



**Diseño y construcción de una celda robótica inteligente basada en un planificador de tipo**

**“STRIPS”**

García Mora, Gelin Michelle y Mayancela Vaca, David Ricardo

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería en Mecatrónica

Trabajo de titulación, previo a la obtención del título de Ingeniero/a en Mecatrónica

Mgs. Loza Matovelle, David César

16 de agosto del 2021

## Document Information

---

Analyzed document	Escrito Cap1-5 - García - Mayancela.pdf (D111350650)
Submitted	8/16/2021 5:29:00 PM
Submitted by	
Submitter email	dcloza@espe.edu.ec
Similarity	3%
Analysis address	dcloza.espe@analysis.arkund.com

## Sources included in the report

---

<b>SA</b>	<p>Universidad de las Fuerzas Armadas ESPE / TESIS_CORDOVA_QUIMBIAMBA.pdf  Document TESIS_CORDOVA_QUIMBIAMBA.pdf (D40883264)  Submitted by: dcloza@espe.edu.ec  Receiver: dcloza.espe@analysis.arkund.com</p>	  <b>8</b>
<b>W</b>	<p>URL: <a href="https://docplayer.es/76095915-Instituto-politecnico-nacional.html">https://docplayer.es/76095915-Instituto-politecnico-nacional.html</a>  Fetched: 7/19/2020 5:21:20 AM</p>	  <b>1</b>
<b>SA</b>	<p>Universidad de las Fuerzas Armadas ESPE /  RevisiónTesis_PoorDog_EdwinJimenez_ChirstianMoreno.pdf  Document RevisiónTesis_PoorDog_EdwinJimenez_ChirstianMoreno.pdf (D54346750)  Submitted by: eajimenez7@espe.edu.ec  Receiver: wgaguilar.espe@analysis.arkund.com</p>	  <b>1</b>
<b>SA</b>	<p>Universidad de las Fuerzas Armadas ESPE / Trabajo de Titulación_Aguirre-De la Cruz.pdf  Document Trabajo de Titulación_Aguirre-De la Cruz.pdf (D57817164)  Submitted by: bhcortez@espe.edu.ec  Receiver: bhcortez.espe@analysis.arkund.com</p>	  <b>3</b>



Firmado electrónicamente por:  
**DAVID CESAR  
LOZA  
MATOVELLE**

Ing. David César Loza Matovelle  
Director de la Carrera de Ingeniería Mecatrónica



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA  
CARRERA DE INGENIERÍA EN MECATRÓNICA**

**Certificación**

Certifico que el trabajo de titulación, “**Diseño y construcción de una celda robótica inteligente basada en un planificador de tipo “STRIPS”**” fue realizado por los señores **García Mora, Gelin Michelle y Mayancela Vaca, David Ricardo**, el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 16 de agosto del 2021

Firma:



**Loza Matovelle, David César**

C. C. 1708661549



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA**  
**CARRERA DE INGENIERÍA EN MECATRÓNICA**

**Responsabilidad de autoría**

Nosotros, **García Mora, Gelin Michelle y Mayancela Vaca, David Ricardo**, con cédulas de ciudadanía n°1719791285 y n°1725349888, declaramos que el contenido, ideas y criterios del trabajo de titulación: **“Diseño y construcción de una celda robótica inteligente basada en un planificador de tipo “STRIPS””** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

**Sangolquí, 16 de agosto del 2021**

Firmas:

  
.....

**García Mora, Gelin Michelle**

C.C.: 1719791285

  
.....

**Mayancela Vaca, David Ricardo**

C.C.: 1725349888



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA  
CARRERA DE INGENIERÍA MECATRÓNICA**

**Autorización de publicación**

Nosotros, **García Mora, Gelin Michelle y Mayancela Vaca, David Ricardo**, con cédulas de ciudadanía n°1719791285 y n°1725349888, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“Diseño y construcción de una celda robótica inteligente basada en un planificador de tipo “STRIPS”**” en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

**Sangolquí, 16 de agosto del 2021**

Firmas:

  
.....

**García Mora, Gelin Michelle**

C.C.: 1719791285

  
.....

**Mayancela Vaca, David Ricardo**

C.C.: 1725349888

## **Dedicatoria 1**

Quiero dedicar este trabajo a mis padres, que siempre me apoyan en todo y a aquellas personas que en algún momento de su vida se sintieron perdidas y sin motivos para continuar, al final del camino siempre hay una luz que nos empuja a seguir adelante y con esto a ser cada vez más fuertes, la recompensa es la resiliencia.

*Gelin Michelle García Mora*

**Dedicatoria 2**

A mis padres...

*David R. Mayancela Vaca*

## **Agradecimiento 1**

Agradezco a Dios por darme la oportunidad de seguir adelante siempre, a pesar de las adversidades, a mis padres que me han dado todo lo necesario para llegar a este punto y ser quien soy, a mi hermana que siempre me apoya en todo sin importar el resultado y a mi prometido que estuvo a mi lado desde el principio apoyándome.

*Gelin Michelle García Mora*

## Agradecimiento 2

Como no estar completamente agradecido con mis padres y mi hermano a quienes debo enteramente el haber llegado hasta este punto. Agradezco sobre todo a mi familia y a todas las personas que de alguna forma aportaron y me acompañaron en algún momento a lo largo de todo este camino que, aunque sigan o no, sin duda cada uno ha dejado algo bueno en mí.

Gracias.

*David R. Mayancela Vaca*

## Índice de Contenidos

Certificación .....	3
Responsabilidad de autoría .....	4
Autorización de publicación .....	5
Dedicatoria 1.....	6
Dedicatoria 2.....	7
Agradecimiento 1.....	8
Agradecimiento 2.....	9
Resumen .....	18
Abstract.....	19
Capítulo 1.....	20
Descripción general del proyecto .....	20
Introducción.....	20
Planteamiento del problema .....	21
Justificación e Importancia .....	22
Objetivos .....	23
Objetivo general .....	23
Objetivos específicos .....	23
Alcance.....	24
Estructura del documento .....	24
Capítulo 2.....	25
Estado del arte .....	25
Manipuladores Robóticos.....	25
Tipos de manipuladores Robóticos en la industria.....	26
Manipulador de configuración PRR.....	28
Inteligencia Artificial .....	28

	11
Planificación Automática .....	29
Tipos de planificadores .....	31
Planificación Clásica .....	32
Stanford Research Institute Problem Solver (STRIPS) .....	37
Visión Artificial en procesos Industriales .....	41
Espacios de Colores .....	41
Modelo de color HSV .....	42
Segmentación .....	42
Capítulo 3 .....	43
Diseño mecatrónico .....	43
Matriz de Calidad (QFD).....	43
Definición de necesidades .....	43
Definición de Requerimientos .....	44
Despliegue de las funciones de calidad (QFD).....	45
Factibilidad y Planeación .....	46
Módulo 1. Manipulador Robótico PRR .....	47
Subsistema 1. Cinemática y Dinámica .....	47
Subsistema 2. Estructural .....	56
Módulo 2. Espacio de trabajo .....	65
Subsistema 1. Estructura de la celda .....	66
Subsistema 2. Elementos didácticos.....	69
Módulo 3. Potencia y Control .....	70
Subsistema 1. Potencia del Robot .....	70
Subsistema 2. Control del Robot .....	76
Módulo 4. Interacción.....	78
Subsistema 1. Visión Artificial del Robot .....	78

	12
Subsistema 2. Planificador .....	80
Subsistema 3. Interfaz Gráfica .....	82
Resumen del capítulo .....	83
Capítulo 4.....	87
Construcción, implementación y evaluación.....	87
Módulo 1. Sistema Mecánico .....	87
Construcción y ensamblaje del manipulador .....	87
Construcción y ensamblaje de la celda de trabajo .....	92
Módulo 2. Sistema Eléctrico .....	94
Implementación componentes electrónicos .....	94
Consumo energético de los motores.....	95
Módulo 3. Programación de la Celda .....	95
Sistema Visión artificial.....	95
Integración del planificador STRIPS .....	97
Programación de la Celda Robótica.....	98
Programación HMI.....	100
Módulo 4. Pruebas de Funcionamiento .....	103
Prueba de Precisión Manipulador .....	103
Prueba de Repetibilidad del Manipulador.....	104
Prueba de repetibilidad de la Celda de Trabajo .....	105
Capítulo 5.....	108
Conclusiones, recomendaciones y trabajos futuros.....	108
Conclusiones .....	108
Recomendaciones.....	109
Trabajos Futuros .....	110
Bibliografía.....	111

Anexos .....	115
--------------	-----

## Índice de Tablas

<b>Tabla 1</b> Necesidades del usuario. ....	44
<b>Tabla 2</b> Requerimientos y metas operativas .....	44
<b>Tabla 3</b> Matriz QFD del Proyecto.....	45
<b>Tabla 4</b> Resultados de la Matriz QFD.....	46
<b>Tabla 5</b> Módulos y subsistemas del proyecto .....	47
<b>Tabla 6</b> Parámetros de D-H.....	48
<b>Tabla 7</b> Características de los materiales para la manufactura de la estructura .....	57
<b>Tabla 8</b> <i>Ventajas y desventajas de cada alternativa para el sistema de transmisión.</i> .....	58
<b>Tabla 9</b> <i>Ponderaciones de cada criterio para el sistema de transmisión.</i> .....	59
<b>Tabla 10</b> <i>Ponderaciones de cada alternativa respecto a cada criterio para el sistema de transmisión.</i> .....	59
<b>Tabla 11</b> <i>Selección de alternativa para el sistema de transmisión.</i> .....	60
<b>Tabla 12</b> <i>Ventajas y desventajas de cada alternativa para el efector final.</i> .....	61
<b>Tabla 13</b> <i>Ponderaciones de cada criterio para el efector final.</i> .....	62
<b>Tabla 14</b> <i>Ponderaciones de cada alternativa respecto a cada criterio para el efector final.</i> ...	62
<b>Tabla 15</b> <i>Selección de alternativa para el efector final.</i> .....	63
<b>Tabla 16</b> <i>Características del electroimán.</i> .....	63
<b>Tabla 17</b> Eslabones diseñados .....	65
<b>Tabla 18</b> <i>Ventajas y desventajas de cada alternativa para la celda de trabajo</i> .....	67
<b>Tabla 19</b> <i>Ponderaciones de cada criterio para el material de la celda.</i> .....	68
<b>Tabla 20</b> <i>Ponderaciones de cada alternativa respecto a cada criterio para el material de la celda.</i> .....	68
<b>Tabla 21</b> <i>Selección de alternativa para el material de la celda.</i> .....	69
<b>Tabla 22</b> <i>Características de los motores seleccionados.</i> .....	72
<b>Tabla 23</b> <i>Características del driver A4988 para los motores a pasos.</i> .....	72
<b>Tabla 24</b> <i>Consumo energético de cada elemento del sistema.</i> .....	73
<b>Tabla 25</b> <i>Ventajas y desventajas de cada alternativa para la fuente de alimentación.</i> .....	73
<b>Tabla 26</b> <i>Ponderaciones de cada criterio para la fuente de alimentación</i> .....	74

<b>Tabla 27</b> <i>Ponderaciones de cada alternativa respecto a cada criterio para la fuente de alimentación.</i> .....	74
<b>Tabla 28</b> <i>Selección de alternativa para la fuente de alimentación.</i> .....	75
<b>Tabla 29</b> <i>Características de la fuente de alimentación.</i> .....	75
<b>Tabla 30</b> <i>Características de Arduino vs Raspberry pi.</i> .....	76
<b>Tabla 31</b> <i>Ponderaciones de cada criterio para la Raspberry pi4.</i> .....	77
<b>Tabla 32</b> <i>Ponderaciones de cada alternativa respecto a cada criterio para la Raspberry pi4.</i> 77	
<b>Tabla 33</b> <i>Selección de alternativa para la Raspberry pi4.</i> .....	77
<b>Tabla 34</b> <i>Características de la Raspberry Pi4</i> .....	78
<b>Tabla 35</b> <i>Características de la cámara para la Raspberry Pi4</i> .....	80
<b>Tabla 36</b> <i>Características de la pantalla para la Raspberry Pi4</i> .....	83
<b>Tabla 37</b> <i>Selección final para los subsistemas del manipulador.</i> .....	83
<b>Tabla 38</b> <i>Especificaciones técnicas de la impresora.</i> .....	87
<b>Tabla 39</b> <i>Parámetros de impresión.</i> .....	87
<b>Tabla 40</b> <i>Prueba de precisión de posicionamiento del robot sin planificador.</i> .....	104
<b>Tabla 41</b> <i>Prueba de repetibilidad de la posición del robot sin planificador.</i> .....	104
<b>Tabla 42</b> <i>Prueba de repetibilidad del plan del robot con el planificador.</i> .....	105

## Índice de Figuras

<b>Figura 1</b> Esquema de un sistema de manufactura automatizado.....	22
<b>Figura 2</b> Esquema de un robot y su interacción con el entorno.....	25
<b>Figura 3</b> Configuraciones de los manipuladores Robóticos.....	27
<b>Figura 4</b> Esquema de un manipulador robótico de configuración PRR.....	28
<b>Figura 5</b> Transiciones de un Plan .....	29
<b>Figura 6</b> Esquema de un sistema de planificación.....	30
<b>Figura 7</b> Planificación como búsqueda .....	35
<b>Figura 8</b> Pseudocódigo Forward .....	36
<b>Figura 9</b> Pseudocódigo Backward.....	36
<b>Figura 10</b> Shakey con sus principales componentes. ....	37
<b>Figura 11</b> <i>Flujograma de un planificador STRIPS.</i> ....	39
<b>Figura 12</b> Representación del modelo HSV .....	42
<b>Figura 13</b> Boceto del esquema del Robot PRR. ....	48
<b>Figura 14</b> Esquema del Robot en el eje X-Z .....	49
<b>Figura 15</b> Esquema de los eslabones del robot (codo abajo) en el eje X-Y .....	50
<b>Figura 16</b> Esquema de los triángulos que forman los eslabones. ....	50
<b>Figura 17</b> <i>Curva Torque vs Velocidad del motor Nema 17 a 12 VDC.</i> .....	56
<b>Figura 18</b> Mecanismo de etapa linear motorizada de eje único con transmisión de tornillo de bolas.....	58
<b>Figura 19</b> <i>Análisis de esfuerzos para el eslabón con mayor carga.</i> ....	64
<b>Figura 20</b> Primeros bosquejos de la celda de trabajo y manipulador. ....	66
<b>Figura 21</b> Celda de trabajo detallada.....	67
<b>Figura 22</b> Diseño de los cilindros .....	69
<b>Figura 23</b> <i>Manipulador vista desde el eje X,Y</i> .....	70
<b>Figura 24</b> <i>Representación del área de trabajo obtenida al usar (a) servomotores y (b) motores a paso.</i> .....	71
<b>Figura 25</b> Pasos para procesar la imagen con la cámara.....	79
<b>Figura 26</b> Rangos de HSV para cada color .....	79
<b>Figura 27</b> Espacios de trabajo de la Celda Robótica .....	81
<b>Figura 28</b> Diagrama general del Subsistema Planificador .....	82
<b>Figura 29</b> Montaje de los motores – articulaciones rotacionales. ....	88

