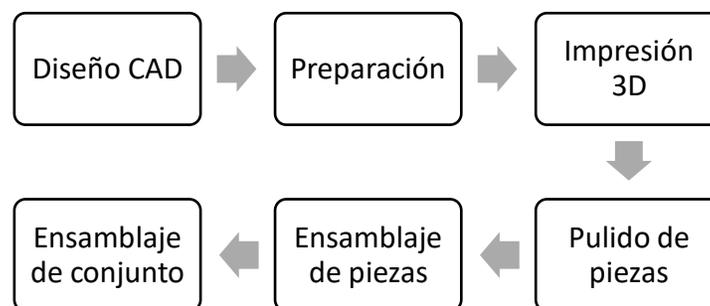
CONTINÚA

Cara base	1h 18min	15	
Tapa base	13h 58min	218	
Eslabón 1 1/2	17h 40min	281	
Eslabón 1 2/2	13h 08min	224	
Acople banda	0h 42min	10	
Eslabón 2 1/2	9h 04min	169	
Eslabón 2 2/2	9h 14min	178	

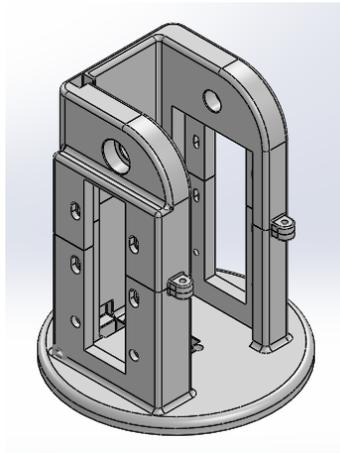
CONTINÚA

Eslabón 3	7h 43min	145	
Muñeca	5h 11min	79	
GT2_20T (x3)	0h 40 min	10	
<b>TOTAL:</b>	<b>92h 46min</b>	<b>1560</b>	

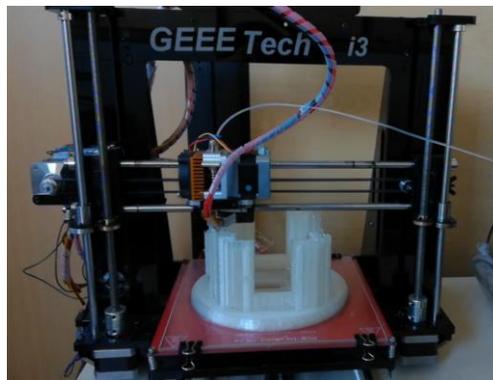
El proceso de implementación del subsistema mecánico se resume en la Figura 24. Un ejemplo de la ejecución del proceso descrito se ilustra a continuación (Figura 25 - Figura 30). El subsistema mecánico ensamblado se presenta en la Figura 31.



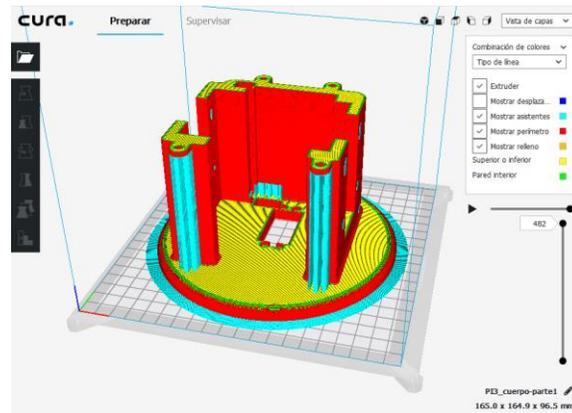
**Figura 24.** Proceso de implementación de subsistema mecánico



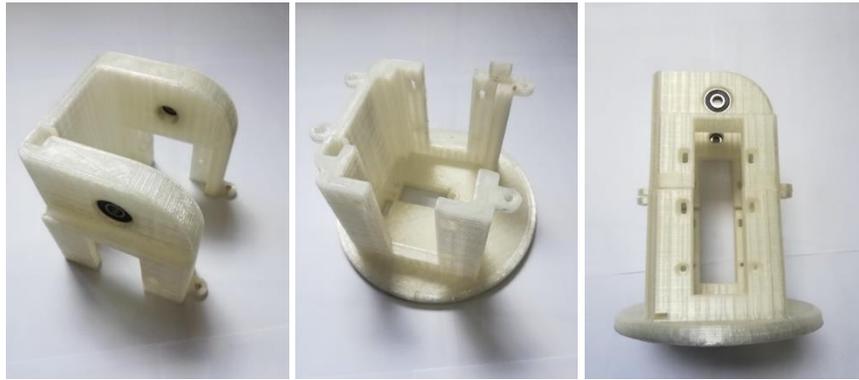
**Figura 25.** Eslabón 1. Diseño CAD



**Figura 26.** Eslabón 1. Impresión 3D



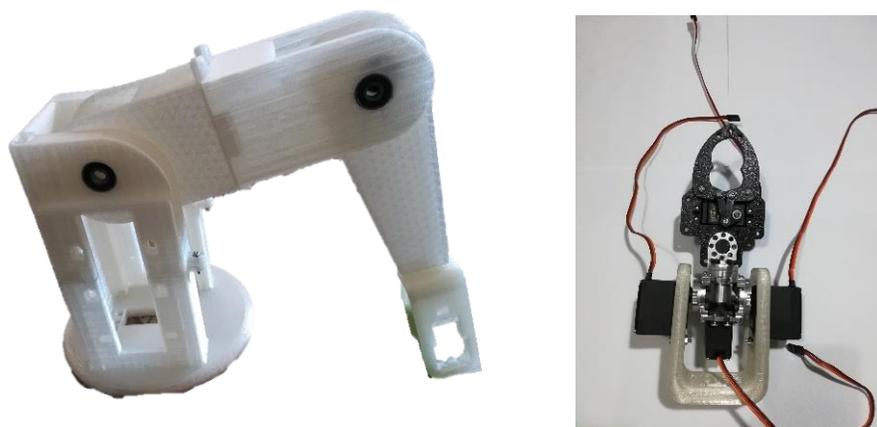
**Figura 27.** Eslabón 1. Simulación CAM para impresión 3D



**Figura 28.** Eslabón 1. Ensamblaje de piezas



**Figura 29.** Partes impresas en 3D



**Figura 30.** Ensamblaje del subsistema mecánico



*Figura 31.* Ensamblaje de conjunto

### 3.3 Diseño del subsistema eléctrico y/o electrónico

#### 3.3.1 Requerimientos

##### A. Identificación de necesidades

Los requerimientos generales que debe cumplir el subsistema electrónico se presentan en la Tabla 19, y se basan en la necesidad de obtener un sistema modular, independiente y de bajo costo.

**Tabla 19**

*Subsistema electrónico. Identificación de necesidades*

	<b>Requerimiento</b>	<b>Prioridad</b>
<b>1</b>	El cableado debe ser sencillo de conectar para el usuario	2
<b>2</b>	El diseño debe ser modular	1
<b>3</b>	El sistema debe ser de fácil mantenimiento	4
<b>4</b>	Solución de bajo costo	2
<b>5</b>	Compatible con plataforma ROS	1
<b>6</b>	Sistema independiente	1

## B. Lista de métricas

Los requerimientos se asocian con una variable que luego pueda ser evaluada para determinar el grado de cumplimientos de los objetivos planteados para el desarrollo del subsistema electrónico. Las métricas establecidas para cada necesidad planteada se presentan en la Tabla 20.

**Tabla 20**

*Lista de métricas para el subsistema electrónico*

	Necesidades	Métrica	Importancia	Unidades
1	1	Facilidad de ensamblaje	1	Subj.
2	3	Facilidad de mantenimiento	3	Subj.
3	4	Costo	2	Dólares
4	2, 6	Modularidad	2	Subj.
5	5	Sistema operativo del robot	1	Subj.

## C. Matriz de necesidades – métricas

La matriz de necesidades-métricas para el subsistema electrónico se presenta en la Tabla 21.