<b>Figura 30</b> Montaje del motor – articulación prismática.....	88
<b>Figura 31</b> Conjunto articulación prismática del manipulador .....	89
<b>Figura 32</b> <i>Eslabones impresos mediante SLA (a)Eslabón ‘brazo 1’ y (b) Eslabón ‘brazo 2’.</i> ....	89
<b>Figura 33</b> <i>Banda dentada de sincronización de perfiles curvilíneos.</i> .....	90
<b>Figura 34</b> <i>Polea de aluminio para motor a pasos.</i> .....	90
<b>Figura 35</b> Implementación del sistema de transmisión por correa y poleas.....	92
<b>Figura 36</b> (a) <i>Diseño para corte</i> (b) <i>Construcción de la celda de trabajo</i> .....	92
<b>Figura 37</b> <i>Sistema de Visión e Iluminación de la celda</i> .....	93
<b>Figura 38</b> Implementación total de la celda de trabajo (a) en Software CAD y (b) Real. ....	93
<b>Figura 39</b> <i>Esquema del sistema eléctrico y electrónico del manipulador</i> .....	94
<b>Figura 40</b> <i>Montaje del sistema eléctrico y electrónico del manipulador</i> .....	94
<b>Figura 41</b> <i>Prueba de reconocimiento de colores</i> .....	95
<b>Figura 42</b> Flujograma del sistema de visión artificial.....	96
<b>Figura 43</b> Acciones del sistema descritas en STRIPS.....	97
<b>Figura 44</b> Ejemplo de planificación.....	98
<b>Figura 45</b> Diagrama de los nodos de la Celda Robótica.....	99
<b>Figura 46</b> Navegación entre pantallas de la Interfaz .....	100
<b>Figura 47</b> <i>Pantalla principal.</i> .....	101
<b>Figura 48</b> Pantalla Modo Libre.....	101
<b>Figura 49</b> Interfaz gráfica: ingreso de las condiciones iniciales.....	102
<b>Figura 50</b> <i>Interfaz gráfica: ingreso del objetivo del sistema.</i> .....	102
<b>Figura 51</b> Interfaz gráfica: Plan elaborado con STRIPS. ....	103
<b>Figura 52</b> <i>Plan usado para la prueba del robot con el planificador.</i> .....	105
<b>Figura 53</b> Esquema de trabajo conjunto de manipuladores con Planificación .....	110
<b>Figura 54</b> Esquema de integración de planificaciones en un Robot móvil.....	110

## Resumen

El documento detalla el desarrollo del diseño, construcción e implementación de una celda robótica inteligente basado en un planificador de tipo “STRIPS”, desde el diseño cinemático y dinámico de un manipulador PRR (Prismático/Rotacional/Rotacional) para el dimensionamiento de los motores y los demás componentes electrónicos, hasta la construcción e implementación total de la celda robótica para la clasificación de elementos. En los primeros capítulos se describen los objetivos, la justificación e importancia del proyecto, el marco teórico que permite entender las definiciones básicas sobre el manipulador y algunos planificadores. En los siguientes capítulos se describen el diseño realizado para la construcción e implementación de la celda con el manipulador, usando una selección de alternativas para cada subsistema, para el control de todo el sistema se usa una Raspberry Pi, por lo que la pantalla táctil y cámara de la visión artificial son para Raspberry, la programación de nodos se hace en ROS (Robotics Operating System) con un gestor de acciones para el funcionamiento de cada etapa, obteniendo así, el funcionamiento correcto de la celda robótica. Las pruebas de funcionamiento son de gran importancia en el proyecto para conocer el rendimiento de la celda, por lo que se realizan pruebas de posicionamiento y repetibilidad sin el planificador para determinar la precisión de los motores del manipulador, tomando en cuenta que los motores usados en este proyecto no tienen una gran precisión el error obtenido en las pruebas es, sin embargo, aceptable. De la misma manera, se realiza una prueba de repetibilidad con el planificador para determinar el funcionamiento del mismo, logrando obtener una eficiencia de más del 95%.

### **PALABRAS CLAVES:**

- **MANIPULADOR PRR**
- **STRIPS**
- **CELDA ROBÓTICA**

## **Abstract**

The paper details the development of, design, construction, and implementation of a robotic intelligent cell, based on a “STRIPS” kind of plane, from the cinematic and dynamic design of a PRR (Prismatic/Rotational/Rotational) manipulator for the sizing of the engines and the rest of the electronic components, up to the construction and full implementation of the robotic cell for the cylinder classification. The first chapters describe the objectives, justification and importance of the project, the theoretical framework allow us to understand the basic definitions about manipulator and some planners. The next chapters describe the design for the construction and implementation of the manipulator cell, using a selection of alternatives for every subsystem, to control the entire system we use a Raspberry Pi, so the touch screen and the artificial vision camera are for the Raspberry, nod programing is made on ROS (Robotics Operating System) with an action manager for the operation of each stage, thus obtaining the correct operation of the robotic cell. Functional tests are of great importance in the project to recognize the performance of the cell, so, positioning and repeatability tests are performed, without the planner to determinate the precision of the manipulator engines, acknowledging that the engines used in this project don't have a great precision the error obtained in the test is, however, acceptable. In the same way, a repeatability test is performed with the planner to determinate the way it operates, accomplishing an efficiency of more than 95%.

### **KEYWORDS:**

- **MANIPULATOR PRR**
- **STRIPS**
- **ROBOTIC CELL**

## Capítulo 1

### Descripción general del proyecto

En el siguiente capítulo se describe la introducción, planteamiento del problema, justificación e importancia, objetivos, alcance y un breve resumen del desarrollo del proyecto.

### Introducción

En la actualidad vemos como se presenta la necesidad de que las máquinas actúen de una forma eficiente y segura por sí solas, como puede ser el ejemplo de autos autónomos y celdas de manufactura completamente automatizadas que son capaces de armar un auto en su totalidad con la mínima intervención de una persona que pudiera dirigir las acciones. Para lograr esta independencia se utiliza la Inteligencia Artificial que actúa como un cerebro para una máquina, o más específicamente, para un robot, dotándole de la capacidad para percibir el mundo que lo rodea con la ayuda de todo tipo de sensores y poder interactuar sobre este de una forma lógica.

Una de las áreas de investigación de la inteligencia artificial es la planificación automatizada, cuyo objetivo es generar automáticamente planes para pasar de un estado del mundo, considerado como el estado inicial y conocido, a otro estado o conjunto de condiciones del mundo, considerado como el estado objetivo o deseado. Los planes resultantes pueden ser ejecutados por robots o agentes y sus aplicaciones pueden ser útiles en diversas disciplinas, como logística militar, los videojuegos o la robótica de todo tipo. La planificación automática es un proceso de razonamiento abstracto que utiliza la computación para seleccionar y organizar actividades con el fin de lograr un resultado deseado. En otras palabras, es un proceso de búsqueda de actividades en un orden determinado para lograr un objetivo.

En un ambiente industrial, implementar un planificador que gobierne las acciones de un robot o toda una celda de trabajo puede traer consigo beneficios instantáneos y evidentes como lograr que una máquina llegue por sí sola a una meta que puede llegar a ser demasiado compleja si no se efectúan acciones en una secuencia concreta que permita llegar y cumplir con ese objetivo, Guzmán (2015), Además, este tipo de planificación automatizada puede servir de apoyo para procesos ya automatizados optimizando recursos o reduciendo tiempos.

En la planificación automática, se tienen tres tipos de planificadores dependiendo de la forma en la que este trabaja con el dominio: específicos de dominio, independientes del dominio y configurables de dominio.

Los planificadores específicos de un dominio se adaptan a un dominio concreto y no funcionan en otros dominios. Por otro lado, un programador independiente del dominio puede ejecutarse en cualquier dominio que satisfaga una serie de condiciones. Entre estos planificadores se encuentran los planificadores clásicos, incluyendo el planificador de tipo STRIPS (Stanford Research Institute Problem Solver), que utilizamos en este proyecto. Por último, los planificadores de dominio configurables tienen dos características: son independientes del dominio e incluyen el conocimiento necesario del dominio en su entrada para restringir y acortar la búsqueda de una solución.

### **Planteamiento del problema**

La automatización de procesos como puede ser un sistema de manufactura que trabaja con Robots industriales definidos por la RIA (Robot Industry Association) como manipuladores reprogramables multifuncionales diseñados para cumplir tareas específicas, conlleva a que una de las partes más importantes de este sistema sea el software controlador programado que gestiona y coordina las acciones que se deben realizar para su funcionamiento, de este dependen cómo y cuándo se van a realizar estas acciones, influyendo directamente no solo en que el sistema sea capaz de cumplir con una tareas sino también en la eficiencia y su capacidad para responder frente a un posible error.

Desde la perspectiva de un manipulador industrial (Robot), un programa es una secuencia de posiciones y acciones como, por ejemplo, el de abrir o cerrar una pinza, que se lo debe especificar para que el robot resuelva un supuesto problema. El robot toma estas indicaciones y mediante algoritmos y señales eléctricas activa o desactiva los motores o actuadores según se las instrucciones que se le dieron realizando de esta forma algún tipo de acción como puede ser el de soldar o pintar. Por otra parte, desde la perspectiva del programador existen dos formas de lograr que el Robot realice la acción deseada, por un lado, está el programar cada acción y movimiento que el Robot debe efectuar y por el otro formular la tarea a realizar mediante dos situaciones, una inicial y otra final sin indicarle de manera explícita la forma de pasar de la una a la otra.

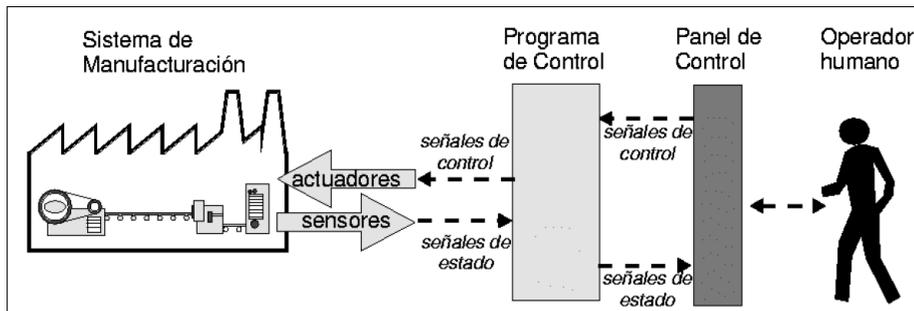
De acuerdo con Guzmán (2003), el diseñar el programa de control secuencial que gobierna y dirige un sistema de manufactura (Figura 1) de forma “tradicional”, es decir, dando cada acción detallada al robot para cumplir la tarea, se ve expuesto a aspectos negativos por el hecho de ser

una tarea ardua y en muchos casos difícil, que tradicionalmente se ha llevado a cabo de forma manual por personal de control especializado:

- Es un proceso que requiere de demasiado tiempo de ingeniería y su complejidad se puede dar lugar a errores humanos.
- Cada vez que el sistema de manufactura requiera algún cambio o mejora, este proceso se complica aún más al estar estrechamente ligado a su ciclo de vida.
- Al presentarse un error, el sistema de manufactura solo puede responder y actuar dentro de lo ya programado. (pp. 25-31)

**Figura 1**

*Esquema de un sistema de manufactura automatizado.*



Nota. Tomado de Guzmán (2003).

### **Justificación e Importancia**

En la industria actual, y especialmente en los sistemas de manufactura automatizados que trabajan con manipuladores robóticos, como ya se mencionó y como lo explican Lagos C., Buitrago J. & Timarán R. (2010), las técnicas de planificación con inteligencia artificial han tomado un relevante protagonismo en la eficiencia de la producción con bajo costo.

La resolución de problemas, en el ámbito industrial, se ha dado mediante las técnicas de planificación con inteligencia artificial, mismas que apoyan a la secuencia óptima de operaciones para obtener un producto final eficiente en todo sentido.

Se toma en consideración que, una secuencia de operaciones para realizar una tarea específica requiere de un planificador que mediante algunas pautas generales y la descripción cinemática y geométrica del manipulador robótico genera las instrucciones de movimiento y

acciones que resuelvan la tarea, aliviando no solo el trabajo de programación no automatizada, sino que flexibilizando toda una línea de manufactura si fuese el caso.

El uso de la Inteligencia Artificial con la planificación automatizada se plantea como la mejor opción de solución y alternativa a implementar en manipuladores robóticos. En lo que refiere a planificadores automatizados hoy en día existe una gran variedad de ellos con sus propias ventajas y limitaciones, pero en base al estudio de Vassos (2012), el planificador STRIPS se presenta como un modelo lineal y sencillo de la planificación clásica, y es el tipo de planificador elegido para el proyecto por las siguientes razones:

El modelo de planificación STRIPS implementa la planificación lineal y es adecuado para el proceso de mecanizado propuesto en el proyecto, ya que se trata de un solo manipulador robótico equipado con los sensores necesarios y controlado por un sistema de control secuencial, de modo que sólo se ejecutará un paso a la vez.

Tiene un modelo de acción muy simple que probablemente sea utilizado por la mayoría de los planificadores.

El algoritmo STRIPS es uno de los más estudiados sobre sistemas de planificación, por lo que se facilitó su análisis y posterior adaptación a la celda robótica.

## **Objetivos**

### ***Objetivo general***

Diseñar y construir una celda robótica inteligente mediante el planificador tipo "STRIPS" para tareas de clasificación.

### ***Objetivos específicos***

- Diseñar y construir un manipulador de 3 grados de libertad para la celda robótica considerando la cinemática, dinámica y control.
- Diseñar y construir una celda de trabajo con un sistema de detección de bloques basado en visión artificial para la implementación de la lógica de planificación.
- Programar un sistema de planificación tipo STRIPS con una jerarquía de acciones para la implementación de tareas de clasificación en bloques.
- Realizar pruebas de funcionamiento y registrar los resultados para evaluar el rendimiento de la celda robótica inteligente.

## **Alcance**

El proyecto consiste en el diseño y construcción de una Celda Robótica Inteligente para la clasificación planificada de elementos didácticos mediante un manipulador robótico de 3 grados de libertad de configuración PRR (Prismático, Rotacional, Rotacional) con el uso de reconocimiento por visión artificial y planificación STRIPS.

En primera instancia el proyecto consistirá de un diseño cinemático, dinámico y de control del manipulador PRR. Una vez diseñado y construido el manipulador y la celda sobre la que trabajará, se programarán los nodos de control de movimiento y acciones a efectuar dadas por un planificador offline diseñado en STRIPS para solucionar los problemas de clasificación que se le planteen.

El control total de la Celda Robótica se efectuará mediante una interfaz HMI montada en una pantalla táctil en la misma Celda. Los elementos didácticos serán cinco pequeños cilindros etiquetados por colores.

## **Estructura del documento**

Para el desarrollo del proyecto se tiene cinco capítulos, dentro del capítulo 1 se encuentra detallado los aspectos generales del proyecto como: descripción, justificación y objetivos.

En el capítulo 2 se presenta el Marco Teórico del proyecto, en el que se detalla las definiciones básicas y específicas del manipulador y planificadores.

En el capítulo 3, se detalla cada módulo y subsistema del proyecto, éstos son: diseño y control de un manipulador, diseño de la celda de trabajo, programación de la planificación de tareas y desarrollo del software para implementación, análisis y simulación.

El capítulo 4, contiene la construcción del manipulador, celda de trabajo, al igual que la implementación del planificador de tareas que se encargará del funcionamiento de toda la celda de trabajo, además, se realiza las pruebas del manipulador en la celda robótica, pruebas de la interacción con el planificador y resultados comparativos del rendimiento de la misma.

Finalmente, en el capítulo 5, se redactan conclusiones que se obtuvieron durante el desarrollo del proyecto, recomendaciones para la optimización de recursos en el campo industrial, así como también para la conservación de la celda robótica y se anexan planos y los manuales de uso y mantenimiento correspondientes.

## Capítulo 2

### Estado del arte

Durante el desarrollo del capítulo se presentan conceptos sobre manipuladores robóticos, planificación automática, visión por computador, haciendo énfasis en el planificador STRIPS.

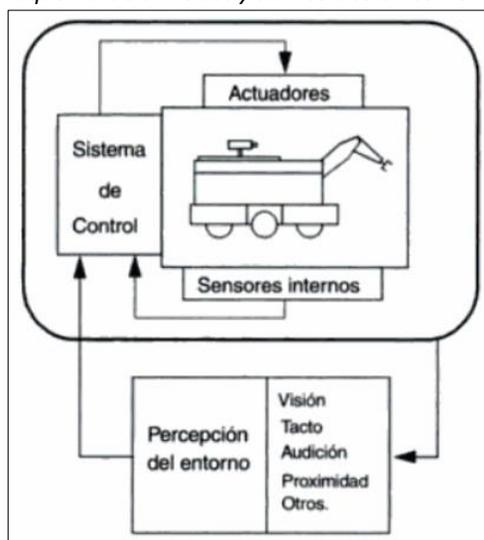
### Manipuladores Robóticos

Un robot es una máquina que integra elementos mecánicos, eléctricos, electrónicos y de comunicación con sistemas informáticos de control en tiempo real, detección del entorno y programación según Ollero (2001), la robótica industrial puede considerarse un campo de la robótica que aumenta sustancialmente la flexibilidad del proceso de producción manteniendo la productividad que antes se conseguía con máquinas automatizadas específicas.

Puede decirse que los robots industriales se desarrollaron y se siguen utilizando con el único fin de sustituir las actividades directas de los trabajadores en la cadena de producción, pero sustancialmente son utilizados para realizar tareas en zonas de difícil acceso, con riesgo de accidentes, con condiciones peligrosas para la salud, con objetos de gran tamaño y, por tanto, difíciles de trabajar, etc.

**Figura 2**

*Esquema de un robot y su interacción con el entorno.*



**Nota:** Tomado de Ollero (2001)

Desde sus inicios, los robots industriales se han centrado en la función de manipulación. De hecho, está ampliamente aceptado que los robots industriales son esencialmente robots de manipulación. Muchos de los robots industriales actuales son esencialmente articulados. Según la definición de la Asociación Americana de Robótica, un robot industrial es un manipulador multifuncional y programable que está diseñado para mover materiales, piezas, herramientas o equipos especializados en una variedad de movimientos y está programado para realizar diversas tareas.

La descripción básica de un sistema robótico consta de un sistema mecánico, actuadores, sensores y sistemas de control. En cuanto al tratamiento de la información, la robótica incluye el control del movimiento, la detección y la planificación.

Enfocándonos en la planificación en manipuladores industriales, el objetivo es encontrar un patrón o trayectoria desde una posición o estado inicial hasta un estado objetivo (por ejemplo, crear una trayectoria o realizar una tarea) minimizando ciertos índices. En el caso más sencillo, el problema se plantea en un entorno que se asume determinado e invariable.

Teniendo en cuenta los sistemas y las funciones programables con las que puede contar un robot y la flexibilidad que puede llegar a tener dentro de un ambiente industrial son las razones por las que hoy en día con la necesidad de producir a altas velocidades con bajos costos y bajas tasas de error en las líneas de producción industrial, las industrias optan por implementar diferentes tipos de robots en aplicaciones industriales.

### ***Tipos de manipuladores Robóticos en la industria***

Hay muchos tipos diferentes de robots industriales según la configuración de sus articulaciones, cada uno de ellos diseñado para realizar una tarea específica de la forma más adecuada.

Los **robots cartesianos** o de coordenadas cartesianas están equipados con tres articulaciones deslizantes denominadas PPP. Las principales características de estos robots son su pequeña área de trabajo y su gran rigidez mecánica. Además, según Rosario (2005), este tipo de manipulador se caracteriza por una gran precisión de la posición del actuador.

Los **robots cilíndricos** o de coordenadas cilíndricas se componen básicamente de tres articulaciones, una rotacional y dos prismáticas (RPP). como lo indica COCOTA (2013), la combinación de estas articulaciones permite al robot realizar movimientos rotacionales y lineales

en un espacio de trabajo ligeramente mayor que el de un robot con coordenadas cartesianas. Debido a la naturaleza bastante compleja de los movimientos de las articulaciones y su control, la rigidez mecánica es ligeramente inferior a la de un robot con coordenadas cartesianas.

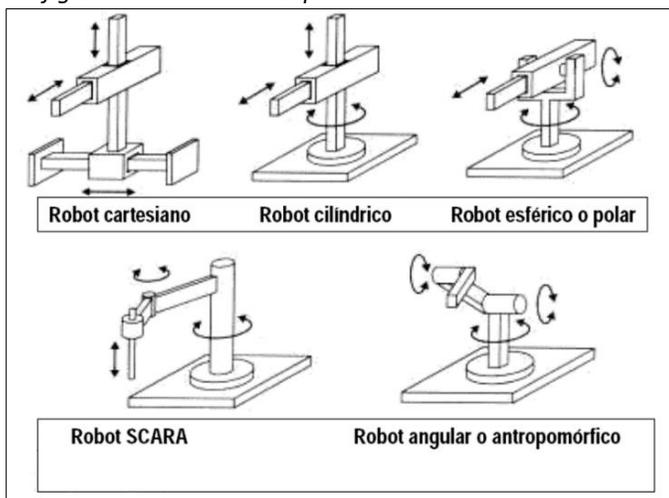
Los **robots esféricos** o de coordenadas esféricas (RRP) tienen un espacio de trabajo mayor y una rigidez mecánica menor que los robots cilíndricos. Este tipo de robot está equipado con dos movimientos de rotación y su control es aún más complejo que el de los robots cilíndricos.

Los robots de **articulación horizontal** (RRP), también conocidos como robots SCARA, no tienen estructura de articulación y constan de dos articulaciones rotativas y una prismática. Según BAYER (2011), estos robots se utilizan principalmente para tareas de ensamblaje, normalmente con cuatro grados de libertad, y son muy utilizados en la industria electrónica para fijar placas de componentes a placas de circuitos impresos por su rapidez y precisión.

Los **robots articulados** (Figura 3), tanto vertical como horizontalmente (RRR) tienen más grados de libertad y una mayor área de trabajo. Estos tipos de robots suelen funcionar con accionamientos eléctricos. Sin embargo, tienen un gran inconveniente: tienen una cinemática científica compleja y sus movimientos lineales son difíciles de controlar. Como menciona BAYER (2011), su versatilidad y su gran capacidad de carga mecánica lo convierten en el modelo más utilizado en la industria, realizando tareas como mover piezas, soldar, pintar y paletizado principalmente.

**Figura 3**

*Configuraciones de los manipuladores Robóticos.*



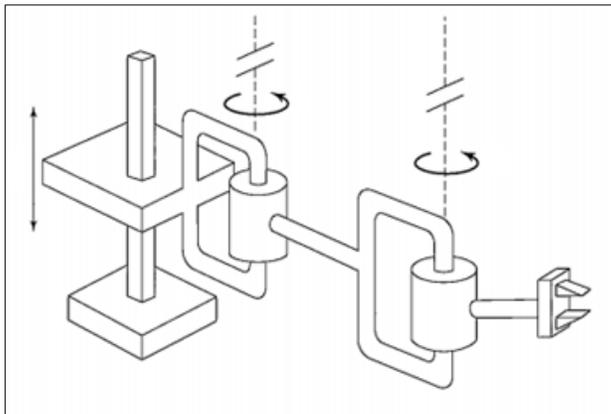
**Nota: Tomado de WordPress (2010)**

### **Manipulador de configuración PRR**

Un robot de configuración PRR corresponde a un tipo de robot SCARA de tres grados de libertad en el que sus articulaciones tienen una configuración prismática, rotacional y rotacional. Presentan facilidad para movimientos lineales en dirección vertical que solo son controlados por la junta prismática de la primera articulación. Presenta las ventajas de un típico robot SCARA con alta precisión y velocidad, pero con la cualidad de que el motor de la articulación prismática levanta toda la estructura del robot, por lo tanto, su motor tiene un par mayor lo que lo hace adecuado para aplicaciones donde levantar cargas pesadas de forma lineal es el reto. En este proyecto, se diseña y construye un robot manipulador de tipo PRR (Prismático, Rotacional, Rotacional).

**Figura 4**

*Esquema de un manipulador robótico de configuración PRR*



**Nota: Tomado de WordPress (2010)**

### **Inteligencia Artificial**

La Inteligencia Artificial es un campo que pertenece a la informática, nacido en la época posterior a la Segunda Guerra Mundial. El objetivo de la Inteligencia Artificial es estudiar el comportamiento de las máquinas inteligentes y desarrollar máquinas que puedan percibir, razonar, comunicarse y actuar tan bien o mejor que los humanos.

Existen diferentes enfoques de la inteligencia artificial que pueden dividirse según dos aspectos. Por un lado, estos enfoques pueden implicar procesos mentales y de razonamiento o comportamiento y, por otro lado, implican una racionalidad o acción lo más cercana posible a las formas humanas.

En el campo de la inteligencia artificial se pueden distinguir diferentes subcampos, como el procesamiento del lenguaje natural, el aprendizaje automático, la representación del conocimiento, la robótica, la búsqueda en el espacio de estados o la planificación automática, que es el campo de este proyecto.

### Planificación Automática

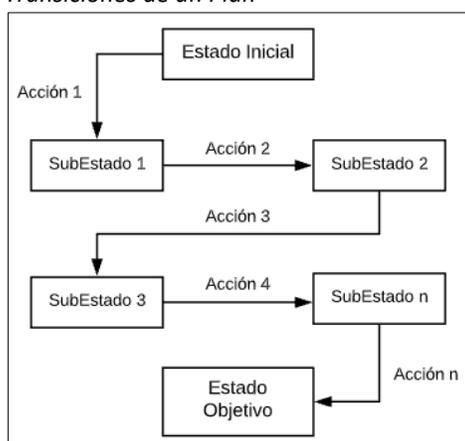
Antes de empezar a describir cómo trabajan los planificadores, debemos saber qué es la planificación. La planificación según Guzmán & Gutiérrez (2012), es el proceso de encontrar y organizar una serie de acciones, denominadas plan, para alcanzar un objetivo.

Para ser eficaz, un planificador debe ser capaz de dividir los problemas en cuestiones más pequeñas o subproblemas, en lugar de tratar todo como un único problema. Esto es una desventaja porque la mayoría de los problemas no pueden descomponerse; por lo tanto, la mayoría de los planificadores se centran en los problemas que pueden descomponerse durante el diseño. De hecho, los planificadores se basan en la suposición de que los problemas en entornos realistas pueden descomponerse. Al poder descomponer un problema en subproblemas, el planificador puede procesar estos subproblemas de forma independiente, aunque debe combinar los resultados al terminar.

Basándose en los parámetros de entrada, el planificador tiene que cambiar el estado del mundo para alcanzar el estado objetivo (Figura 5), para lo cual utiliza las acciones recibidas en los parámetros de entrada, que también incluyen el estado inicial y la meta o conjunto de objetivos (estado objetivo).

**Figura 5**

*Transiciones de un Plan*



El planificador necesita una entrada en la que se especifican los siguientes elementos: la descripción del sistema donde se detalla en si como funciona este, una descripción del momento inicial del mundo y el estado objetivo o meta que está representado por los hechos deseados en el estado del mundo al finalizar el plan.

Por otro lado, la salida del planificador es un plan para llegar del estado inicial al estado meta, una secuencia de acciones que el agente (según sea el caso) debe realizar en el mundo.

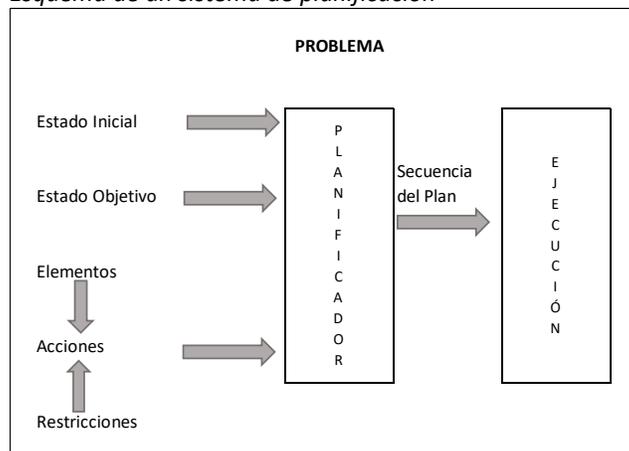
La planificación se puede usar en varios campos como en la robótica usada, en su gran mayoría, para generar trayectorias a robots móviles, simulación de juegos, logística militar, celdas de trabajo en fábricas y en gestión de evacuaciones e incendios. En la guía de Sistemas Expertos e Inteligencia Artificial (2015), la idea de un planificador es saber representar el objetivo que se quiere alcanzar, representar las acciones, el entorno y la capacidad de generar un plan adecuado.

En su artículo Valenzuela, Moreno, Cedeño, & Valenzuela (2014), manifiestan que un agente planificador debe tener ciertas características como la autonomía de operar por sí mismos sin necesidad de un operador humano, habilidad para responder oportunamente a ciertos ambientes y sobre todo la proactividad para tomar la iniciativa de actuar y conseguir los objetivos.

Arias, Jiménez & Ovalle (2009), presentan gráficamente, (Figura 6), una planificación en la que consideran que las acciones deben estar sobre un conjunto de elementos que son propios del dominio, para cumplir con las restricciones definidas, que se reúnen en el planificador, para obtener un plan secuencial llegando a la etapa de ejecución.

**Figura 6**

*Esquema de un sistema de planificación*



*Nota.* Tomado de Arias, Jiménez, & Ovalle (2009)

El proceso de planificación puede realizarse desde dos enfoques diferentes según sea el caso, conocido como planificación offline y online, relacionado estrechamente con la dinámica del mundo. En el modelo más limitado, la planificación offline, el planificador realiza el proceso de planificación sin tener en cuenta la dinámica del sistema. En cambio, la planificación en online significa que el planificador no puede ignorar cómo evoluciona el sistema, aunque no tenga que ocuparse de todos los detalles. Por ello, es importante comprobar si el plan sigue siendo válido y, en caso contrario, volver a calcularlo.