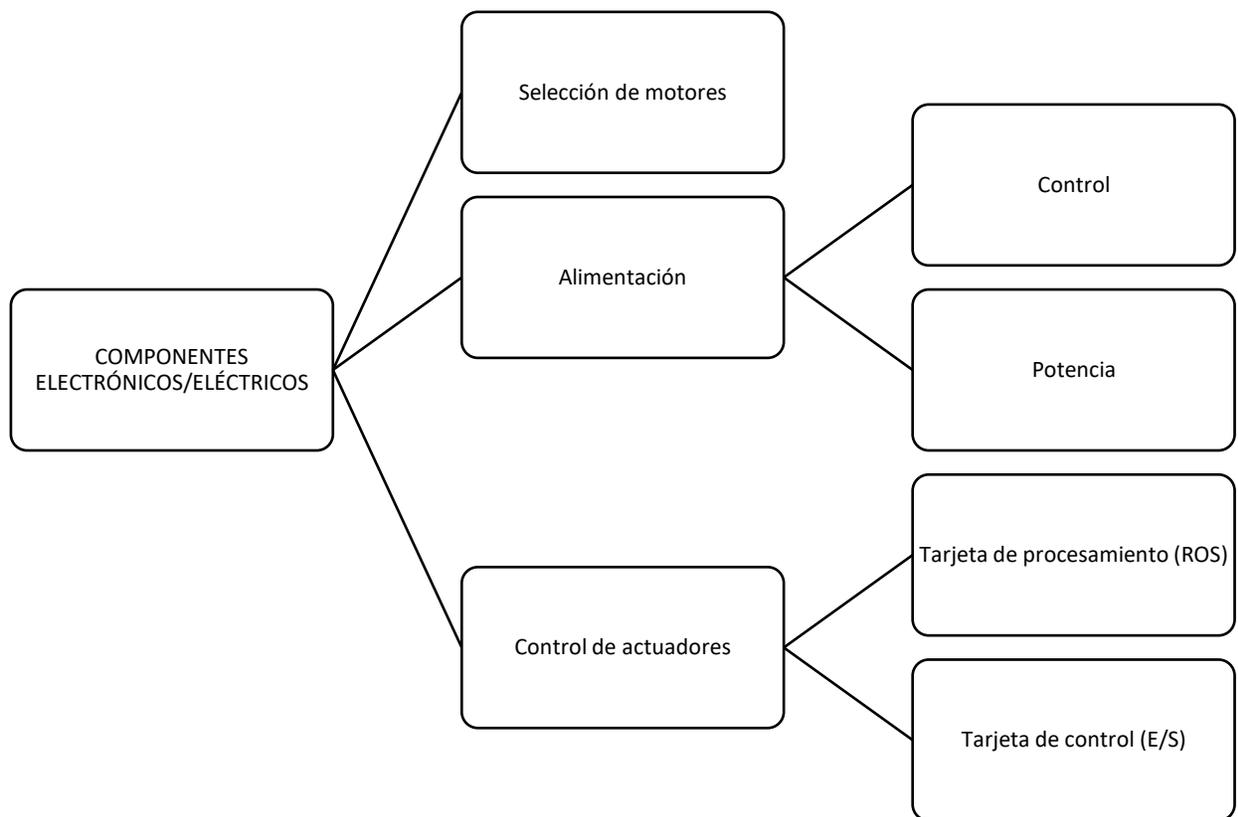
**Tabla 21**

*Matriz de necesidades – métricas. Subsistema electrónico*

		1	2	3	4	5
Métrica		Facilidad de ensamblaje	Facilidad de mantenimiento	Costo	Modularidad	Sistema operativo del robot
Necesidad						
1	El cableado debe ser sencillo de conectar para el usuario	*				
2	El diseño debe ser modular				*	
3	El sistema debe ser de fácil mantenimiento		*			
4	Solución de bajo costo			*		
5	Compatible con plataforma ROS					*
6	Sistema independiente				*	

### 3.3.2 Diseño

El diseño del subsistema electrónico requiere de la consideración de varias etapas, las mismas que se indican en la Figura 32. Se requiere garantizar la alimentación para los elementos de control y potencia, a partir de los parámetros de corriente y voltaje requeridos. Los componentes electrónicos para el control de los actuadores se implementarán tomando en consideración los requerimientos de procesamiento de datos y las características de control específicas de cada actuador. Se realizarán las adecuaciones necesarias para garantizar la compatibilidad entre la tarjeta de procesamiento y los actuadores seleccionados a través de una tarjeta de control.



**Figura 32.** Componentes del subsistema electrónico

## A. Selección de motores

Los motores a utilizar se seleccionan de acuerdo a los requerimientos de torque obtenidos a partir del modelamiento matemático. Los modelos comercialmente disponibles seleccionados se describen en la Tabla 22.

**Tabla 22**  
*Selección de servomotores*

Componente	Cantidad	Especificaciones	Ilustración
Servo con caja reductora TM-805MG-400	2	<ul style="list-style-type: none"> <li>• Peso: 359 [g]</li> <li>• Voltaje: 4.8-6 [V]</li> <li>• Torque: 123.5 [kg*cm]</li> <li>• Velocidad: 0.70 [seg/60°] @ 6V</li> <li>• Corriente máxima: 6 [A]</li> <li>• Rotación máxima: 400°</li> <li>• Potenciómetro externo</li> <li>• Relación de transmisión: 5:1</li> </ul> (ServoCity, 2018)	
LewanSoul LD-27MG	1	<ul style="list-style-type: none"> <li>• Peso: 65 [g]</li> <li>• Voltaje: 6-7.4 [V]</li> <li>• Torque: 20 [kg*cm] @ 6.6V</li> <li>• Velocidad: 0.16 [seg/60°] @ 7.4V</li> <li>• Corriente máxima: 1 [A]</li> <li>• Rotación máxima: 270°</li> </ul> (LewanSoul, 2017)	
LewanSoul LD-220MG	1	<ul style="list-style-type: none"> <li>• Peso: 65 [g]</li> <li>• Voltaje: 6-7.4 [V]</li> <li>• Torque: 20 [kg*cm] @ 6.6V</li> <li>• Velocidad: 0.16 [seg/60°] @ 7.4V</li> <li>• Corriente máxima: 1 [A]</li> <li>• Rotación máxima: 180°</li> </ul> (LewanSoul, 2017)	

CONTINÚA

TowerPro MG946R	4	<ul style="list-style-type: none"> <li>• Peso: 55 [g]</li> <li>• Voltaje: 4.8-6.6 [V]</li> <li>• Torque: 13[ kg*cm] @ 6V</li> <li>• Velocidad: 0.17 [seg/60°] @ 6V</li> <li>• Corriente máxima: 1.2 [A]</li> <li>• Rotación máxima: 180°</li> </ul> <p><b>(Torq Pro &amp; Tower Pro, 2014)</b></p>	
--------------------	---	--	--

## B. Alimentación

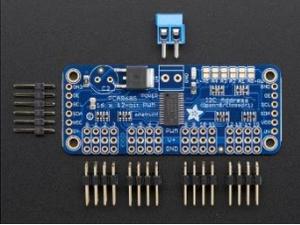
Tomando en consideración los motores seleccionados, ahora se conocen los requerimientos de potencia del sistema y el número de salidas de control requeridas. Se requiere, por lo tanto, un mínimo de 10 amperios de corriente para controlar 8 servomotores simultáneamente. En base a los requerimiento presentados y disponibilidad de componentes, se seleccionan los elementos electrónicos de acuerdo con la Tabla 23.

**Tabla 23**

*Lista de componentes electrónicos*

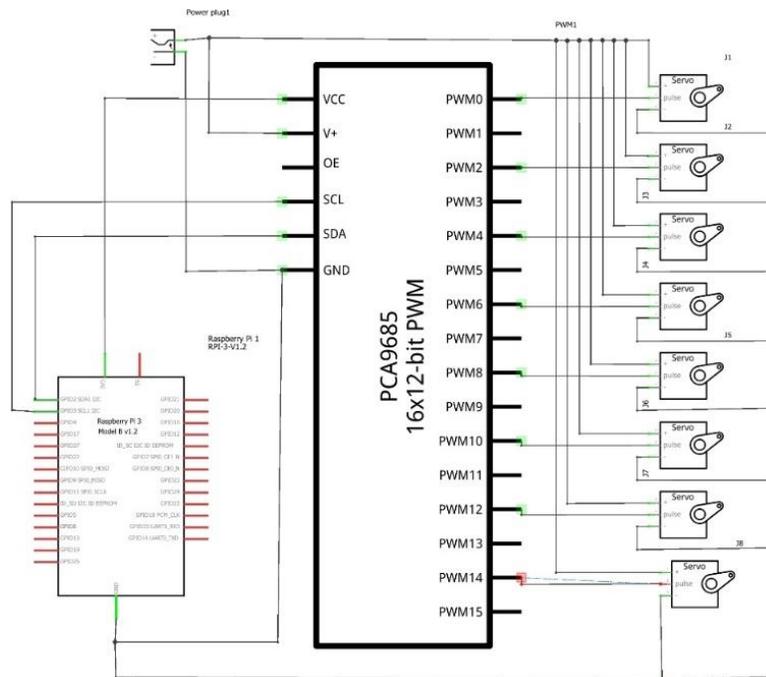
Componente	Especificaciones	Ilustración
Fuente de poder – transformador	<ul style="list-style-type: none"> <li>• Voltaje de entrada: 110/220V AC</li> <li>• Voltaje de salida: 12V DC</li> </ul>	
Convertidor Buck	<ul style="list-style-type: none"> <li>• Voltaje de entrada: 9-40V DC</li> <li>• Voltaje de salida: 6V DC</li> <li>• Corriente de salida: 10A (MAX 20A)</li> <li>• Potencia de salida: 60W (MAX 120W)</li> <li>• Eficacia de conversión:&gt; 93%</li> </ul>	

**CONTINÚA**

<p>Adafruit PCA9685</p>	<ul style="list-style-type: none"> <li>• Controlador PWM controlado por i2c con un reloj incorporado.</li> <li>• VCC:3-5 V.</li> <li>• V+: 6V máx.</li> <li>• Frecuencia: 400-1600 Hz.</li> <li>• Resolución de 12 bits para cada salida: para servos, eso significa aproximadamente 4us de resolución a una velocidad de actualización de 60Hz.</li> <li>• Pin de habilitación de salida para deshabilitar rápidamente todas las salidas.</li> </ul> <p>(Adafruit, 2015)</p>	
<p>Raspberry Pi 3 Model B</p>	<ul style="list-style-type: none"> <li>• Quad Core 1.2GHz.</li> <li>• 1GB RAM.</li> <li>• 40 pines de propósito general.</li> <li>• 4 puertos USB.</li> <li>• Puerto HDMI.</li> <li>• Bluetooth, LAN, Wireless LAN.</li> </ul> <p>(Raspberry Pi Foundation, 2018)</p>	
<p>Fuente de poder Micro USB</p>	<ul style="list-style-type: none"> <li>• Voltaje de entrada: 100-240V AC</li> <li>• Voltaje de salida: 5V DC, 2500 mA</li> </ul>	