### ***Tipos de planificadores***

Según Nau (2007), se pueden distinguir tres tipos de planificadores en función de la configuración del dominio en el que se ejecutará el planificador, los cuales se muestran a continuación.

#### **Planificadores de dominio específico.**

Este tipo de planificador están pensados para ejecutarse en un dominio específico. Si se quiere ejecutar este planificador en un dominio distinto al que fue diseñado, habría que realizarle cambios importantes. Se podía decir que es el más desarrollado en la práctica.

#### **Planificadores independientes del dominio**

Son planificadores diseñados para funcionar sin importar el dominio, siempre que cumpla una serie de requisitos previos. Sin embargo, existe un inconveniente evidente: un planificador no será igual de eficaz en todos los dominios, es decir, para algunos problemas el planificador será más eficaz que en otros.

#### **Planificadores de dominio configurable**

Es un planificador que no tiene dependencia directa del dominio, acepta como entrada conocimientos sobre el propio dominio para que el planificador pueda acotar su búsqueda y minimizar así el espacio de búsqueda en el que debe operar.

Aunque los planificadores de dominio específico son ventajosos para resolver problemas de forma específica adaptando el planificador, el coste de crear un planificador para cada nuevo problema es elevado y no se tiene en cuenta los aspectos comunes que pueden tener los distintos planificadores como lo presentan Ghallab, Nau, & Traverso (2004). Por lo tanto, los investigadores

tienden a centrarse en planificadores independientes del dominio concretamente en problemas que forman parte de la planificación clásica.

### ***Planificación Clásica***

La planificación clásica pertenece a la planificación automática, es un campo de esta. Su diseño y forma de actuar son del tipo de planificación independiente del dominio y es uno de los más estudiados por investigadores. El sistema representado en la planificación clásica es el que presenta la mayor cantidad de restricciones, pero por su parte, es también el más simple de todos los modelos posibles. En la planificación clásica, el sistema debe cumplir con una serie de supuestos estrictos: el sistema es finito, en su totalidad debe ser observable, determinista y estático, cada cambio del estado inicial lleva al sistema únicamente al estado objetivo, el plan resultante nos entrega una secuencia simple de acciones a ejecutar, el planificador supone que los cambios que se producen de un estado A a un estado B consecutivo se realiza en un instante de tiempo y, por último, se trata de una planificación fuera de línea u offline, fuera del conocimiento del planificador no ocurre nada en el sistema.

Representar un problema de planificación clásica se lo puede hacer de tres formas: **representación de teórica de conjuntos** donde un estado es una colección de proposiciones y una acción es una representación sintáctica mediante proposiciones; mediante una **representación de las variables de estado** en el cual un estado se representa por un conjunto de variables de estado; y mediante la **representación clásica** que representa sus acciones y estados de una forma similar a la representación de conjuntos salvo que se utilizan literales de primer orden como puede ser  $M(x)$  y cambia el uso de proposiciones por conectores lógicos.

### **Representación Clásica**

Como hemos visto, las representaciones clásicas utilizan representaciones teóricas de conjuntos con predicados en lugar de proposiciones y un lenguaje lógico de primer orden compuesto por varios símbolos como variables, constantes, funciones, relaciones, símbolos lógicos y auxiliares según la publicación de “Lenguajes de primer orden” de la Universidad de Granada (2015), para las representaciones clásicas, la lógica de primer orden tiene un número finito de símbolos y constantes.

Para definir un problema de planificación en esta representación clásica se requiere del estado inicial del mundo o el problema, las condiciones o requerimientos del estado objetivo y los estados que satisfacen las condiciones del objetivo.

### **Representación de Estados**

Para indicar, en la representación clásica, cómo está ordenado el mundo y sus objetos se define este estado por medio de una secuencia de literales lógicos positivos. Estos literales deben representar de una manera simple lo que se desea decir y no deben contener dependencias funcionales; por ejemplificar, se admite una forma como  $On(A, B)$ , pero no  $OnTable(On(A, B))$ . Si el literal no está en la definición de estado, se lo toma como falso.

### **Representación de Operadores y Acciones**

Los operadores son acciones sin unidad, es decir, con variables representadas por tres partes.

**Nombre:** Especifica el nombre del operador en el problema. Es una lista de variables, algunos ejemplos pueden ser  $move(A, B)$  o  $catch(A)$ . No puede haber dos operadores con el mismo nombre en el mismo problema como  $catch(A)$  y  $catch(A, B)$  no están permitidos.

**Precondiciones:** No es más que un conjunto de predicados que deben satisfacerse en el estado en el que se encuentre para poder realizar la acción. Estos literales pueden ser positivos o negativos. El hecho de que el literal sea positivo quiere decir que debe estar en el estado actual, mientras que si por su parte es negativo significa que no se debe de encontrar en el estado actual.

**Postcondiciones:** Las postcondiciones se pueden definir como las consecuencias que tiene la acción sobre el sistema. Son de igual forma un conjunto de predicados con la misma forma y posibilidad de ser positivos o no.

### **Lenguajes de Planificación Automática**

Para resolver los problemas de planificación es necesario representar los estados, acciones y objetivos mediante un lenguaje que debe representar cada factor para describir un problema, pero también debe permitir que los algoritmos sean operativos y eficientes, así lo determinan los Sistemas Expertos e Inteligencia Artificial (2015).

En el campo de la planificación automatizada, se tienen lenguajes especializados llamados lenguajes de acción, son modelos que describen de una forma clara los efectos de las acciones

sobre el sistema. Se los puede dividir en: los lenguajes de descripción de acciones, utilizados con el fin de dar una descripción de los sistemas de transición, y los lenguajes de consulta de acciones, que nos indican afirmaciones sobre un sistema de transición de estados como lo indican Gelfond y Lifschitz (2015).

Los lenguajes de descripción de acciones incluyen los lenguajes de descripción de acciones (ADL) y STRIPS. Dentro de estos lenguajes, también hay diferentes variantes que han evolucionado con el tiempo.

Los lenguajes de descripción de acciones son representados en su mayoría por STRIPS y ADL. STRIPS ha sido desarrollado como un lenguaje para el planificador del mismo nombre como se verá más adelante, y es la base de una gran parte de los lenguajes de planificación. STRIPS presenta serios problemas al tratar de representar dominios reales en él debido a que se trata de un lenguaje diseñado para la resolución de problemas simples, pero de una forma eficiente. De ahí el desarrollo de varios lenguajes a partir de este, como el ADL.

ADL es un lenguaje que evolucionó a partir de STRIPS tratando de solventar las falencias de este y que entre sus principales diferencias o avances se puede mencionar que trabaja con un mundo abierto en lugar de uno cerrado lo que significa que literales que no aparecen en los estados no sean considerados como falsos si no que se los toma como desconocidos, permite trabajar con literales negativos en estados, soporta tipos y cuantificar variables, entre otros.

Existe además un lenguaje denominado Planning Domain Definition Language (PDDL), que trata de unificar los diferentes tipos de lenguajes incluyendo STRIPS y ADL recopilando las distintas formas de representación de planificación en un estándar permitiendo compartir resultados y problemas entre investigadores.

### **Planificación en el Espacio de Estados**

Resolver un problema de planificación se puede definir como encontrar el plan o camino que se debe seguir para llevar al sistema de un estado inicial a un estado objetivo y para eso se requiere de un “buscador” (Figura 7). En esta sección se describe el primer enfoque de planificación clásica, que contiene los algoritmos de búsqueda más sencillos para resolver los problemas de este tipo de planificación.

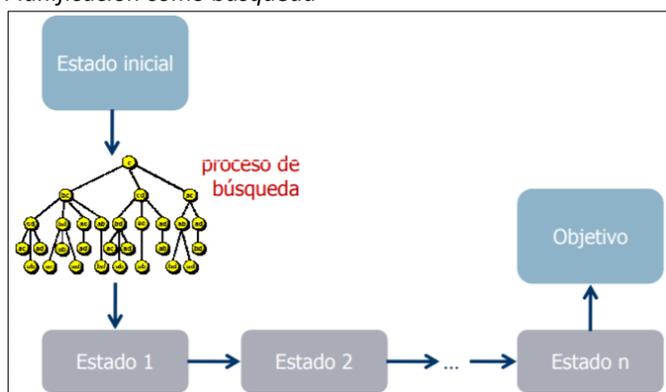
Los algoritmos de búsqueda del espacio de estados contemplan cuatro elementos con los que trabajan y son: el **espacio de búsqueda** que se puede definir como un subconjunto del espacio de

estados del sistema, los **nodos** que se pueden corresponder con los estados del sistema, los **arcos** con las transiciones entre estados y el **camino de solución** que representa el plan o la solución del problema de planificación. En este tipo de planificador se pueden distinguir dos tipos de algoritmos de búsqueda diferentes: la búsqueda hacia delante y la búsqueda hacia atrás.

Planificar en el espacio de estados se basa en la idea de aplicar un algoritmo de búsqueda a un problema de planificación, como la búsqueda en profundidad, la búsqueda en amplitud o la búsqueda A\* según la publicación “State-Space Search and the STRIPS Planner” de la Universidad de Edimburgo (2015).

**Figura 7**

*Planificación como búsqueda*



*Nota.* Tomado de Berzal (2013)

### **Busqueda hacia adelante**

Para un algoritmo de búsqueda hacia adelante según Russell y Norvig (2010), es necesario definir los siguientes puntos:

**Estado inicial:** es el estado inicial del problema. Se supone que los predicados son simples y positivos, mientras que los predicados que no se declaran se consideran falsos.

Las acciones que se aplican a un estado son aquellas cuyas precondiciones se satisfacen, los predicados positivos deben estar presentes en estado y por su parte los negativos no. El estado cambia al incluir las postcondiciones positivas y eliminar las postcondiciones negativas.

**Función objetivo:** comprueba si el problema está resuelto comparando los predicados de un estado con el conjunto de predicados objetivo.

**Costo de la acción:** cada acción puede tener un costo, si no, cada acción tendría un costo de 1.

**Figura 8***Pseudocódigo Forward*

```

Busqueda(O, s, g)
  estado <- s
  plan <- []
  Bucle
    Si s satisface g
      devolver plan
    Sino
      Obtener acciones que se pueden aplicar
      Si accionesPosibles esta vacia
        devolver fallo
      Sino
        Seleccionar una accion
        aplicar accion sobre s
        plan <- plan \cup accion

```

El algoritmo de búsqueda *forward* (Figura 8) es ineficiente por dos principales razones: en primer lugar, se consideran todas las acciones que satisfacen las precondiciones, incluso si algunas de ellas son irrelevantes; en segundo lugar, tiene un gran problema de atascarse, por ejemplo, cuando se realiza una búsqueda en profundidad, el algoritmo puede entrar en un bucle infinito de una rama de la búsqueda, haciendo que el algoritmo nunca encuentre una solución.

**Búsqueda hacia atrás**

Como la búsqueda hacia adelante plantea problemas como la creación de estados irrelevantes e innecesarios y utiliza un espacio de búsqueda demasiado grande, se introdujo la búsqueda hacia atrás (Figura 9) que, a diferencia de la búsqueda hacia adelante, presenta una mayor dificultad de implementación ya en la práctica.

Mientras que la búsqueda hacia adelante parte del estado inicial del problema, la búsqueda hacia atrás parte del estado objetivo que contiene el predicado objetivo del problema y trata de alcanzar el estado inicial. La principal ventaja de la búsqueda hacia atrás es que el plan final solo contendrá acciones relevantes eliminando las acciones innecesarias de la búsqueda hacia adelante.

**Figura 9***Pseudocódigo Backward*

```

Busqueda(O, s, g)
  submetas <- g
  plan <- []
  Bucle
    Si s satisface g
      devolver plan
    Sino
      Obtener acciones que se pueden aplicar (acciones relevantes para g)
      Si accionesPosibles esta vacia
        devolver fallo
      Sino
        Seleccionar una accion
        aplicar accion sobre g
        plan <- accion \cup plan

```

## Heurísticas de búsqueda

Para que un algoritmo de búsqueda ya sea hacia adelante o hacia atrás pueda llegar a ser eficiente, necesita de una estrategia que se conoce como función heurística. Las funciones heurísticas calculan el costo que tendría el pasar de un estado actual al estado objetivo, o del estado objetivo al estado inicial en el caso de la búsqueda hacia atrás. El costo del camino se calcula en tomando en cuenta el costo de cada acción, sin embargo, para problemas de STRIPS el costo de todas las acciones es igual a 1. Para ello, se aplica una función de evaluación que toma el nodo con el menor valor para pasar del estado actual a otro estado que satisfaga el objetivo de la tarea.

Se tienen distintos tipos de búsqueda que utilizan funciones heurísticas como:

**Búsqueda Voraz:** Se aplica en el nodo más próximo al objetivo. La búsqueda voraz tiene algunos problemas, por ejemplo, no es completa ni óptima.

**A\*:** Trabaja evitando nodos con un peso demasiado alto. Por el contrario de la búsqueda voraz es óptima. Requiere que la función heurística sea admisible.

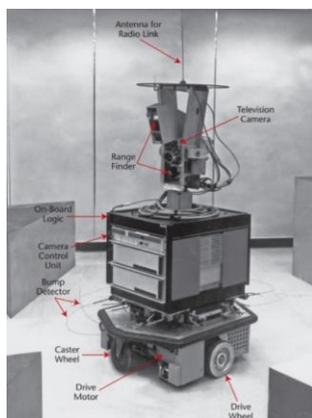
## *Stanford Research Institute Problem Solver (STRIPS)*

### Shakey y STRIPS

En los años de 1966-1972, el SRI (Stanford Research Institute) desarrolló un robot móvil apodado Shakey (Figura10). Shakey podía realizar tareas que requerían planificación, búsqueda de rutas y la reorganización de objetos simples. El robot tenía dos metros de altura y estaba provisto de una cabeza formada por una cámara de televisión y otra telemétrica para percibir el entorno.

**Figura 10**

Shakey con sus principales componentes.



*Nota. Tomado de Sánchez (2017).*

Shakey lograba la planificación de sus acciones gracias a uno de los primeros sistemas de resolución de problemas de robot denominado STRIPS según Nilsson (1987). En concreto STRIPS es un planificador automático desarrollado por Nils Nilsson y Richard Fikes para Shakey y que logra su fin haciendo uso de una pila de objetivos. STRIPS se encuentra dentro del grupo de planificadores en el espacio de estados y de búsqueda hacia atrás. STRIPS trabaja reduciendo el espacio de búsqueda considerablemente partiendo de los objetivos.

### **Funcionamiento de STRIPS**

La forma en la que trabaja STRIPS es enfocándose en las precondiciones del último operador añadido. Como se ha mencionado, STRIPS utiliza una pila y siempre trata de resolver los problemas en la parte superior de la pila. Esta pila contiene inicialmente sólo los objetivos más importantes para un determinado problema. Cuando la pila de objetivos está completamente vacía, el problema se ha solucionado con STRIPS.

Cuando STRIPS intenta resolver un problema (objetivo) en la parte superior de la pila, existen tres posibilidades:

**Objetivo Simple y sin resolver:** STRIPS intenta hacer coincidir el predicado con el estado actual del mundo. Si se encuentra una coincidencia con el estado del mundo, la sustitución obtenida se aplica a todos los objetivos por debajo de ese objetivo en la pila. Si el objetivo se encuentra en el estado de mundo, se elimina de la pila y se avanza al siguiente objetivo. Si no coinciden, buscamos una acción cuya lista de efectos positivos incluya el objetivo simple que estamos tratando de resolver, y añadimos esa acción a la pila de objetivos, sustituyendo el objetivo simple y añadiendo las precondiciones de esa acción como un nuevo objetivo.

**Objetivos compuestos:** Cuando STRIPS encuentra una meta compuesta, es decir, una que contiene más de un predicado, intenta dividirla en submetas que se añaden a la pila por encima de la meta compuesta para que STRIPS pueda resolverlas. Cuando todos los subobjetivos del objetivo compuesto se satisfacen en el estado actual del mundo, los comprueba de nuevo para evitar la comprobación de una acción que invalide un subobjetivo previamente satisfecho. Se trata de una herramienta muy sencilla para comprobar los subobjetivos y evitar el problema de que una acción anule un predicado previamente satisfecho.

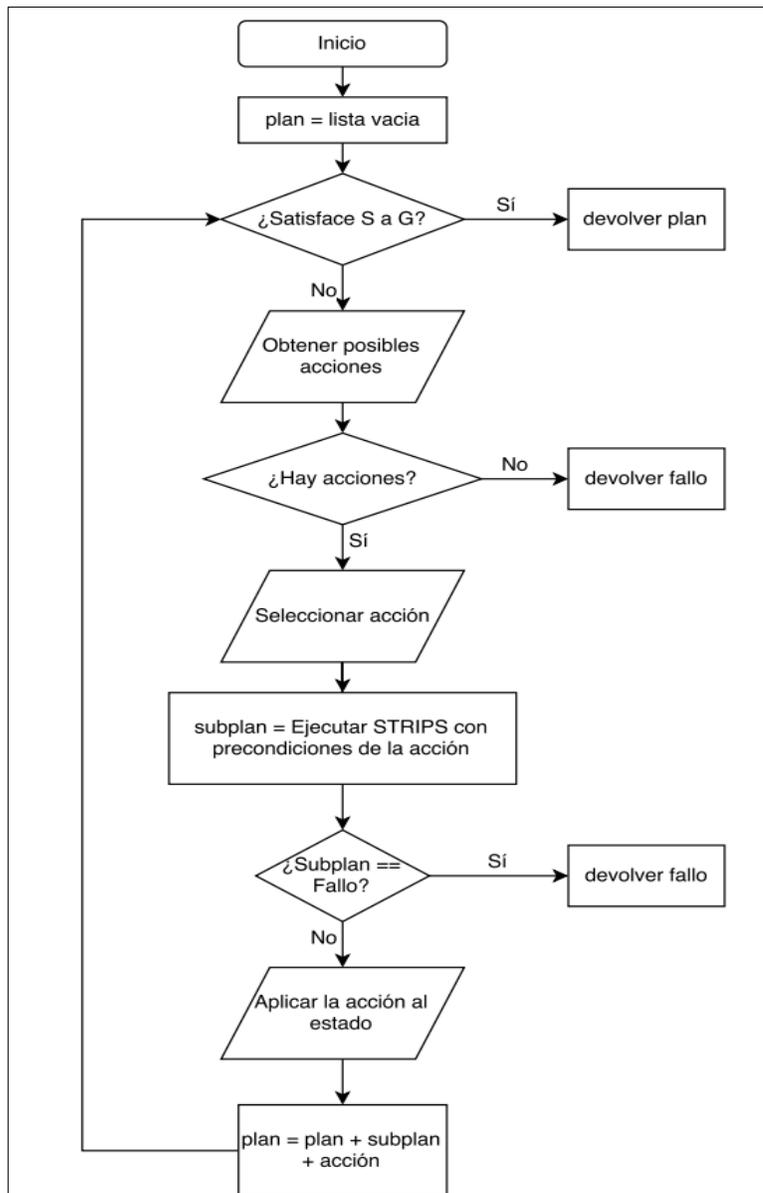
**Acciones:** Cuando STRIPS encuentra una acción en la parte superior de la pila, significa que las precondiciones de esa acción, que se consideraron como objetivo, se ha satisfecho y está en el

estado actual del mundo. Esto se traduce en que: la acción se elimina de la parte superior de la pila, y que la acción se debe aplicar al estado del mundo.

Por tanto, STRIPS funciona de forma recursiva, ejecutando un proceso recursivo cuando hay que añadir una acción. Este proceso recursivo se realiza pasando las precondiciones de esta acción a STRIPS como un nuevo objetivo.

**Figura 11**

*Flujograma de un planificador STRIPS.*



**Nota.** Tomado de Vassos (2012)

Cuando la pila que se generó ya no contiene más objetivos insatisfechos, es decir, cuando se ha resuelto todo el conjunto de objetivos del problema, la solución es una serie ordenada de acciones que forman un plan para pasar del estado inicial del problema a otro estado en el que se satisfacen todos los predicados del conjunto de objetivos.

### **Estrategias de control de STRIPS**

Según Nilsson (1987) para que STRIPS trabaje de una manera adecuada se requiere del uso de estrategias de control:

Para la resolución de un objetivo complejo, se lo debe dividir en objetivos simples que lo conforman. El orden de estos objetivos debe ser controlado, ya sea dando prioridad a los objetivos que ya han sido satisfechos o seleccionando al azar el orden en que se insertan en la pila.

Es posible que se encuentren variables con más de un unificador, y para lidiar con esto, se puede crear un nodo sucesor para cada unificador.

Un caso que se podría dar es que para un objetivo que no sea complejo se tengan varias acciones que cumplan con las condiciones. Por lo tanto, es necesario seleccionar la prioridad de las diferentes acciones. Como antes, se crea un nuevo nodo sucesor para cada acción diferente.

### **Problemas con STRIPS**

Dado que STRIPS es un planificador básico, existen problemas marcados como son:

Algunas soluciones pueden no ser óptimas, nos podemos encontrar con el hecho de que se produce una anomalía de Sussman en una solución. Esta anomalía se produce cuando se intenta resolver un subobjetivo después de que ya se haya alcanzado otro y la acción que se ejecuta deshace el subobjetivo anterior. Esto se traduce en que STRIPS llega a solucionar un problema, pero puede que generando un plan aún más largo del que en verdad se requiere.

El segundo problema que se puede ocasionar con STRIPS es que puede no encontrar la solución a un problema, como un ejemplo, si nuestro objetivo requiere intercambiar los datos de dos variables no es posible encontrar una solución con STRIPS por varias limitantes de su lenguaje y forma de buscar la solución como que los estados se representan simplemente como una conjunción de literales positivos, los objetivos son conjunciones (no se permite la disyunción) y que STRIPS no soporta la igualdad.

## **Visión Artificial en procesos Industriales**

Como lo hemos visto vivimos en una época en la que la automatización y la robótica industrial son tecnologías cada vez más necesarias en la industria y las empresas. Se ha desarrollado una amplia gama de herramientas para mejorar y apoyar la producción industrial y las líneas de montaje. La razón es que, sin duda, el proceso de producción tiene que cumplir normas precisas y estrictas.

La visión artificial es un conjunto de herramientas y técnicas para adquirir, procesar y analizar imágenes del mundo real para que puedan ser procesadas por los ordenadores como lo indica Infaimon (2017). Esto permite automatizar una amplia gama de tareas, dando a las máquinas la información que necesitan para tomar las decisiones correctas para cada tarea.

Uno de los sectores que más se ha beneficiado de estas tecnologías es el de la visión artificial industrial. El uso de la visión artificial en la industria permite automatizar los procesos de producción y calidad, lo que permite mejorar los resultados de producción.

Los sistemas de visión por ordenador (CVS) pretenden imitar la visión humana basándose en el análisis de datos multivariados y el reconocimiento de imágenes, utilizando sistemas de visión por ordenador para extraer características específicas como los atributos de calidad y los colores.

La visión artificial se utiliza cada vez más en los procesos industriales para la supervisión y el control, como la medición, el corte, la soldadura, la inspección de soldaduras, el taladrado y el montaje de componentes mecánicos como lo menciona Loja (2015). Un mundo tridimensional potencialmente dinámico se describe mediante las propiedades geométricas y materiales del mundo bidimensional, desde el mundo real hasta un mundo abstracto formado por vectores y matrices, por lo que se aplican técnicas propias de este mundo abstracto.

### ***Espacios de Colores***

Un espacio de color es un sistema de interpretación de los colores, una forma particular de organizar los colores de una imagen o un vídeo. Depende del modelo de color y del dispositivo físico que pueda captar los colores, como la señal electromagnética analógica de un televisor en color o la señal electromagnética digital de una cámara de vídeo.

Las características que distinguen un color de otro son las siguientes La luminancia es la intensidad de la luz emitida o reflejada por un color y está relacionada con la cromaticidad de ese

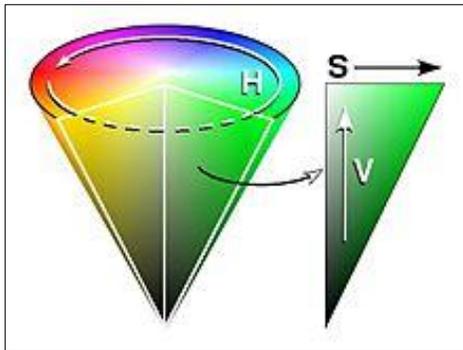
color. El matiz es el sistema de colores y las tonalidades que tiene este sistema. Y el tono es una propiedad que se relaciona con el color tal y como lo percibe el observador.

### **Modelo de color HSV**

El modelo de color HSV, como lo mencionan Cuevas, Zaldívar, & Pérez (2010), se caracteriza por la separación de la información de color y luminancia. La información del color está representada por componentes (tono H, saturación S, valor V), este modelo también se denomina modelo. Este modelo se representa como un cono invertido (Figura 12), en donde el eje de simetría representa la componente de valor V que representa la iluminación, la distancia radial o polar medida hasta cualquier distancia radial con el eje V representa la componente de saturación S, y la variable angular con el mismo eje V mide el ángulo en radianes y representa la componente de color H. El color "negro" corresponde al vértice de un cono o pirámide, mientras que el color "blanco" está en el eje V, opuesto al vértice en el centro de la base de la pirámide o cono invertido.

**Figura 12**

*Representación del modelo HSV*



**Nota. Tomado de Cuevas, Zaldívar, & Pérez (2010)**

### **Segmentación**

Existen varios métodos de segmentación, como el filtrado, el enmascaramiento, el suavizado y el refinado, etc. Según Pajares & De la Cruz (2007), la segmentación es un proceso general que debe realizarse para extraer la información necesaria para la identificación y el reconocimiento de las imágenes digitales.

Aspectos como la discontinuidad y la similitud no deben faltar en todo proceso de segmentación como indican Fu & Mui (1981), la discontinuidad se tiene en cuenta al segmentar los bordes y la similitud al segmentar las regiones de una imagen.

## Capítulo 3

### Diseño mecatrónico

El capítulo detalla el proceso de diseño de la celda robótica con un manipulador Prismático – Rotacional – Rotacional (PRR) y del planificador STRIPS, para lo cual se consideran las necesidades del usuario y los requerimientos de funcionamiento, planteados en la matriz de calidad. Posteriormente se determinan los subsistemas con respecto a las características que debe tener la celda robótica para un diseño establecido, tomando en cuenta las ventajas y desventajas de cada alternativa. Una vez establecido el diseño final se realiza un boceto más detallado, con los respectivos cálculos y dimensionamiento de cada componente, así mismo, se diseña el planificador, la programación de nodos y la visión por computador que requiere la planificación del robot para que todo el sistema trabaje de forma conjunta.

#### Matriz de Calidad (QFD)

La QFD (por sus siglas en inglés Quality Function Deployment), González, (2014) menciona que el despliegue de las Funciones de la Calidad es un método que conforma todas las demandas y expectativas que tienen los clientes y las convierte en características técnicas y metas operativas que al final son satisfactorias para los mismos.

La matriz se implementa en el proyecto pues para ser competitivo, en cualquier ámbito, es necesario el desarrollo de nuevos productos en cortos períodos de tiempo, de bajo costo, pero con una alta calidad, considerando que los requerimientos y las necesidades de los clientes cambiarán constantemente.

Por lo que, la QFD es la voz sistemática del cliente directamente pasando por todas las fases que conlleva la construcción de la casa de la calidad, las cuales permiten evaluar de manera concreta los requerimientos del cliente, obteniendo un resultado objetivo y sistemático respecto a la competencia.

#### Definición de necesidades

En esta parte, se definen todas las características que debe tener el producto de acuerdo con el cliente, especificaciones que se requieren para el diseño final (Tabla 1).

**Tabla 1***Necesidades del usuario.*

<b>N</b>	<b>¿Qué? (Necesidades)</b>	<b>Segmento (Realiza el grupo)</b>
1	Resolución de problemas rápidamente	Resolución/Implementación en tecnología que ayude a resolver ciertos problemas
2	Flexibilidad con las condiciones iniciales del sistema	Planificación/Innovar en la planificación manteniendo la logística adecuada y correcta
3	Seguridad dentro de la celda de trabajo	Seguridad/Cumplimiento de todas las normas de seguridad
4	Interfaz amigable	Observación/Simulación de estados en una interfaz y físicamente
5	Visión Artificial	Codificación/Implementación por colores
6	Fácil operación	Operación/Diseño sencillo
7	Tamaño reducido	Tamaño/Diseño amigable
8	Alta repetibilidad	Repetibilidad/Diseño amigable
9	Fácil mantenimiento	Mantenimiento/Conservación sencilla de las piezas del robot

**Definición de Requerimientos**

Se refiere a todas las características de las necesidades anteriores pero establecidas con una metodología para cada una y las metas operativas que ayudarán al proceso de diseño de la celda (Tabla 2).

**Tabla 2***Requerimientos y metas operativas*

	<b>REQUERIMIENTOS</b>	<b>METAS OPERATIVAS</b>
<b>METODOLOGÍA</b>	Inversión en equipamiento de alta tecnología	Fabricación de un robot
	Reingeniería en procesos de planificación	Nuevas líneas de planificación
	Inversión en protocolos de seguridad	Garantizar la seguridad en la celda de trabajo
	Interacción con el usuario	Conocimiento previo de la meta
	Visión Artificial	Implementación por colores
	Capacitación al personal	Reducción de complejidad y tiempos de operación
	Ensamblaje con manufactura aditiva	Reducción de costos de fabricación
	Fácil diseño	Diseño que se puede repetir fácilmente
	Programa de mantenimiento de equipos	Certificación de calidad del prototipo



Posición (Donde, 1 es poco importante; y 3 muy importante)	3	1				2			
--	---	---	--	--	--	---	--	--	--

En la matriz de Calidad, se registran las necesidades y requerimientos establecidos por el usuario anteriormente, a cada una se le asigna una ponderación para clasificar y priorizar cada una de ellas, dándole mayor importancia a las características esenciales y tomando en cuenta siempre las necesidades del usuario.