### C. Control de actuadores

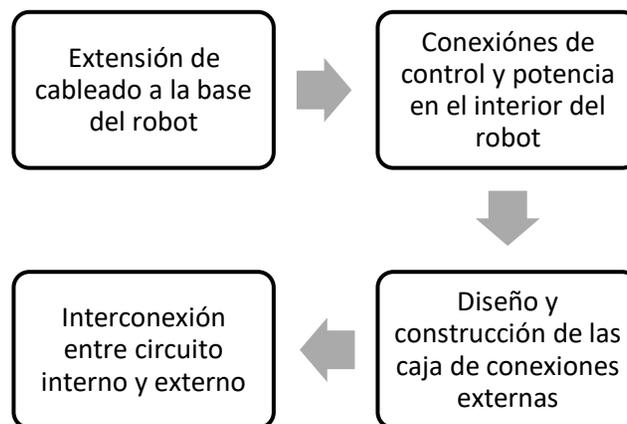
A partir de los componentes seleccionados, se diseña el esquema electrónico para el control de los motores para cada variable articular. El resultado se presenta en la Figura 33.



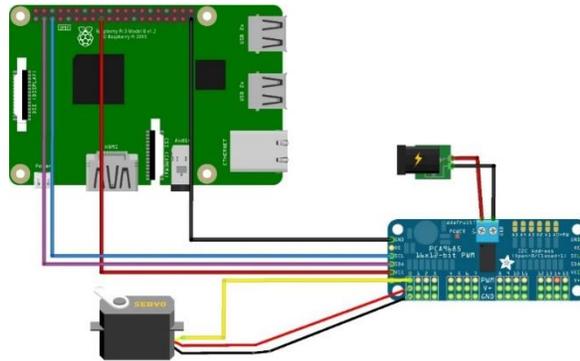
**Figura 33.** Esquema electrónico

### 3.3.3 Implementación

El proceso de implementación del subsistema electrónico se describe en la Figura 34. El diagrama de conexión entre el microprocesador y el módulo de control de los servomotores se ilustra en la Figura 35.

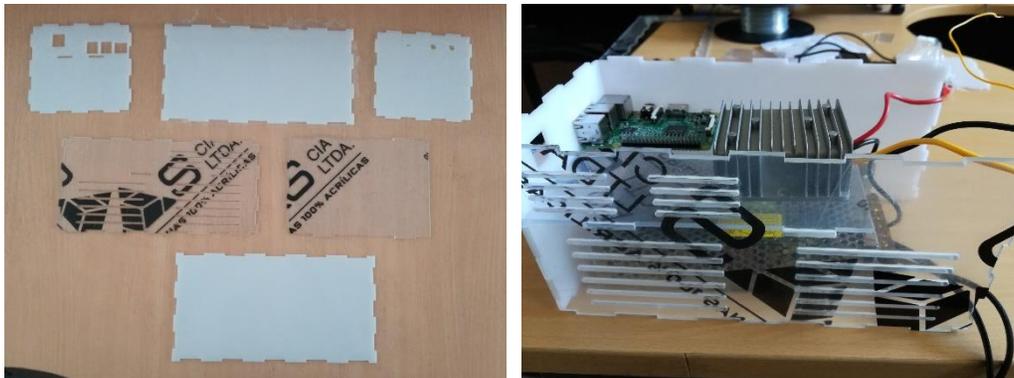


**Figura 34.** Proceso de implementación de subsistema electrónico



**Figura 35.** Diagrama de conexión entre Raspberry Pi y módulo Adafruit PCA9685

La caja de conexiones se diseñó para fabricarse en acrílico mediante el proceso de corte láser, con una tapa lateral transparente que permita apreciar la disposición de los componentes en el interior. El proceso de fabricación y ensamblaje se presenta en la Figura 36.



**Figura 36.** Diseño y construcción de la caja de conexiones

### 3.4 Diseño del subsistema TICs

#### 3.4.1 Requerimientos

##### A. Identificación de necesidades

Los requerimientos generales que debe cumplir el subsistema TICs se presentan en la Tabla 24, y se basan en la necesidad de obtener un sistema modular, independiente y de bajo costo.

**Tabla 24***Subsistema TICs. Identificación de necesidades*

	<b>Requerimiento</b>	<b>Prioridad</b>
<b>1</b>	Interfaz humano-máquina amigable para el usuario	2
<b>2</b>	Modos de operación similares a un manipulador robótico industrial	3
<b>3</b>	Facilidad de programación	1
<b>4</b>	Optimizado para el aprendizaje	1

**B. Lista de métricas**

Los requerimientos se asocian con una variable que luego pueda ser evaluada para determinar el grado de cumplimiento de los objetivos planteados para el desarrollo del subsistema TICs. Las métricas establecidas para cada necesidad planteada se presentan en la Tabla 25.

**Tabla 25***Lista de métricas para el subsistema TICs*

	<b>Necesidades</b>	<b>Métrica</b>	<b>Importancia</b>	<b>Unidades</b>
<b>1</b>	1	Comprensión de interfaz	2	Subj.
<b>2</b>	2	Facilidad de operación	3	Subj.
<b>3</b>	3	Interpretación de código	1	Subj.
<b>4</b>	4	Potencial didáctico	1	Subj.

**C. Matriz de necesidades – métricas**

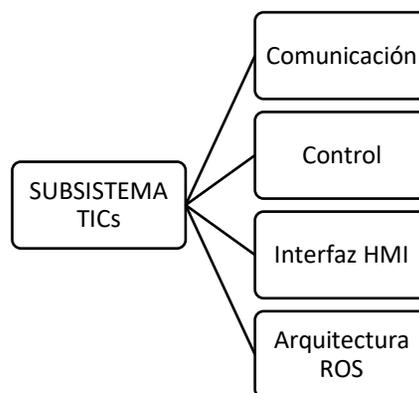
La matriz de necesidades-métricas para el subsistema electrónico se presenta en la Tabla 26.

**Tabla 26***Matriz de necesidades – métricas. Subsistema TICs*

		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
Métrica		Comprensión de interfaz	Facilidad de operación	Interpretación de código	Potencial didáctico
Necesidad					
<b>1</b>	Interfaz humano-máquina amigable para el usuario	*			
<b>2</b>	Modos de operación similares a un manipulador robótico industrial		*		
<b>3</b>	Facilidad de programación			*	
<b>4</b>	Optimizado para el aprendizaje				*

### 3.4.2 Diseño

La Figura 37 resume las etapas que componen el desarrollo del subsistema TICs.



**Figura 37.** Componentes del subsistema TICs

#### A. Comunicación

Con el objetivo de simplificar el número de periféricos requeridos para manipular y programar la plataforma robótica desarrollada, se plantea el uso de un servidor VNC para la visualización y

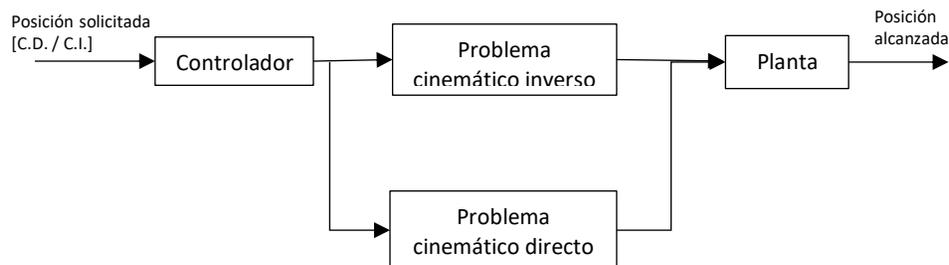
control de los procesos ejecutados en la tarjeta Raspberry Pi 3 desde un computador. Se seleccionó TightVNC (GlavSoft LLC, 2018) como servidor y VNC Viewer (RealVNC Limited, 2018) como visualizador VNC respectivamente. El proceso básico de configuración se describe en la Figura 38.



**Figura 38.** Proceso de configuración para acceso remoto

## B. Control

El lazo básico de control en lazo abierto correspondiente a la plataforma robótica desarrollada se presenta en la Figura 39.



**Figura 39.** Lazo de control

## C. Interfaz HMI

La interfaz HMI desarrollada se compone de una pantalla principal y cuatro pantallas derivadas. La pantalla principal puede apreciarse en la Figura 40, y permite navegar entre los diferentes modos de operación del manipulador robótico.



**Figura 40.** Interfaz HMI: pantalla principal

La interfaz “cinemática directa” permite posicionar al robot en función de su espacio articular, tal como se describe en la Figura 41. Cada una de las articulaciones se encuentran limitadas por software, de tal forma que no exista la posibilidad de solicitar una posición fuera del espacio articular definido para el movimiento de cada articulación.



**Figura 41.** Interfaz HMI: cinemática directa

La interfaz “cinemática inversa” permite solicitar al robot que alcance una posición y orientación definidas dentro del espacio de tarea, tal como se describe en la Figura 42. Si se solicita una posición que no puede ser alcanzada por el manipulador robótico, se muestra el error en pantalla y se solicita el ingreso de una ubicación diferente.