### Factibilidad y Planeación

La factibilidad que va a tener este proyecto, se deriva directamente de los resultados de la matriz QFD, (Tabla 4), las características y la ponderación de cada una de ellas de acuerdo a su desempeño

**Tabla 4**

*Resultados de la Matriz QFD*

Requisitos del Cliente	Muestra	Calif.	© Relación Fuerte (9) Ω Correlación (3) Δ Relación Débil (1) ☐ No existe relación (0) - Correlación Negativa (-1)									
			Inversión en equipamiento de alta tecnología	Reingeniería en procesos de planificación	Inversión en protocolos de seguridad	Interacción con el usuario	Visión Artificial	Capacitación al personal	Ensamblaje con manufactura aditiva	Fácil diseño	Programa de mantenimiento de equipos	
1 Resolución de problemas rápidamente	9	9	9	3	0	1	3	3	0	0	0	
2 Flexibilidad con las condiciones iniciales del sistema	8	8	0	9	0	0	3	1	0	0	0	
3 Seguridad dentro de la celda de trabajo	3	7	1	0	9	0	0	3	3	3	3	
4 Interfaz amigable	5	6	3	0	-1	9	3	3	3	3	1	
5 Cilindros de colores	4	5	3	1	0	1	9	0	0	0	0	
6 Fácil operación	6	4	1	0	1	3	3	9	0	0	0	
7 Tamaño reducido	7	3	3	0	1	3	1	3	9	3	1	
8 Alta repetibilidad	2	2	3	0	0	0	1	3	3	9	1	
9 Fácil mantenimiento	1	1	3	0	3	0	0	1	3	3	9	
<b>Cálculo</b>			147	103	38	97	129	141	96	66	32	
<b>Importancia relativa (%)</b>			17.9	12.6	4.7	11.9	15.8	17.3	11.8	8.1	3.9	
<b>Posición (Donde, 1 es poco importante; y 3 muy importante)</b>			3	1				2				

Una vez analizados los resultados obtenidos en la matriz QFD se realiza la planeación del proyecto, la cual se despliega en módulos y estos a su vez, en subsistemas, (Tabla 5), obteniendo una estructura funcional del proyecto para satisfacer todas las necesidades señaladas anteriormente.

**Tabla 5**

*Módulos y subsistemas del proyecto*

<b>N</b>	<b>Módulo</b>	<b>Subsistema</b>	<b>Función</b>	<b>Hitos</b>
<b>1</b>	Manipulador Robótico PRR	Cinemática y Dinámica	Posicionamiento y movimiento del robot	1
		Estructural	Geometría y diseño del robot	1,7,8,9
<b>2</b>	Espacio de trabajo	de Celda de Trabajo	Área de trabajo para la clasificación de las piezas	3,7,8,9
<b>3</b>	Potencia y Control	Potencia del Robot	Sensores, fuente de poder, actuadores	6,8
		Control del Robot	Microcontroladores para el control de los movimientos del robot	5,6
<b>4</b>	Interacción	Visión Artificial	Detección de los cubos por medio de colores	5,6
		Planificación	Predicción de posición final de piezas	2
		Interfaz Gráfica	Interacción con el usuario	4,6

### **Módulo 1. Manipulador Robótico PRR**

#### ***Subsistema 1. Cinemática y Dinámica***

El estudio de la cinemática directa e inversa se usan para obtener las ecuaciones que determinan la posición del efector final, en este caso se analizará la cinemática directa mediante la matriz de parámetros Denavit Hartenberg, y con respecto a la cinemática inversa, se analiza mediante el método Geométrico. La Dinámica del Robot es parte fundamental para encontrar las fuerzas y torques a las que se somete el mismo cuando realiza el posicionamiento.

#### **Cinemática Directa**

El robot que se usará en el desarrollo del proyecto es un manipulador PRR (Prismático Rotacional Rotacional), (Figura 13), la cual define todos los parámetros que serán de utilidad para determinar las relaciones que determinan el posicionamiento respecto a  $x,y,z$ , en función de las variables articulares del robot.

Figura 13

Boceto del esquema del Robot PRR.

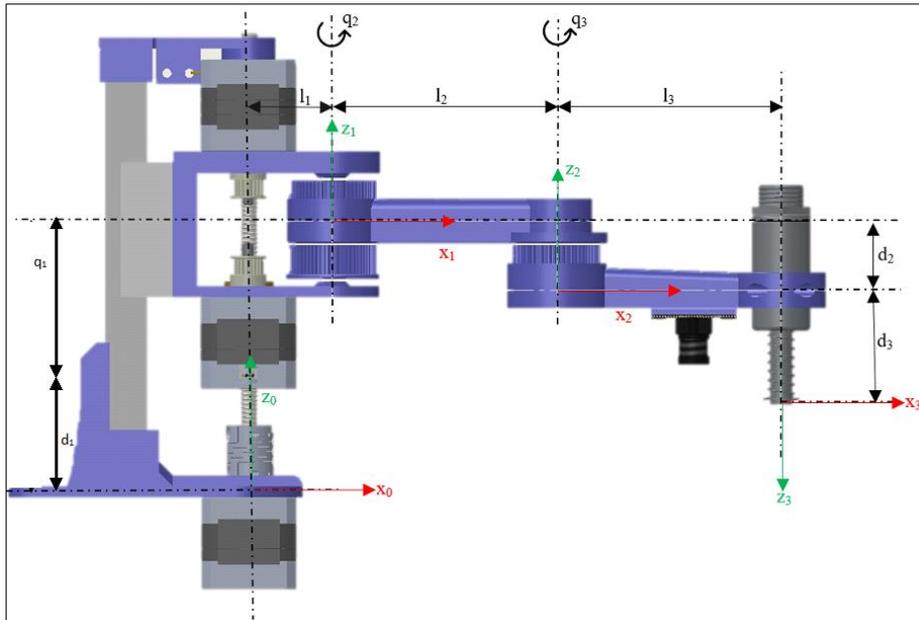


Tabla 6

Parámetros de D-H

$i$	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	0	$d_1 + q_1$	$l_1$	0
2	$q_2$	$-d_2$	$l_2$	0
3	$q_3$	$-d_3$	$l_3$	0

Con los parámetros D-H, (Tabla 6), se obtienen las matrices de transformación homogénea de cada articulación.

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

$$A_1^0(q_1) = \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 + q_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$A_2^1(q_2) = \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 & l_2 \cos q_2 \\ \sin q_2 & \cos q_2 & 0 & l_2 \sin q_2 \\ 0 & 0 & 1 & -d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$A_3^2(q_3) = \begin{bmatrix} \cos q_3 & -\sin q_3 & 0 & l_3 \cos q_3 \\ \sin q_3 & \cos q_3 & 0 & l_3 \sin q_3 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$T_3^0 = A_1(q_1) * A_2(q_2) * A_3(q_3) \quad (5)$$

$$T_3^0 = \begin{bmatrix} \cos(q_2 + q_3) & -\sin(q_2 + q_3) & 0 & l_1 + l_2 C_2 + l_3 \cos(q_2 + q_3) \\ \sin(q_2 + q_3) & \cos(q_2 + q_3) & 0 & l_2 S_2 + l_3 \sin(q_2 + q_3) \\ 0 & 0 & 1 & d_1 - d_2 - d_3 + q_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Las coordenadas x, y, z se definen a continuación.

$$x = l_1 + l_2 C_2 + l_3 \cos(q_2 + q_3) \quad (7)$$

$$y = l_2 S_2 + l_3 \sin(q_2 + q_3) \quad (8)$$

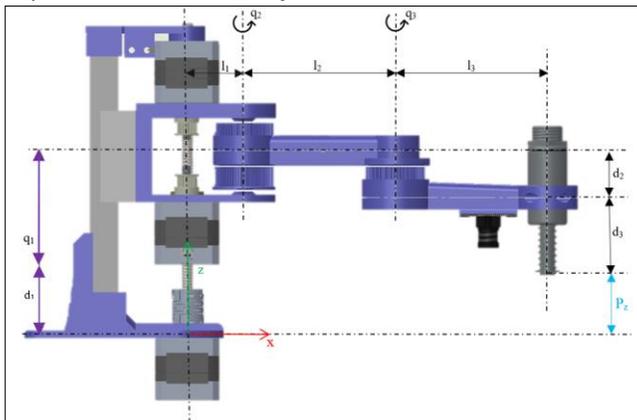
$$z = d_1 - d_2 - d_3 + q_1 \quad (9)$$

### Cinemática Inversa

En la cinemática inversa el objetivo es encontrar los valores que deben tener las coordenadas articulares del robot para que el efector final tome una posición y orientación determinada en el espacio (x,y,z), para este caso se utilizará el método geométrico, el cual usa relaciones geométricas y trigonométricas mediante la resolución de triángulos.

**Figura 14**

*Esquema del Robot en el eje X-Z*

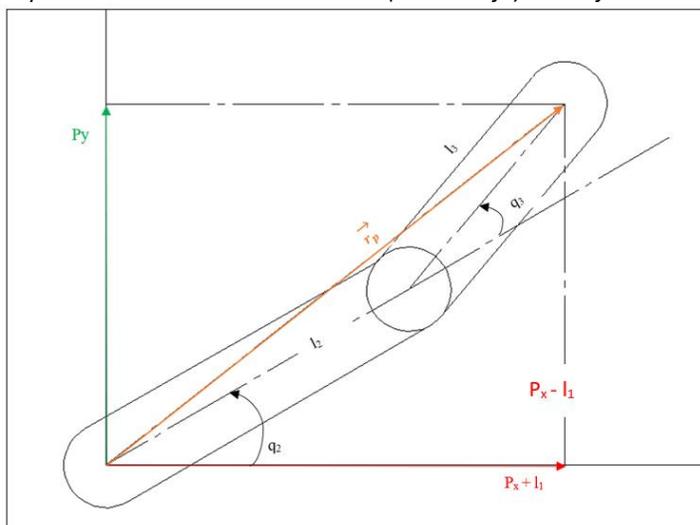


Para encontrar la posición del efector final en el eje X-Z, (Figura 14), en la que se toman en cuenta las distancias necesarias para llegar a la posición indicada.

$$q_1 = d_2 + d_3 - d_1 + P_z \quad (10)$$

**Figura 15**

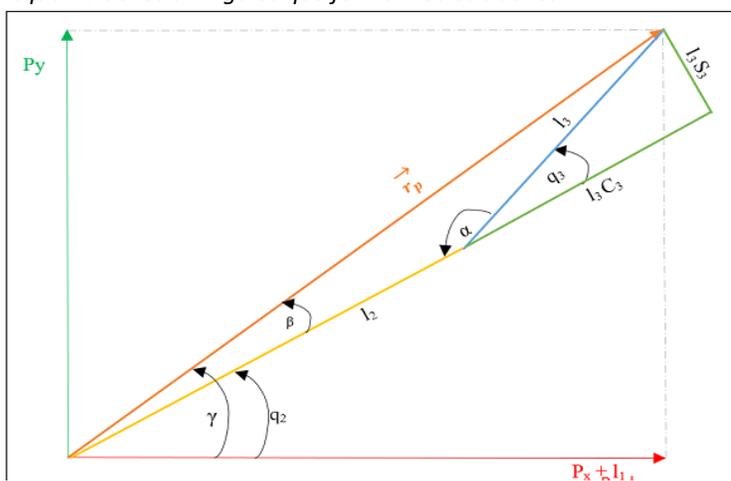
*Esquema de los eslabones del robot (codo abajo) en el eje X-Y*



Para el desarrollo del método geométrico tomamos en cuenta a los eslabones del robot codo abajo, para que no exista un conflicto con la posición y orientación, (Figura 16), los eslabones forman triángulos y los ángulos son más fáciles de visualizar.

**Figura 16**

*Esquema de los triángulos que forman los eslabones*



Se muestran, (Figura 16), más definidos los triángulos que forman los eslabones se puede realizar el método geométrico, usando la Ley de Cosenos.

$$\overline{r_p^2} = (P_x - l_1)^2 + P_y^2 \quad (11)$$

$$\alpha + q_3 = 180^\circ = \pi \quad (12)$$

$$\alpha = \pi - q_3 \quad (13)$$

$$\cos(\alpha) = \cos(\pi - q_3) = -\cos(q_3) \quad (14)$$

Ley de cosenos

$$\overline{r_p^2} = l_2^2 + l_3^2 - 2l_2l_3 \cos(\alpha) = l_2^2 + l_3^2 + 2l_2l_3 \cos(q_3) \quad (15)$$

$$q_3 = \cos^{-1} \left( \frac{(P_x - l_1)^2 + P_y^2 - l_2^2 - l_3^2}{2l_2l_3} \right) \quad (16)$$

$$\gamma = \beta + q_2 \quad (17)$$

$$q_2 = \gamma - \beta \quad (18)$$

$$\beta = \cos^{-1} \left( \frac{(P_x - l_1)^2 + P_y^2 + l_2^2 - l_3^2}{2l_2 \sqrt{(P_x - l_1)^2 + P_y^2}} \right) \quad (19)$$

$$\gamma = \tan^{-1} \left( \frac{P_y}{P_x - l_1} \right) \quad (20)$$

$$q_2 = \tan^{-1} \left( \frac{P_y}{P_x - l_1} \right) - \cos^{-1} \left( \frac{(P_x - l_1)^2 + P_y^2 + l_2^2 - l_3^2}{2l_2 \sqrt{(P_x - l_1)^2 + P_y^2}} \right) \quad (21)$$

De esta manera los valores de las variables articulares del robot son las que se muestran a continuación.

$$q_1 = d_2 + d_3 - d_1 + P_z \quad (22)$$

$$q_2 = \tan^{-1} \left( \frac{P_y}{P_x - l_1} \right) - \cos^{-1} \left( \frac{(P_x - l_1)^2 + P_y^2 + l_2^2 - l_3^2}{2l_2 \sqrt{(P_x - l_1)^2 + P_y^2}} \right) \quad (23)$$

$$q_3 = \cos^{-1} \left( \frac{(P_x - l_1)^2 + P_y^2 - l_2^2 - l_3^2}{2l_2l_3} \right) \quad (24)$$

### Dinámica

Para el modelo dinámico del robot, que determinará el torque necesario que necesitan los motores se utiliza la formulación de Euler-Lagrange, mediante el balance de energías, cinética y potencial que almacena el sistema.