**Figura 42.** Interfaz HMI: cinemática inversa

La interfaz “generar secuencia” permite posicionar al robot en función de su espacio articular, y una vez que la posición alcanzada es satisfactoria para el usuario, permite almacenar las variables articulares en un archivo de texto y actualiza los datos en pantalla. Las funcionalidades de la pantalla puede apreciarse en la Figura 43.



**Figura 43.** Interfaz HMI: generar secuencia

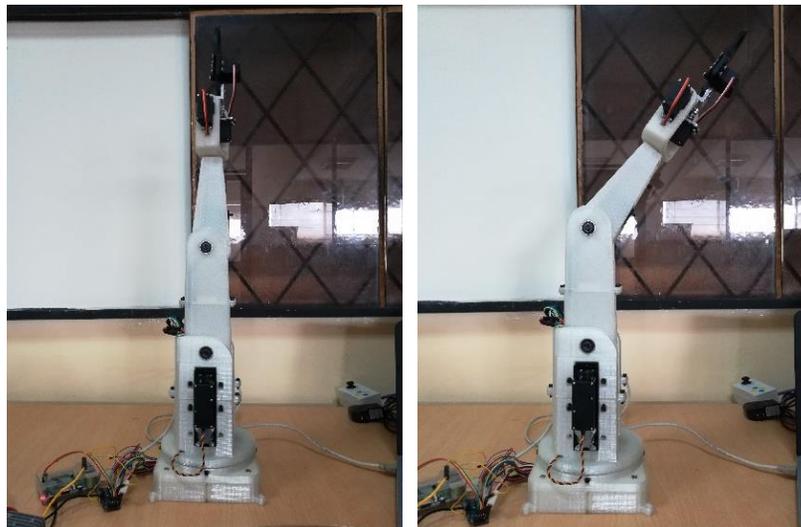
La interfaz “reproducir secuencia” permite seleccionar un archivo de texto con una secuencia previamente programada, ya sea mediante la pantalla correspondiente o de forma manual, y

presenta los datos en pantalla (Figura 44). Una vez que los datos presentados son satisfactorios para el usuario, la secuencia puede ser ejecutada y transferida en forma de comandos para ser interpretados por el robot.



**Figura 44.** Interfaz HMI: reproducir secuencia

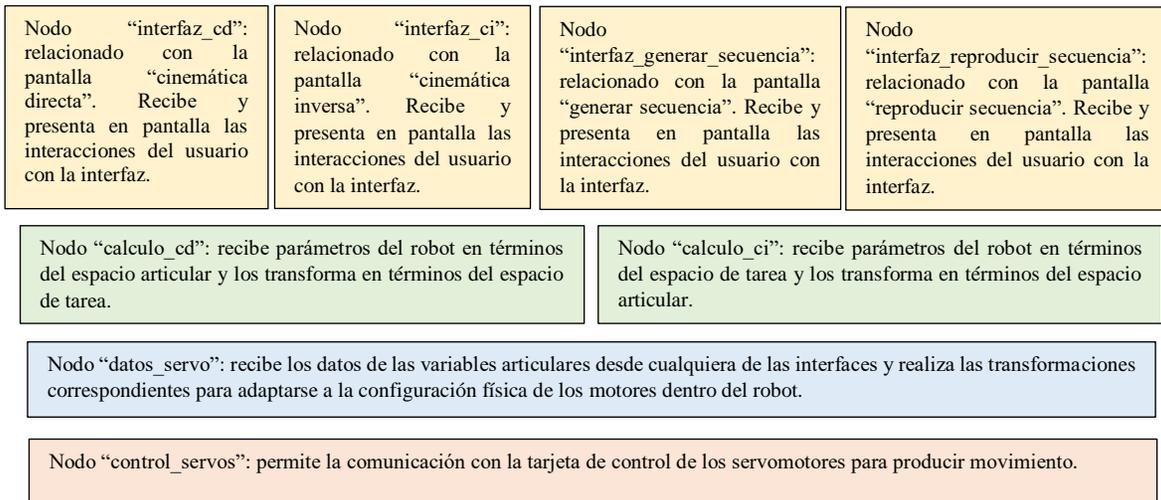
Una demostración de los movimientos generados con la ayuda de la interfaz HMI se presenta en la Figura 45.



**Figura 45.** Movimientos generados mediante la interfaz HMI

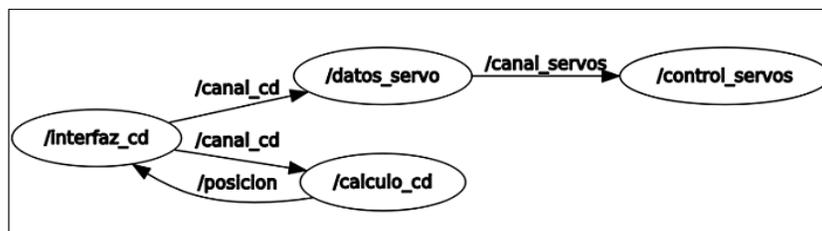
## D. Arquitectura ROS

La arquitectura de software se basó en programación jerárquica modular basada en software libre. La Figura 46 describe la jerarquía de los nodos desarrollados en ROS con su respectiva funcionalidad y nivel jerárquico:



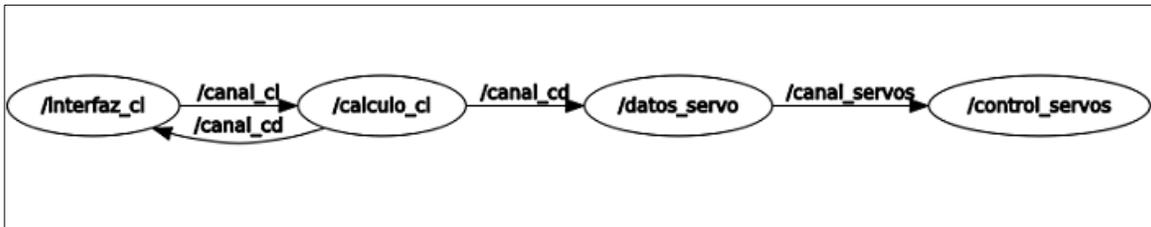
**Figura 46.** Jerarquía de nodos desarrollados en ROS

Cada uno de los nodos, mientras se encuentran en ejecución, envían y reciben información específica a través de “canales” de comunicación designados para tal efecto. Los nodos que interactúan en la pantalla “cinemática directa” se describen en la Figura 47.



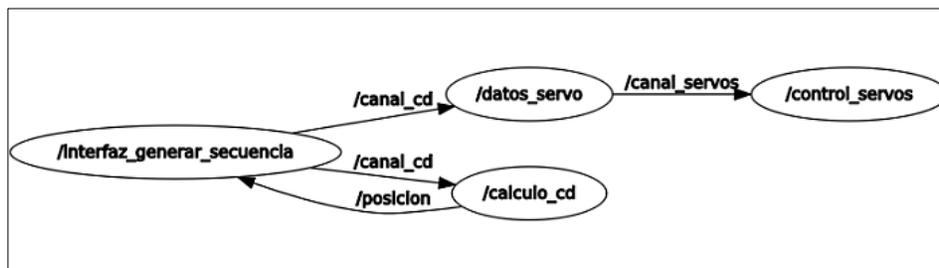
**Figura 47.** Pantalla cinemática directa. Diagrama de nodos ROS

La interacción de los nodos en la pantalla “cinemática inversa” se presenta en la Figura 48.



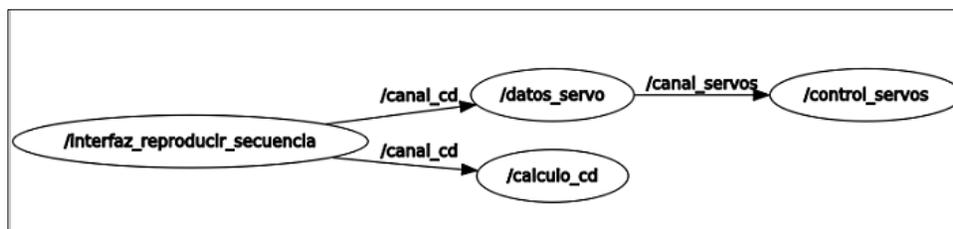
**Figura 48.** Pantalla cinemática inversa. Diagrama de nodos ROS

Los nodos requeridos para el funcionamiento adecuado de la pantalla “generar secuencia” y sus interacciones se describen en la Figura 49.



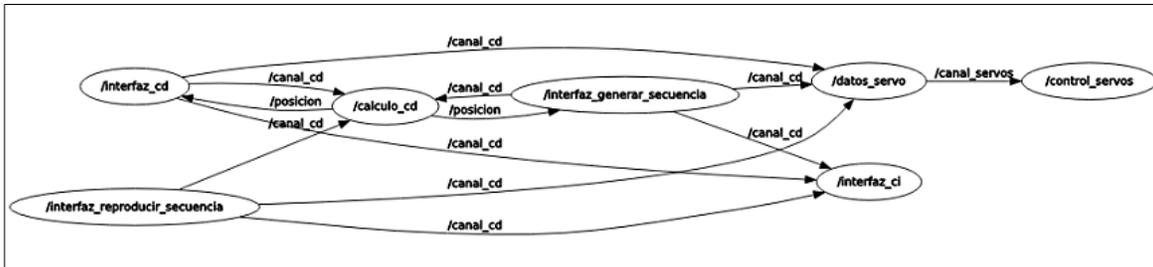
**Figura 49.** Pantalla generar secuencia. Diagrama de nodos ROS

La Figura 50 describe la interacción de los nodos dentro de la pantalla “reproducir secuencia”.



**Figura 50.** Pantalla reproducir secuencia. Diagrama de nodos ROS

Existen diversos nodos que comparten el mismo tipo de información dentro de varias pantallas de la interfaz. La interacción de todos los nodos que componen la plataforma robótica con sus respectivos canales de comunicación se presentan en la Figura 51.



*Figura 51.* Diagrama de nodos ROS. Árbol general

## CAPÍTULO IV

### PRUEBAS Y RESULTADOS

#### 4.1 Desarrollo del protocolo de pruebas

Para el desarrollo del protocolo de pruebas se consideró como punto de partida la norma ISO 9283:1998 (ISO (International Organization for Standardization), 1998), relacionada con la manipulación de robots industriales. La norma, en términos generales, define las especificaciones y protocolos de pruebas para diversas características de desempeño de sistemas robóticos. Las pruebas descritas permiten evaluar individualmente las especificaciones de manipuladores industriales; su uso puede extenderse a prueba de prototipos.

##### 4.1.1 Prueba de precisión

Para un manipulador robótico, la precisión puede evaluarse en base a dos componentes:

- Precisión de posición: La diferencia entre la posición solicitada por la pose y el baricentro (promedio) de la posición obtenida (véase Figura 52) .

$$AP_p = \sqrt{(\bar{x} - x_c)^2 + (\bar{y} - y_c)^2 + (\bar{z} - z_c)^2} \quad (49)$$

$$AP_x = (\bar{x} - x_c) \quad (50)$$

$$AP_y = (\bar{y} - y_c) \quad (51)$$

$$AP_z = (\bar{z} - z_c) \quad (52)$$

Donde:

$$\bar{x} = \frac{1}{n} \sum_{j=1}^n x_j$$

$$\bar{y} = \frac{1}{n} \sum_{j=1}^n y_j$$

$$\bar{z} = \frac{1}{n} \sum_{j=1}^n z_j$$

- Precisión de orientación: La diferencia entre la orientación solicitada por la pose y el promedio de las orientaciones obtenidas.

$$AP_a = (\bar{a} - a_c) \quad (53)$$

$$AP_b = (\bar{b} - b_c) \quad (54)$$

$$AP_c = (\bar{c} - c_c) \quad (55)$$

Donde:

$$\bar{a} = \frac{1}{n} \sum_{j=1}^n a_j$$

$$\bar{b} = \frac{1}{n} \sum_{j=1}^n b_j$$

$$\bar{c} = \frac{1}{n} \sum_{j=1}^n c_j$$

#### 4.1.2 Prueba de repetitividad

La repetitividad expresa la proximidad entre las poses obtenidas luego de n visitas repetidas a la pose objetivo desde la misma dirección. Para una pose dada, la repetitividad se expresa como:

- El valor de  $RP_l$  (véase Figura 52), que se expresa como el radio de la esfera cuyo centro es el baricentro, y se calcula como:

$$RP_l = \bar{l} + 3S_l \quad (56)$$

Donde:

$$\bar{l} = \frac{1}{n} \sum_{j=1}^n l_j$$

$$l_j = \sqrt{(x_j - \bar{x})^2 + (y_j - \bar{y})^2 + (z_j - \bar{z})^2}$$

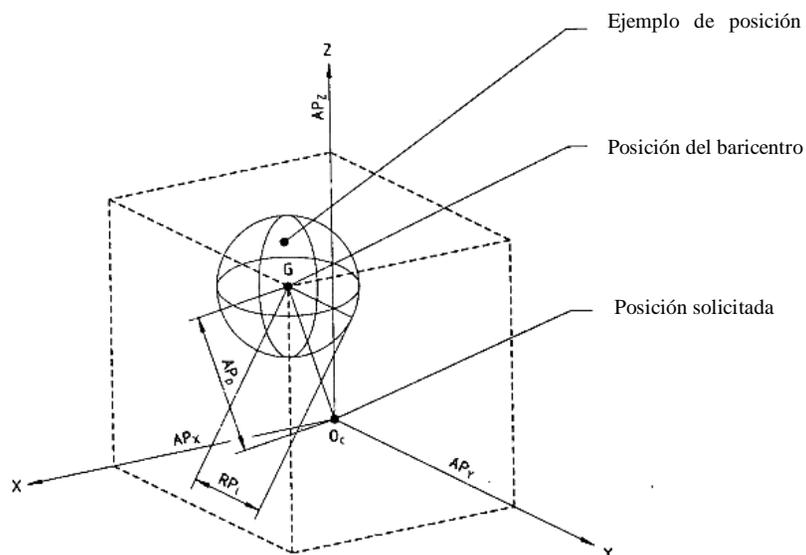
$$S_l = \sqrt{\frac{\sum_{j=1}^n (l_j - \bar{l})^2}{n - 1}}$$

- La distribución de los ángulos  $3S_a$ ,  $3S_b$ ,  $3S_c$  alrededor de los valores promedio  $\bar{a}$ ,  $\bar{b}$ ,  $\bar{c}$ , donde  $S_a$ ,  $S_b$  y  $S_c$  representan las desviaciones estándar.

$$RP_a = \pm 3S_a = \pm 3 \sqrt{\frac{\sum_{j=1}^n (a_j - \bar{a})^2}{n-1}} \quad (57)$$

$$RP_b = \pm 3S_b = \pm 3 \sqrt{\frac{\sum_{j=1}^n (b_j - \bar{b})^2}{n-1}} \quad (58)$$

$$RP_c = \pm 3S_c = \pm 3 \sqrt{\frac{\sum_{j=1}^n (c_j - \bar{c})^2}{n-1}} \quad (59)$$



**Figura 52.** Precisión y repetitividad de posición

FUENTE: (Bureau of Indian Standards, 2005)

## 4.2 Resultados

Conforme al protocolo de pruebas descrito en el apartado anterior, se realizaron pruebas con el objetivo de cuantificar las características de precisión y repetitividad para la plataforma robótica

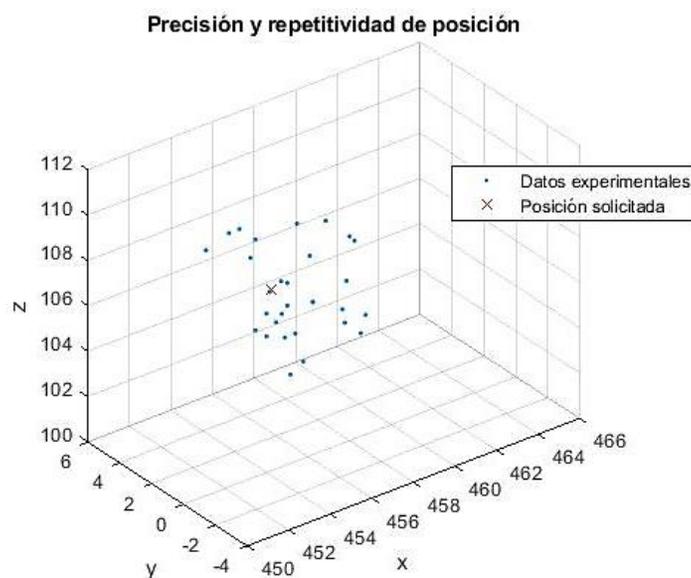
desarrollada, obteniendo los resultados presentados en la Tabla 27. La Figura 53 muestra una representación gráfica de los resultados de posición obtenidos.

**Tabla 27**

*Datos de posición y orientación*

N°	x [mm]	y [mm]	z [mm]	A [°]	B [°]	C [°]
1	450	1	110	179	-38	0
2	453	-2.5	111	178	-30	1
3	<b>454.5</b>	<b>5</b>	<b>109</b>	<b>178</b>	<b>-33</b>	<b>4</b>
4	456	2	105	178	-33	2
5	454	0	107	177	-32	3
6	455	-3	105	177	-33	6
7	454	0	106	176	-33	3
8	454	-1.5	105	178	-31	3
9	457	1	106	177	-30	4
10	457	-2	102	178	-33	2
11	459	-2	102	178	-34	-2
12	459	1.5	106	177	-29	-6
13	458	0	110	180	-30	1
14	454.5	-0.5	106	180	-30	0
15	459	1	108	179	-31	0
16	452.5	-0.5	106	179	-28	1
17	453	-2.5	109	179	-30	-2
18	456.5	-1.5	107	179	-29	1
19	455	0	108	183	-31	3
20	455	0	107	182	-31	4
21	458	1.5	110	184	-31	3
22	455	3	110	182	-30	2
23	457	1	106	183	-29	2
24	455	2	112	184	-30	3
25	459	4	109	182	-32	2
26	458	-2.5	107	181	-30	2
27	458	-1	106	181	-30	3
28	455	2	105	181	-32	2
29	455.5	1	103	180	-29	0
30	459	4	103	181	-32	-1

La interpretación de los datos obtenidos permite obtener los resultados de precisión y repetitividad de posición y orientación, que se describen en la Tabla 28 y Tabla 29, respectivamente.