A partir de las matrices de transformación homogénea se construye el Jacobiano que es necesario para encontrar las velocidades lineales y angulares que rige el sistema.

$$J = [J_1, \dots, J_n] \quad (25)$$

Donde n, es la cantidad de juntas

Si la junta es prismática:

$$J_i = \begin{bmatrix} Z_{i-1} \\ \vec{0} \end{bmatrix} \text{ Donde } \vec{0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (26)$$

Si la junta es rotacional:

$$J_i = \begin{bmatrix} Z_{i-1} \times (O_n - O_{i-1}) \\ Z_{i-1} \end{bmatrix} \quad (27)$$

De la matriz de transformación homogénea, se obtienen los demás factores, de esta manera

$$T_0^i = \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (28)$$

$Z_{i-1}$      $O_i$

De esta manera,

$$\vec{Z}_0 = \vec{Z}_1 = \vec{Z}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (29)$$

$$\vec{O}_1 = \begin{bmatrix} l_1 \\ 0 \\ d_1 + q_1 \end{bmatrix} \quad (30)$$

$$\vec{O}_2 = \begin{bmatrix} l_1 + l_2 \cos(q_2) \\ l_2 \sin(q_2) \\ d_1 - d_2 + q_1 \end{bmatrix} \quad (31)$$

$$\vec{O}_2 = \begin{bmatrix} l_1 + l_2 \cos(q_2) + l_3 \cos(q_2 + q_3) \\ l_2 \sin(q_2) + l_3 \sin(q_2 + q_3) \\ d_1 - d_2 - d_3 + q_1 \end{bmatrix} \quad (32)$$

Como el manipulador es un robot PRR, se obtiene un Jacobiano de la siguiente manera:

$$J = \begin{bmatrix} 0 & -l_2 \sin(q_2) - l_3 \sin(q_2 + q_3) & -l_3 \sin(q_2 + q_3) \\ 0 & l_2 \cos(q_2) + l_3 \cos(q_2 + q_3) & l_3 \cos(q_2 + q_3) \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{matrix} J_v \\ \\ \\ \\ J_\omega \end{matrix} \quad (33)$$

Las singularidades son valores limitantes para el movimiento de cada articulación, se realiza mediante el determinante del Jacobiano, dando como singularidades para  $q_1$  y  $q_3$ :

$$q_1 > 1; \quad q_3 \neq 0, \pi, 2\pi, 3\pi, 4\pi, \dots \quad (34)$$

Velocidades lineales y angulares:

$$v_1 = J_v(q)\dot{q} \quad (35)$$

$$\omega_2 = R_2^T J_{\omega_2}(q)\dot{q} \quad (36)$$

$$\omega_3 = R_3^T J_{\omega_3}(q)\dot{q} \quad (37)$$

La matriz de rotación se obtiene de cada matriz de transformación homogénea de cada articulación:

$$T_0^i = \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (38)$$

$$R_0^2 = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 \\ \sin(q_2) & \cos(q_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (39)$$

$$R_0^3 = \begin{bmatrix} \cos(q_2 + q_3) & -\sin(q_2 + q_3) & 0 \\ \sin(q_2 + q_3) & \cos(q_2 + q_3) & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (40)$$

$$\omega_2 = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_2 + \dot{q}_3 \end{bmatrix} \quad (41)$$

$$\omega_2 = \begin{bmatrix} 0 \\ 0 \\ -\dot{q}_2 - \dot{q}_3 \end{bmatrix} \quad (42)$$

$$v_1 = \begin{bmatrix} l_3 \dot{q}_3 \sin(q_2 + q_3) + \dot{q}_2 (-l_2 \sin(q_2) - l_3 \sin(q_2 + q_3)) \\ l_3 \dot{q}_3 \cos(q_2 + q_3) + \dot{q}_2 (l_2 \cos(q_2) + l_3 \cos(q_2 + q_3)) \\ \dot{q}_1 \end{bmatrix} \quad (43)$$

La velocidad lineal  $v_1$  se debe multiplicar por el paso del tornillo sin fin de 2mm al que irá acoplado el motor.

La matriz de inercias de cada eslabón se toma en cuenta con respecto al centro de la base de cada uno, por lo que es fácilmente encontrar la matriz del software de CAD, en el que fue diseñado el sistema.

### Energía Cinética

- Movimiento lineal

$$K_1 = \frac{m_1}{2} v_1^T v_1 \quad (44)$$

- Movimiento angular

$$K_2 = \frac{1}{2} \omega_2^T I_{233} \omega_2 \quad (45)$$

$$K_3 = \frac{1}{2} \omega_3^T I_{333} \omega_3 \quad (46)$$

### Energía Potencial

$$V_1 = (m_2 + m_3) \times g \times q_1 \quad (47)$$

$$V_2 = m_3 \times g \times (q_1 - d_2) \quad (48)$$

### Ecuación Euler Lagrange

$$L = K - V = (K_1 + K_2 + K_3) - (V_1 + V_2) \quad (49)$$

Reemplazando los datos:

$$L = \frac{l_{233} m_2 \dot{q}_2^2}{2} + l_{233} m_2 \dot{q}_2 \dot{q}_3 + \frac{l_{233} m_2 \dot{q}_3^2}{2} + \frac{l_{333} m_3 \dot{q}_2^2}{2} + l_{333} m_3 \dot{q}_2 \dot{q}_3 + \frac{l_{333} m_3 \dot{q}_3^2}{2} + d_2 g m_3 - g m_2 q_1 - 2g m_3 q_1 + \frac{l_2^2 m_1 \dot{q}_2^2}{2} + l_2 l_3 m_1 \dot{q}_2^2 \cos(q_3) + l_2 l_3 m_1 \dot{q}_2 \dot{q}_3 \cos(q_3) + \frac{l_3^2 m_1 \dot{q}_2^2}{2} + l_3^2 m_1 \dot{q}_2 \dot{q}_3 + \frac{l_3^2 m_1 \dot{q}_3^2}{2} + \frac{m_1 \dot{q}_1^2}{2} \quad (50)$$

El análisis de los datos se realizó en un software matemático, Matlab, en el que solo ingresamos los datos y obtenemos el valor de los torques necesarios para los motores.

$$m_1 = 0.852 \text{ kg} \quad m_2 = 0.075 \text{ kg} \quad m_3 = 0.127 \text{ kg}$$

$$l_1 = 0.0375 \text{ m} \quad l_2 = 0.1 \text{ m} \quad l_3 = 0.1 \text{ m}$$

$$d_1 = 0.057 \text{ m} \quad d_2 = 0.032 \text{ m} \quad d_3 = 0.011 \text{ m}$$

$$I_{233} = 0.000341 \text{ kgm}^2 \quad I_{333} = 0.000876 \text{ kgm}^2$$

El valor de cada torque se toma de la resolución de la ecuación de Euler-Lagrange:

$$\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{q}} \right] - \frac{\partial L}{\partial q} = u \quad (51)$$

Dentro del software se iteran valores de cada ángulo ( $q$ ), en las posiciones más críticas que puede tener cada eslabón, obteniéndose un valor de torque máximo de cada motor.

$$T_2 = 0.14 \text{ Nm} \quad (52)$$

$$T_3 = 0.11 \text{ Nm} \quad (53)$$

Para el motor 1, se realizó otro análisis, tomando en cuenta que va acoplado a un tornillo sin fin, el que tiene que soportar la masa de todo el sistema, según Alciro, en su proyecto *Plotter Router Fresadora CNC Alciro (2007)*, menciona que se debe tomar en cuenta la precarga de los rodamientos, la aceleración máxima del motor, el momento de inercia del motor, longitud del usillo, coeficiente de fricción entre patín y guía.

Para nuestro análisis la fuerza necesaria para mover la carga es  $F_c = 15 \text{ N}$  y ya que la resistencia a la rodadura es prácticamente despreciable con un coeficiente de fricción muy bajo y dado que el peso de la carga es también bajo de 1.5 kg, la fuerza es un par mediante un tornillo de paso 2mm y un rendimiento del 90%.

$$T_c = F_c \times \frac{P}{2 \times \pi \times 0.9} = 0.005305 \text{ Nm} \quad (54)$$

De acuerdo al fabricante, Alciro (2007), el par de arranque para el rodamiento que usa el tornillo con precarga es de 0.02059 Nm, como la tuerca del tornillo no tiene precarga, las pérdidas de par son despreciables, por lo que el par total del motor para mover la carga es:

$$T_1 = T_{rod} + T_c = 0.025895 \text{ Nm} \quad (55)$$

Debido a la aceleración se realiza un análisis de inercias, para un tornillo de 0.25 m de longitud y 0.008 m de diámetro, y dado que el material del tornillo es acero se usa la densidad de este de  $7.90 \times 10^3$ :

$$J_t = \frac{\pi\rho}{32} d^4 l = 7.94 \text{ gcm}^2 \quad (56)$$

La inercia que tiene la carga de 1.5 kg se determina a continuación:

$$J_c = m \left( \frac{P}{2 \times \pi} \right)^2 = 1.52 \text{ gcm}^2 \quad (57)$$

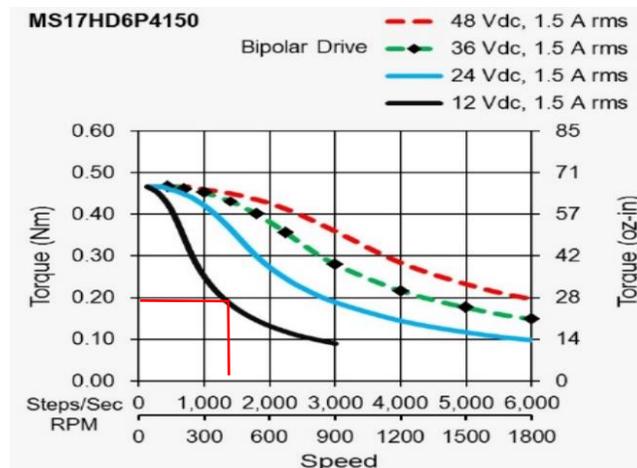
Finalmente, la inercia total del sistema tomando en cuenta la del motor de  $0.0000054 \text{ kgm}^2$ :

$$J = J_m + J_t + J_c = 63.46 \text{ gcm}^2 \quad (58)$$

Para el avance lineal de la carga a 420 rpm en 0.1 seg, el motor acelera a  $140\pi \frac{\text{rad}}{\text{s}^2}$  y mediante la siguiente gráfica, (Figura 17), se obtiene el par del motor T necesario para desplazar y acelerar el sistema:

**Figura 17**

*Curva Torque vs Velocidad del motor Nema 17 a 12 VDC.*



Nota. Tomado de MOONS' moving in better ways (2018)

$$T_1 = T + J \times \frac{\partial \omega}{\partial t} = 0.1828 \text{ Nm} \quad (59)$$

### **Subsistema 2. Estructural**

El subsistema estructural del manipulador robótico comprende el diseño físico de la geometría que tendrá cada parte, la elección del método y material de manufactura y los sistemas mecánicos de transmisión del movimiento en las articulaciones. Para el diseño de los eslabones del robot se toman en cuenta aspectos como las cargas que necesita soportar cada parte, la estética del modelo, la forma que requieren tener por los componentes que van a accionar las articulaciones y el efector final del manipulador.

### Método de Manufactura y material

Para el diseño mecánico de los eslabones del manipulador tomamos en cuenta que queremos lograr, en lo posible, un diseño modular por su facilidad de replicabilidad y mantenimiento y además el método de manufactura para obtener las partes. Dos opciones claras serían una manufactura por arranque de viruta en aluminio o algún metal y la impresión 3D como manufactura aditiva. En la Tabla 7 podemos observar aspectos que tomamos en cuenta entre las dos alternativas planteadas.

**Tabla 7**

*Características de los materiales para la manufactura de la estructura*

<b>Material</b>	<b>Manufactura Metálica</b>	<b>Manufactura Aditiva</b>
<b>COSTO</b>	ALTO	MODERADO
<b>PESO</b>	ALTO	BAJO
<b>MAQUINIBILIDAD</b>	ALTA	BAJA

Realizar la estructura física con manufactura aditiva, es decir en impresión 3D, en el que el objetivo es adherir material por capas, permite construir prototipos diseñados de una manera más rápida a bajo costo y para nuestro proyecto resulta más ventajoso para los eslabones y partes diseñadas con una geometría un tanto complicada del robot frente a una manufactura tradicional por arranque de viruta que resultaría en un costo más elevado y un peso considerablemente más alto.

Sin embargo, la manufactura aditiva se puede realizar en varios materiales y por diferentes métodos. Los principales métodos y materiales son por la parte de FDM (modelado por deposición fundida) en PLA (Polylactic Acid) o ABS (Acrilonitrilo Butadieno Estireno), mientras que por el método SLA (estereolitografía) en Resina, etc.

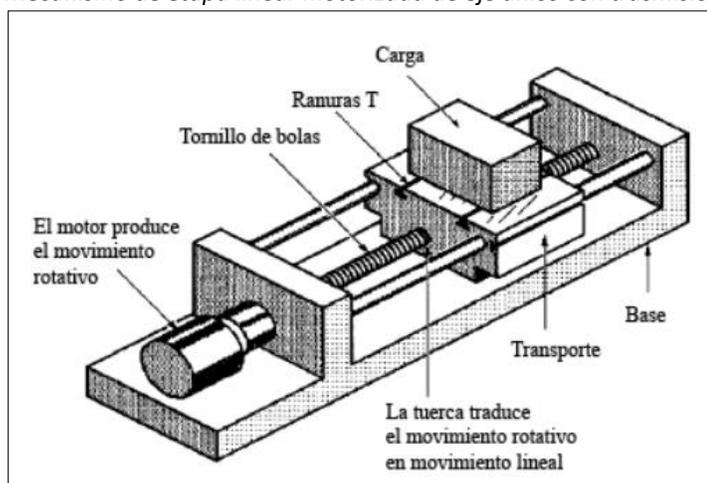
Para la elección del método de manufactura aditiva y el material se tomaron en cuenta aspectos como resistencia mecánica del material, el aspecto físico de la pieza, la factibilidad de impresión de la geometría de cada parte, los costos y principalmente el postproceso que requiere las piezas y la factibilidad nuestra con el método. Por todos estos aspectos el método de impresión que se eligió es el SLA en resina.

### Sistemas de transmisión de las articulaciones

El sistema de transmisión es aquel que se encarga de transmitir el movimiento desde el actuador en cada articulación del manipulador a los eslabones, para la articulación prismática se usa una transmisión por tornillo de posicionamiento, según Sapiensman: Oficios Técnicos (2020), se opta por este sistema por el costo accesible y por la holgura y juego que tiene, además de que cuenta con un acople flexible que permite transmitir la potencia desde un motor a un eje evitando problemas de defectos de alineamiento, a medida que el tornillo posicionador gira, la carga se traslada de manera axial, y ya que el objetivo es convertir el movimiento rotativo del motor en movimiento lineal para el levantamiento de la carga, (Figura 18), se opta por este tipo de transmisión para la junta prismática.

**Figura 18**

*Mecanismo de etapa linear motorizada de eje único con trasmisión de tornillo de bolas*



*Nota.* Tomado de Sapiensman: Oficios Técnicos (2020)

Para el sistema de transmisión de las articulaciones rotacionales se puede usar varias alternativas, los actuadores deben utilizar un sistema de transmisión que trasladen los movimientos de cada motor a cada articulación.