**Figura 53.** Resultados de precisión y repetitividad de posición

**Tabla 28**

*Resultados de precisión*

Precisión	Valor
Posición	2.03 [mm]
A	-0.30 [°]
B	-1.13 [°]
C	1.53 [°]

**Tabla 29**

*Resultados de repetitividad*

Repetitividad	Valor
Posición	$\pm 4.89$ [mm]
A	$\pm 6.71$ [°]
B	$\pm 4.72$ [°]
C	$\pm 6.98$ [°]

### 4.3 Análisis financiero

La Tabla 30 enumera los costos de la materia prima utilizada para la fabricación del prototipo desarrollado. El análisis no toma en consideración costos tales como el tiempo de impresión 3D y la mano de obra debido a que son valores variables de acuerdo al entorno en el que se pueda replicar la fabricación de la plataforma robótica propuesta.

**Tabla 30**  
*Análisis financiero*

Producto/Servicio	Cantidad	Valor unitario	Valor total
Servomotor con caja reductora TM-805MG-400	2	157.48	314.96
Gripper	1	8.49	8.49
Acoples para servomotor	-	-	73.19
Rodamientos 8 mm	4	1.30	5.20
Rodamiento ¼"	1	3.50	3.50
Rodamientos 4 mm	4	3.18	12.72
Banda dentada [m]	3	2.50	7.50
Elementos de sujeción	-	-	5.50
Servomotor analógico 15 kg/cm	4	14	56
Servomotor digital 20 kg*cm	2	21.99	43.98
Filamento PLA	2	15	30
Acrílico 3 mm	1	15	15
<b>TOTAL:</b>			576.04

## CAPÍTULO V

### CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

#### 5.1 Conclusiones

- Se diseñó e implementó un robot antropomórfico de seis grados de libertad Open Source (OSH-OSS) con una precisión de 2.03 [mm] y repetitividad de  $\pm 4.89$  [mm], con una carga útil de 0.15 [kg] y una relación de carga útil de 5.21%.
- La fabricación del prototipo se realizó principalmente utilizando el proceso de impresión 3D con PLA en una impresora Prusa i3 con un volumen de impresión de 225x225x160 [mm], con un tiempo de fabricación aproximado de 93 horas.
- El cálculo de la cinemática directa se realizó utilizando la definición de parámetros de Denavit-Hartenberg, mientras que la cinemática inversa se obtuvo mediante la aplicación del concepto de desacople cinemático.
- El software de control se desarrolló sobre ROS Kinetic, basado en el concepto de nodos intercambiables.
- Se desarrolló una interfaz HMI utilizando el entorno de desarrollo Glade 3.18.3 en Ubuntu Mate 1.16.2.

#### 5.2 Recomendaciones

- A pesar de que se trata de un robot pequeño, se debe tener en consideración las mismas medidas de seguridad que se aplican para robots industriales, debido a que el tamaño reducido de la

plataforma robótica en ningún caso significa que se eliminen los riesgos relacionados al manejo de manipuladores industriales.

- Considerar el uso de ABS como material de fabricación de poleas, engranes y otros elementos sometidos a esfuerzos, para incrementar la vida útil de los componentes.
- Revisar los mecanismos de transmisión de movimiento con el objetivo de mejorar los parámetros de precisión y repetitividad.

### **5.3 Trabajo futuro**

Se propone enfocar esfuerzos en incrementar las funcionalidades del robot, con enfoque a la programación de secuencias y generación de trayectorias. La plataforma robótica desarrollada posee el potencial para aplicaciones en diversos campos tales como:

- Robótica colaborativa.
- Interacción humano-robot.

## BIBLIOGRAFÍA

Adafruit. (2015). *Adafruit 16-Channel 12-bit PWM/Servo Driver - I2C interface - PCA9685*.

Recuperado el Julio de 2018, de <https://www.adafruit.com/product/815>

Association of German Engineers. (2004). *VDI 2206: Design methodology for mechatronic systems*. Berlin.

Bureau of Indian Standards. (2005). *Manipulating industrial iobots - Performance criteria and related test methods*. Nueva Delhi.

Designatronics Inc. (2013). *Drive your System with Timing Belts*. Obtenido de <http://sdp-si.com/product-announcements/Timing%20Belts%20White%20Paper.pdf>

Designatronics Inc. (2018). *Belts and Pulleys - Technical Data*. Recuperado el Septiembre de 2018, de Technical Resources: <http://sdp-si.com/resources/>

Designatronics Inc. (2018). *Guidelines For GT2 Flange Design*. Recuperado el Octubre de 2018, de Technical section: [http://www.sdp-si.com/D265/HTML/D265T063\\_6.html](http://www.sdp-si.com/D265/HTML/D265T063_6.html)

Free Software Foundation Europe. (Marzo de 2018). *Free Software's Four Freedoms*. Recuperado el 07 de Mayo de 2018, de <https://fsfe.org/freesoftware/basics/4freedoms.en.html>

Geeetech. (2016). *Prusa I3 X*. Recuperado el Octubre de 2018, de Geeetech Wiki: [http://www.geeetech.com/wiki/index.php/Prusa\\_I3\\_X](http://www.geeetech.com/wiki/index.php/Prusa_I3_X)

GlavSoft LLC. (2018). Recuperado el Enero de 2019, de TightVNC Software: <https://www.tightvnc.com/>

Government of Prince Edward Island. (s.f.). *Open Source Software*. Recuperado el 07 de Mayo de 2018, de [http://www.gov.pe.ca/photos/original/IPEI\\_ebiz\\_oss.pdf](http://www.gov.pe.ca/photos/original/IPEI_ebiz_oss.pdf)

Hägele, M., Nilsson, K., & Pires, N. (2008). Industrial Robotics. En B. Siciliano, & K. Oussama, *Springer Handbook of Robotics* (págs. 963-986). Berlin: Springer.

ISO (International Organization for Standardization). (01 de Abril de 1998). Manipulating Industrial robots-Performance criteria and related test methods. *Norma ISO 9283:1998*(Segunda Edición).

Jaramillo Botero, A. (s.f.). *Cinemática de Manipuladores Robóticos*. Obtenido de [http://www.wag.caltech.edu/home/ajaramil/libro\\_robotica/cinematica.pdf](http://www.wag.caltech.edu/home/ajaramil/libro_robotica/cinematica.pdf)

Joseph, L. (2015). *Learning Robotics Using Python*. Birmingham: Packt Publishing.

LewanSoul. (2017). *LD-220MG Servo*. Recuperado el Noviembre de 2018, de <http://www.lewansoul.com/product/detail-20.html>

LewanSoul. (2017). *LD-27MG Servo*. Recuperado el Noviembre de 2018, de <http://www.lewansoul.com/product/detail-19.html>

Lynch, K., & Park, F. (2017). *Modern Robotics*. Cambridge University Press.

Mital, A., Desai, A., Subramanian, A., & Mital, A. (2008). *Product Development*. Elsevier.

Niku, S. B. (2001). *Introduction to Robotics: Analysis, Systems, Applications*. Prentice Hall.

Niryo. (2018). *Niryo one*. Recuperado el 06 de Marzo de 2018, de Niryo: Robotics for Everyone: <https://niryo.com/>

Oberloier, S., & Pearce, J. (30 de Diciembre de 2017). General Design Procedure for Free and Open-Source Hardware for Scientific Equipment. Recuperado el 25 de Abril de 2018, de <http://www.mdpi.com/2411-9660/2/1/2/pdf>

Open Source Hardware Association. (2018). *Definition (English)*. Recuperado el 25 de Abril de 2018, de Open Source Hardware Association Web site: <https://www.oshwa.org/definition/>

Open Source Initiative. (22 de Marzo de 2007). *The Open Source Definition*. Recuperado el 24 de Abril de 2018, de Open Source Initiative: <https://opensource.org/docs/osd>

Open Source Robotics Foundation. (s.f.). Recuperado el 06 de Marzo de 2018, de ROS.org: <http://wiki.ros.org/es>

Open Source Robotics Foundation. (s.f.). *Main Page*. Recuperado el 06 de Marzo de 2018, de Robot Operating System (ROS): <http://www.ros.org>

Quigley, M., Gerkey, B., & Smart, W. D. (2015). *Programming Robotswith ROS* (Primera ed.). O'Reilly.

Raspberry Pi Foundation. (2018). *Raspberry Pi 3 Model B*. Recuperado el Noviembre de 2018, de <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

RealVNC Limited. (2018). Obtenido de RealVNC: <https://www.realvnc.com/en/connect/download/viewer/>

Roboteurs. (2018). *RBX1 (Remix)*. Recuperado el 13 de Marzo de 2018, de <https://roboteurs.com/products/rbx1-remix-3d-printed-6-axis-robot-arm-kit>

ROS components. (s.f.). *Manipulator H*. Recuperado el 13 de Marzo de 2018, de ROS components:

<https://www.roscomponents.com/es/brazos-roboticos/120-manipulator-h.html>

Rubow, E. (20 de Noviembre de 2008). Open Source Hardware. Recuperado el 24 de Abril de

2018, de [https://cseweb.ucsd.edu/classes/fa08/cse237a/topicresearch/erubow\\_tr\\_report.pdf](https://cseweb.ucsd.edu/classes/fa08/cse237a/topicresearch/erubow_tr_report.pdf)

ServoCity. (2018). *TM-805MG-400 Servo Gearbox*. Recuperado el Noviembre de 2018, de Servo

Gearboxes: <https://www.servocity.com/tm-805mg-400-servo-gearbox#197=4&198=8>

Spong, M., Hutchinson, S., & Vidyasagar, M. (2005). *Robot Modeling and Control* (Primera ed.).

John Wiley & Sons Inc.

Torq Pro & Tower Pro. (2014). *MG946R*. Recuperado el Noviembre de 2018, de Servos & Parts:

<http://www.towerpro.com.tw/product/mg946r/>

Tucson Amateur Packet Radio Corp. (25 de Mayo de 2007). *The TAPR Open Hardware License*.

Recuperado el 25 de Abril de 2018, de TAPR Web site:

[http://www.tapr.org/TAPR\\_Open\\_Hardware\\_License\\_v1.0.txt](http://www.tapr.org/TAPR_Open_Hardware_License_v1.0.txt)

UK Cabinet Office. (Abril de 2012). *All about Open Source: An Introduction to Open Source*

*Software for Government IT*. Recuperado el 07 de Mayo de 2018, de

[https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment\\_data/file/78959/All\\_About\\_Open\\_Source\\_v2\\_0.pdf](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/78959/All_About_Open_Source_v2_0.pdf)

Ulrich, Karl, & Eppinger, S. (2013). *Diseño y desarrollo de productos* (Quinta ed.). México D.F.:

McGRAW-HILL.

Ultimaker B.V. (2018). *Ultimaker Cura software*. Recuperado el Noviembre de 2018, de <https://ultimaker.com/en/products/ultimaker-cura-software>

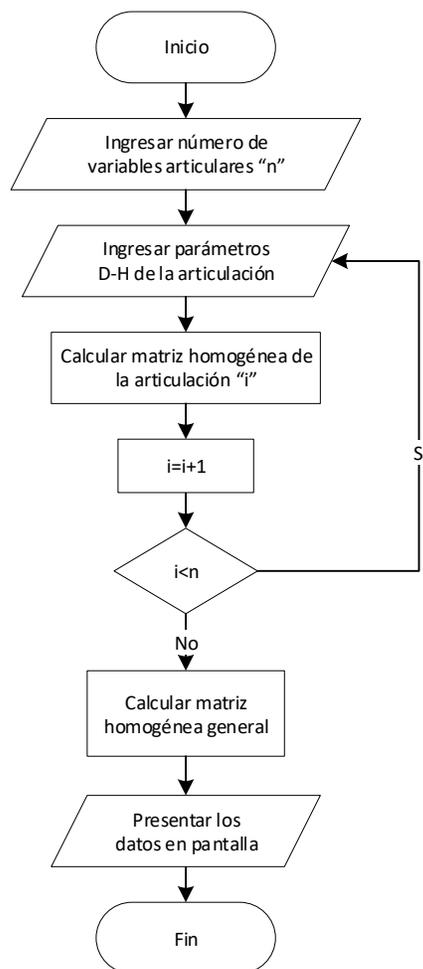
Warnecke, H., Schraft, R., Hägele, M., Barth, O., & Schmierer, G. (1999). Manipulator Design. En S. Y. Nof, *Handbook of Industrial Robotics* (Segunda ed., págs. 43-78). Stuttgart: John Wiley & Sons, Inc.

## ANEXOS

### A. ALGORITMOS DE PROGRAMACIÓN

#### a. Obtención de matriz homogénea a partir de parámetros D-H

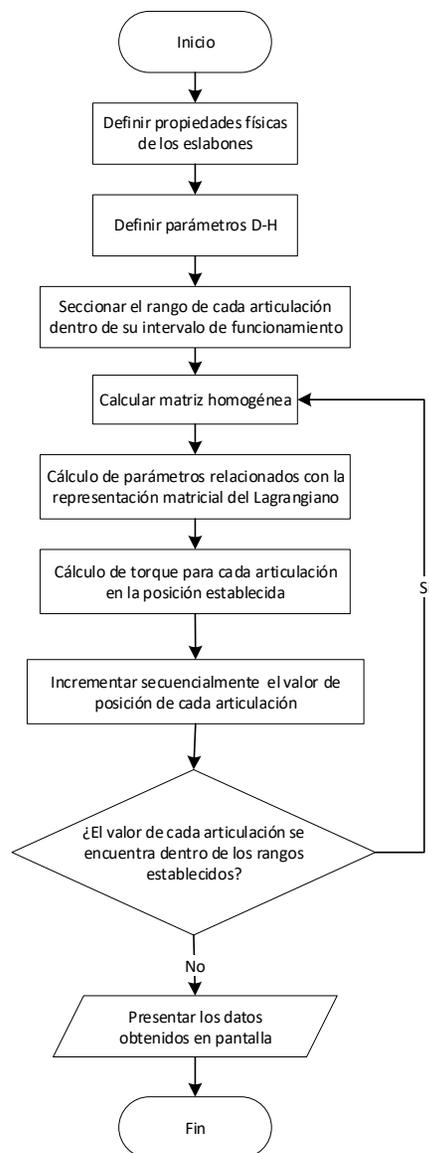
El programa desarrollado en Matlab permite obtener la matriz homogénea de un robot de “n” grados de libertad ingresando por teclado los valores de los parámetros de Denavit-Hartenberg propios de la configuración de sus variables articulares. El diagrama de flujo que resume el algoritmo desarrollado se describe en la Figura 54.



**Figura 54.** Obtención de matriz homogénea. Diagrama de flujo

## b. Modelamiento de torque máximo en cada articulación

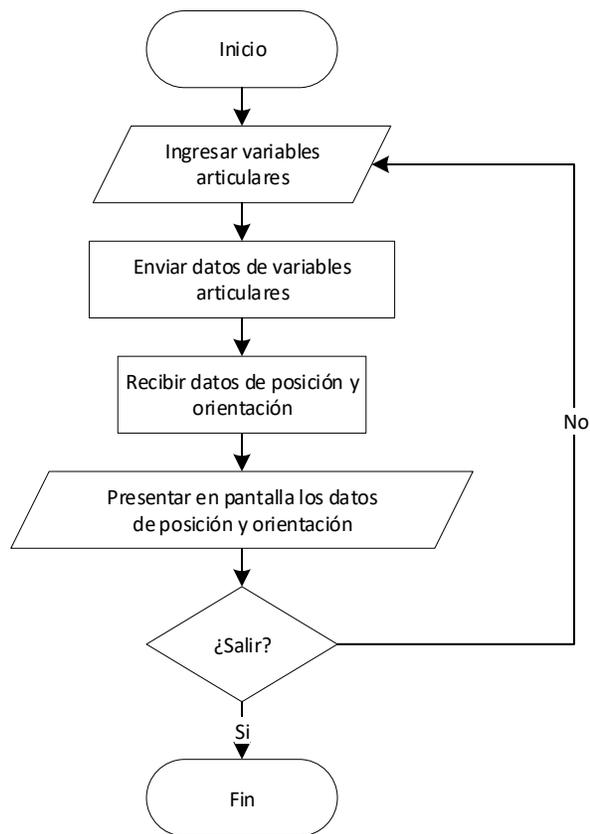
El programa desarrollado en Matlab representa un método iterativo que permite graficar el comportamiento dinámico de un robot con el objetivo de calcular el torque máximo presente en cada articulación. El diagrama de flujo que resume el algoritmo desarrollado se describe en la Figura 54.



**Figura 55.** Modelamiento de torque máximo. Diagrama de flujo

### c. Nodo “interfaz\_cd”

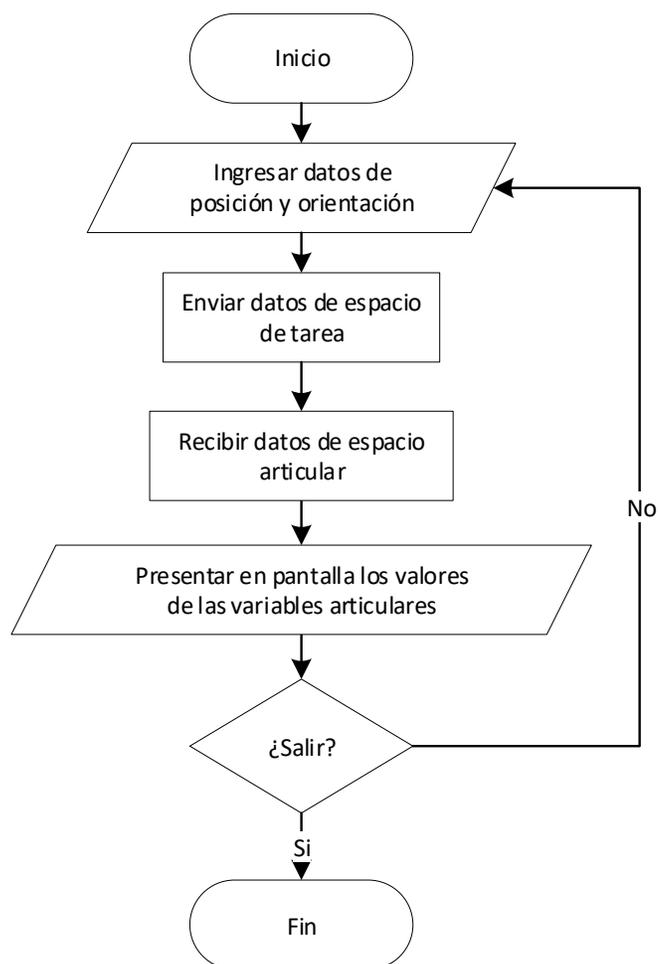
El diagrama de flujo presentado en la Figura 56 describe el funcionamiento de un nodo de ROS que permite la interacción del usuario con la interfaz HMI que describe la cinemática directa del robot.



**Figura 56.** Nodo “interfaz\_cd”. Diagrama de flujo

### d. Nodo “interfaz\_ci”

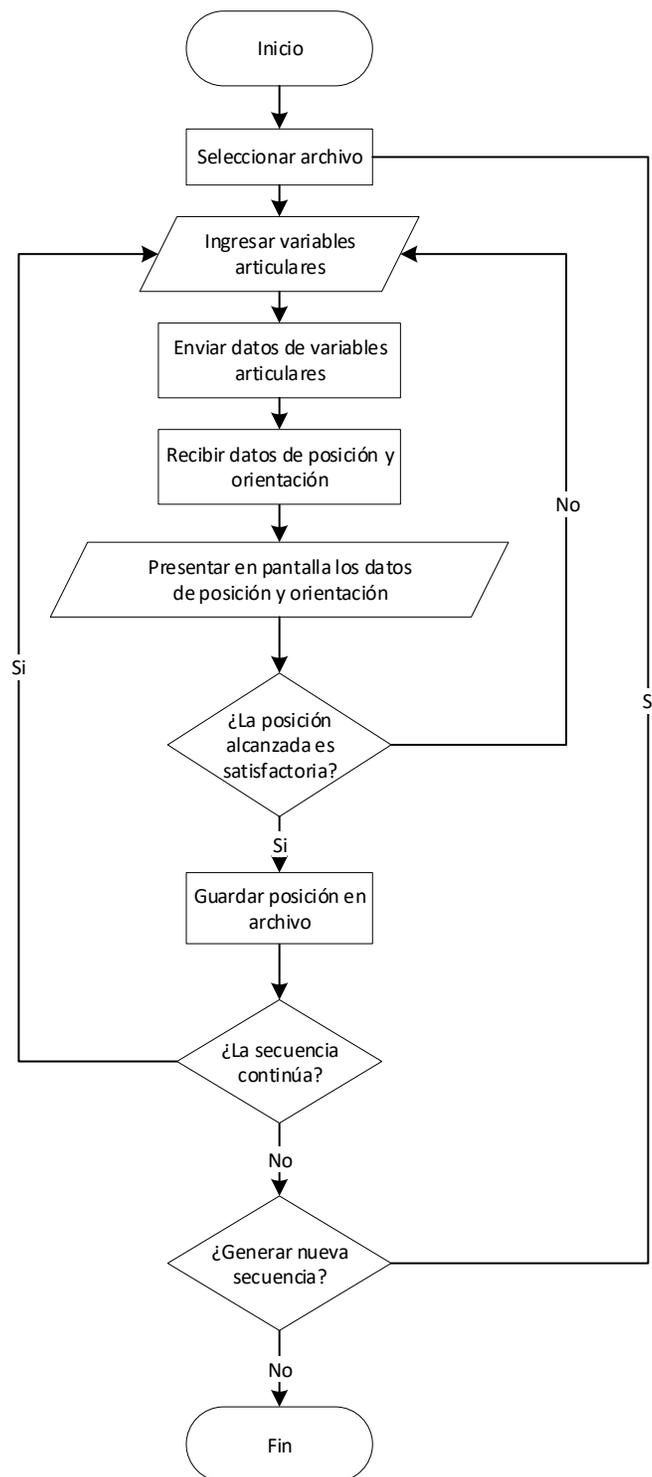
El diagrama de flujo presentado en la Figura 57 describe el funcionamiento de un nodo de ROS que permite la interacción del usuario con la interfaz HMI que describe la cinemática inversa del robot.



**Figura 57.** Nodo “interfaz\_ci”. Diagrama de flujo

#### **e. Nodo “interfaz\_generar\_secuencia”**

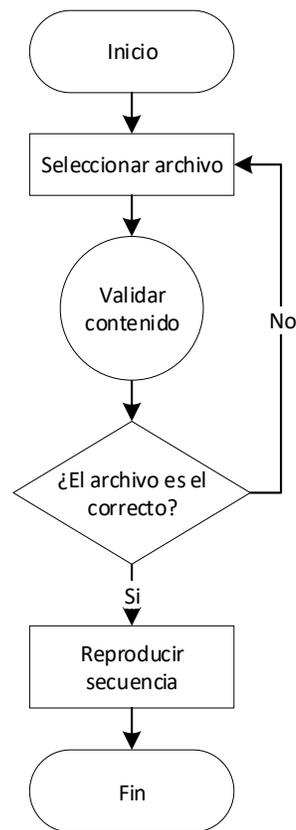
El diagrama de flujo presentado en la Figura 58 describe el funcionamiento de un nodo de ROS que permite la interacción del usuario con la interfaz HMI para programar secuencias a partir de la cinemática directa del robot.



**Figura 58.** Nodo “interfaz\_generar\_secuencia”. Diagrama de flujo

#### f. Nodo “interfaz\_reproducir\_secuencia”

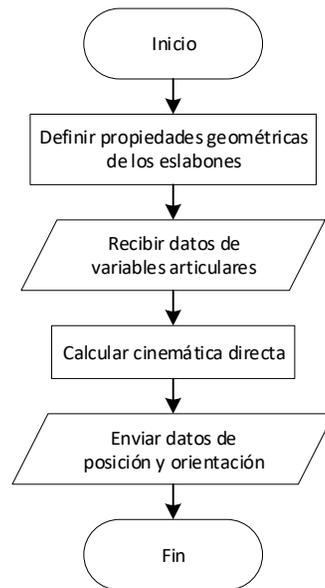
El diagrama de flujo presentado en la Figura 59 describe el funcionamiento de un nodo de ROS que permite la interacción del usuario con la interfaz HMI para reproducir secuencias previamente almacenadas en archivos de texto.



**Figura 59.** Nodo “interfaz\_reproducir\_secuencia”. Diagrama de flujo

#### g. Nodo “calculo\_cd”

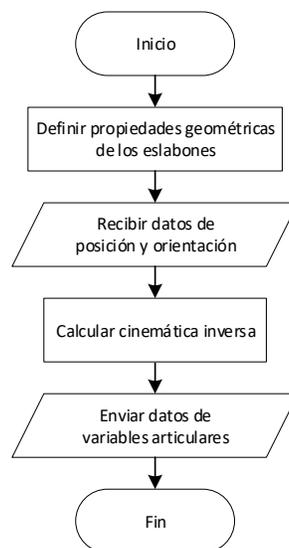
El diagrama de flujo presentado en la Figura 60 describe el funcionamiento de un nodo de ROS que recibe parámetros del robot en términos del espacio articular y los transforma en términos del espacio de tarea.



**Figura 60.** Nodo “calculo\_cd”. Diagrama de flujo

#### **h. Nodo “calculo\_ci”**

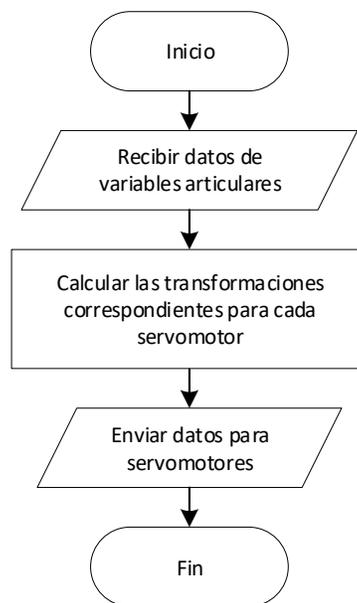
El diagrama de flujo presentado en la Figura 61 describe el funcionamiento de un nodo de ROS que recibe parámetros del robot en términos del espacio de tarea y los transforma en términos del espacio articular.



**Figura 61.** Nodo “calculo\_ci”. Diagrama de flujo

**i. Nodo “datos\_servo”**

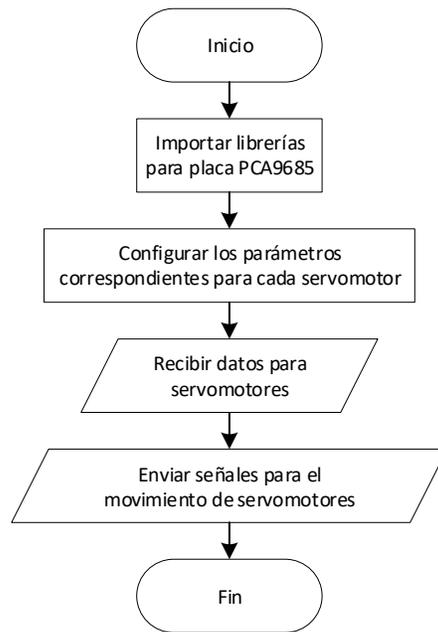
El diagrama de flujo presentado en la Figura 62 describe el funcionamiento de un nodo de ROS que recibe los datos de las variables articulares desde cualquiera de las interfaces y realiza las transformaciones correspondientes para adaptarse a la configuración física de los motores dentro del robot.



**Figura 62.** Nodo “datos\_servo”. Diagrama de flujo

**j. Nodo “control\_servos”**

El diagrama de flujo presentado en la Figura 63 describe el funcionamiento de un nodo de ROS que permite la comunicación con la tarjeta de control de los servomotores para producir movimiento.



**Figura 63.** Nodo “control\_servos”. Diagrama de flujo

**B. PLANOS MECÁNICOS**