**Tabla 8**

*Ventajas y desventajas de cada alternativa para el sistema de transmisión.*

ALTERNATIVA	Tipo de Transmisión	Ventajas	Desventajas
Alternativa 1	Engranés	Par Alto No se resbalan	Necesita lubricación Holguras

		Costo bajo	
<b>Alternativa 2</b>	Correa dentada-polea	Distancia grande Sin ruido Costo bajo	Deslizamiento
<b>Alternativa 3</b>	Cadena	Distancia grande	Ruido Costo moderado

*Nota.* Tomado de E.T.S. de Ingenieros Industriales (2016)

### Selección de alternativa

Para la selección del sistema de transmisión adecuado se definen los siguientes criterios:

- Costo bajo
- Ruido bajo
- Transmisión regular con juego mínimo

**Tabla 9**

*Ponderaciones de cada criterio para el sistema de transmisión.*

CRITERIO	Costo	Ruido	Transmisión	$\Sigma+1$	%	Pond.
<b>Costo</b>	-	0	0.5	1.5	27.28	0.27
<b>Ruido</b>	0.5	-	0	1.5	27.28	0.27
<b>Transmisión</b>	1	0.5	-	2.5	45.45	0.45
<b>TOTAL</b>				5.5	100	1

**Tabla 10**

*Ponderaciones de cada alternativa respecto a cada criterio para el sistema de transmisión.*

Costo	Alternativa 1	Alternativa 2	Alternativa 3	$\Sigma+1$	Pond.
<b>Alternativa 1</b>	-	0.5	1	1.5	0.5
<b>Alternativa 2</b>	0.5	-	1	1.5	0.5
<b>Alternativa 3</b>	0	0	-	0	0
<b>Total</b>				3	1
Ruido	Alternativa 1	Alternativa 2	Alternativa 3	$\Sigma+1$	Pond.

<b>Alternativa 1</b>	-	0.3	1	1.3	0.43
<b>Alternativa 2</b>	0.7	-	1	1.7	0.57
<b>Alternativa 3</b>	0	0	-	0	0
<b>Total</b>				3	1
<b>Transmisión</b>	<b>Alternativa 1</b>	<b>Alternativa 2</b>	<b>Alternativa 3</b>	<b>Σ+1</b>	<b>Pond.</b>
<b>Alternativa 1</b>	-	0.3	0.5	1.8	0.3
<b>Alternativa 2</b>	0.7	-	0.5	2.2	0.37
<b>Alternativa 3</b>	0.5	0.5	-	2	0.33
<b>Total</b>				6	1

**Tabla 11**

*Selección de alternativa para el sistema de transmisión.*

	<b>Costo</b>	<b>Ruido</b>	<b>Transmisión</b>	<b>Σ</b>	<b>Pond.</b>
<b>Alternativa 1</b>	0.135	0.116	0.135	0.386	2
<b>Alternativa 2</b>	0.135	0.154	0.167	0.456	1
<b>Alternativa 3</b>	0	0	0.149	0.149	3

Como se determina, (Tabla 11), se escoge la alternativa 2. Se realiza el sistema de transmisión de movimiento en las articulaciones rotacionales mediante el uso de correas dentadas y poleas dentadas por las ventajas y facilidad que presentan (Tabla 8).

### **Efactor Final**

El efector final, de acuerdo a Rodas (1999), es un dispositivo que se coloca en la última articulación, necesario para realizar la función del objetivo a lograr con el robot, por lo tanto, pueden existir una variedad de efectores finales dependiendo de la función que deba cumplir el manipulador, entre pinzas y los que se denominan herramientas como taladros, sopletes, soldadoras, etc.

Las pinzas, también llamadas “grippers”, según Rodas (1999), se utilizan para agarrar y sostener objetos como botellas, materiales en brutos, cartones, etc., las pinzas pueden ser dispositivos de agarre mecánico, pero también se puede usar imanes o ventosas para la sujeción de objetos.

A continuación, se detallan las ventajas y desventajas de las alternativas para la sujeción de objetos.

**Tabla 12**

*Ventajas y desventajas de cada alternativa para el efector final.*

<b>ALTERNATIVA</b>	<b>TIPO DE EFECTOR</b>	<b>VENTAJAS</b>	<b>DESVENTAJAS</b>
<b>ALTERNATIVA 1</b>	Pinzas Mecánicas	Sujeción de las piezas de forma segura aplicando la fuerza necesaria	Costo alto Necesita dedos mecánicos Requiere mecanismo para agarre de piezas Los dedos deben estar unidos al mecanismo.
<b>ALTERNATIVA 2</b>	Ventosas	Caucho o plástico Liviana	Costo moderado Objetos planos, suaves y limpios Usa bomba de vacío
<b>ALTERNATIVA 3</b>	Magnética	Costo bajo Tolera variación en el tamaño de la pieza Manipula piezas con agujeros. No requiere orientación.	Solo en objetos metálicos Posible deslizamiento Eleva un objeto a la vez.

*Nota.* Tomado de Rodas (1999), Rey & Hernández (1991).

### **Selección de alternativa**

Para la selección del sistema de transmisión adecuado se definen los siguientes criterios:

- Costo bajo
- Peso bajo

- Tolerancia con el tamaño de la pieza

**Tabla 13**

*Ponderaciones de cada criterio para el efector final.*

CRITERIO	COSTO	PESO	TOLERANCIA	$\Sigma+1$	%	POND.
<b>COSTO</b>	-	1	0.5	2.5	41.67	0.42
<b>PESO</b>	1	-	0	2	33.33	0.33
<b>TOLERANCIA</b>	0.5	0	-	1.5	25	0.25
<b>TOTAL</b>				6	100	1

**Tabla 14**

*Ponderaciones de cada alternativa respecto a cada criterio para el efector final.*

Costo	Alternativa 1	Alternativa 2	Alternativa 3	$\Sigma+1$	Pond.
Alternativa 1	-	1	1	2.5	0.42
Alternativa 2	1	-	0.5	1.5	0.25
Alternativa 3	1	1	-	2	0.33
Total				6	1
Peso	Alternativa 1	Alternativa 2	Alternativa 3	$\Sigma+1$	Pond.
Alternativa 1	-	0	0.5	1.5	0.25
Alternativa 2	1	-	1	3	0.5
Alternativa 3	0.5	0	-	1.5	0.25
Total				6	1
Tolerancia	Alternativa 1	Alternativa 2	Alternativa 3	$\Sigma+1$	Pond.
Alternativa 1	-	0.5	0	1.5	0.25

Alternativa 2	0.5	-	0	1.5	0.25
Alternativa 3	1	1	-	3	0.5
Total				6	1

**Tabla 15**

*Selección de alternativa para el efector final.*

	<b>COSTO</b>	<b>PESO</b>	<b>TOLERANCIA</b>	<b><math>\Sigma</math></b>	<b>POND.</b>
<b>ALTERNATIVA 1</b>	0.176	0.083	0.063	0.322	3
<b>ALTERNATIVA 2</b>	0.105	0.165	0.063	0.333	2
<b>ALTERNATIVA 3</b>	0.139	0.083	0.125	0.347	1

Como se determina, (Tabla 15), se escoge la alternativa 3, un efector magnético por las ventajas que presenta este efector para la sujeción de las piezas, y porque para el proyecto no se necesita una orientación determinada del elemento sujetado, sino que solo la posición, (Tabla 12).

A continuación, se muestran las características del efector seleccionado, (Tabla 16).

**Tabla 16**

*Características del electroimán.*

<b>TENSIÓN DE ENTRADA</b>	<b>12VDC</b>
<b>CORRIENTE</b>	0.42A
<b>CONSUMO DE ENERGÍA</b>	5W
<b>FUERZA DE RETENCIÓN</b>	5kg

**Nota.** Tomado de MGSYSTEM (2020)

### **Diseño mecánico de los Eslabones**

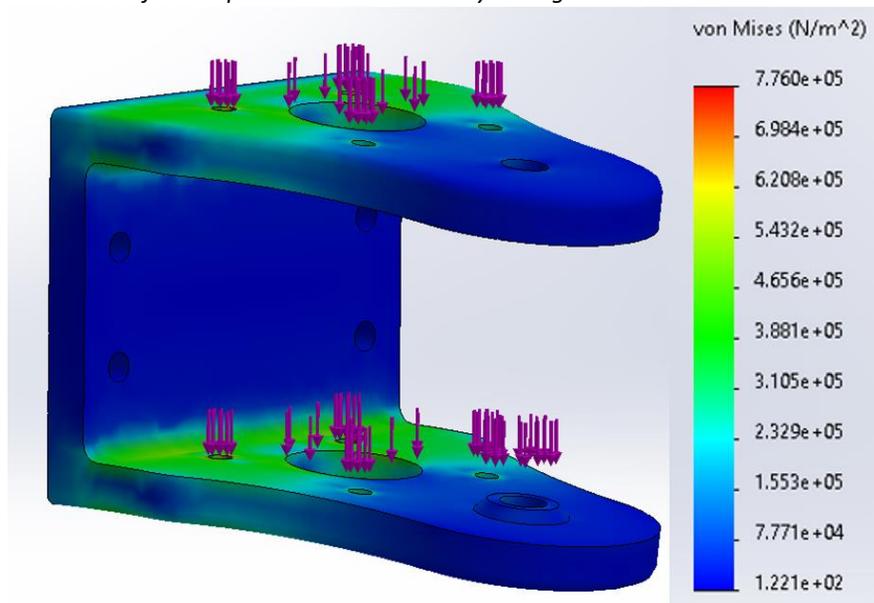
Para el diseño mecánico de los eslabones tomamos en cuenta la configuración del robot (PRR) y la característica principal en cuanto a cargas que caracteriza a este tipo de manipuladores. Por su configuración la mayor carga la va a presentar el eslabón que une la articulación prismática con las demás partes del robot, este eslabón que en verdad actúa como una base para los motores de los dos siguientes eslabones, es el que soporta el peso de todo este conjunto y del actuador

final por lo que sobre este se realiza un estudio simulado de deformación, y podemos así comprobar que el material y la geometría de la pieza cumplan con su función.

Se realiza un estudio mecánico de esfuerzos sobre el eslabón que soporta la mayor carga, es el más crítico y, por lo tanto, se comprueba que el material y la geometría escogidos para las partes es la adecuada. Los esfuerzos máximos que se generan en la pieza (Figura 19) alcanzan los 700kPa y se encuentran muy por debajo del esfuerzo máximo a flexión que puede soportar el material que se encuentra en el orden de los 60 MPa según el fabricante.

**Figura 19**

*Análisis de esfuerzos para el eslabón con mayor carga.*



*Nota.* Tomado de SolidWorks (2018)

### **Diseño geométrico de los eslabones**

El diseño geométrico de los eslabones abarca todos los puntos anteriores. Para llegar a la geometría final de cada eslabón se debe considerar en primer lugar la configuración del manipulador, los sistemas de transmisión de movimiento que va a llevar, el efector final, consideración por el método de manufactura y consideraciones de cableado para el posterior ensamblaje final de la celda. En la Tabla 17 se tienen los elementos diseñados.

Tabla 17

*Eslabones diseñados*

ELEMENTO	DISEÑO
ESLABÓN 1	
ESLABÓN 2	
ESLABÓN 3	

### Módulo 2. Espacio de trabajo

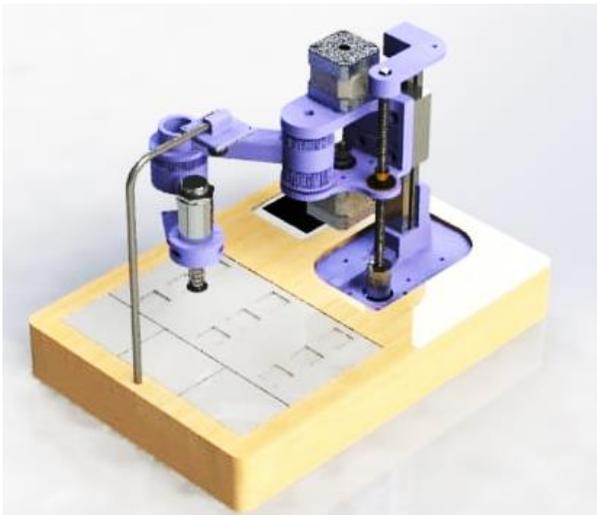
La celda robótica requiere un espacio en el que se van a colocar los elementos a ordenar y sobre el cual va a estar montado el manipulador. Este espacio de trabajo debe estar relacionado con el espacio de trabajo del robot que, según Nava & Ceccarelli (2005), es una característica importante en la aplicación de un robot, es la región del espacio que puede alcanzar el efector final de un manipulador para realizar las acciones específicas dentro de la región posible.

### ***Subsistema 1. Estructura de la celda***

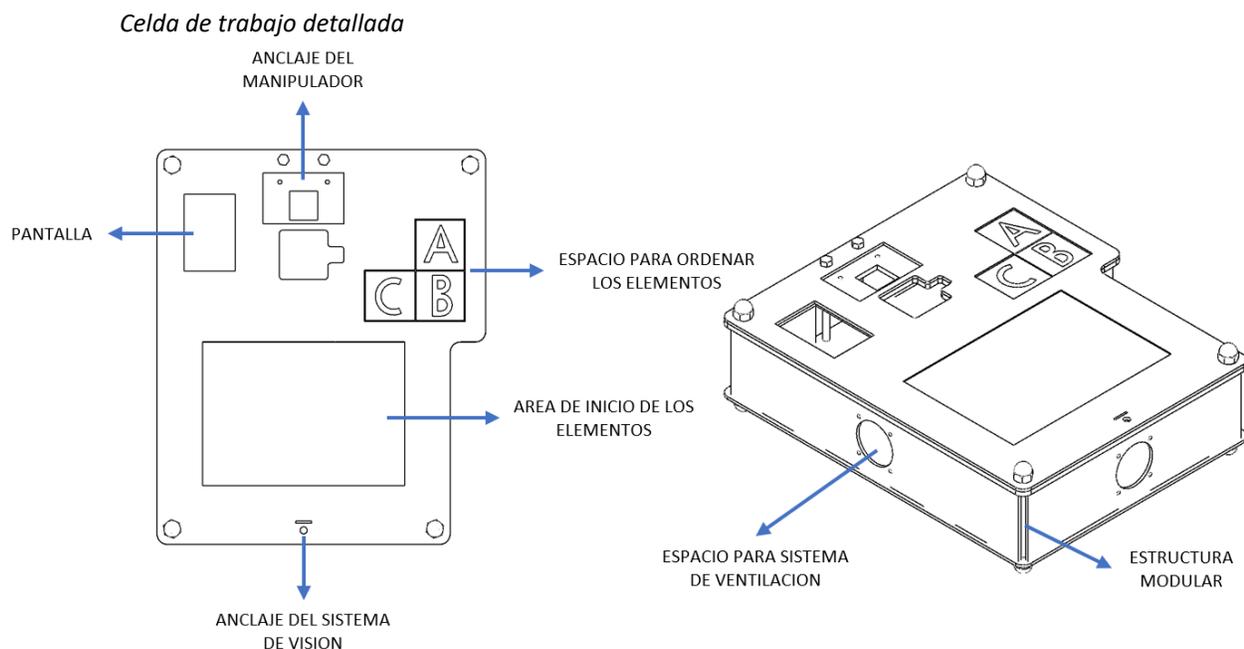
La estructura de la celda está completamente relacionada con las dimensiones del manipulador (370 mm x 386 mm x 327 mm). Los aspectos a considerar para el diseño de la celda son: el montaje del manipulador, la inclusión de una pantalla para la interacción con el usuario, los espacios de inicio y final de los elementos al ser ordenados, el fácil montaje y desmontaje, la inclusión de sistemas de ventilación para los componentes y el montaje del sistema de visión. En la Figura 20 se puede observar un primer diseño de la celda robótica con un primer diseño del manipulador de igual forma.

**Figura 20**

*Primeros bosquejos de la celda de trabajo y manipulador.*



La celda de trabajo se diseña de forma rectangular, cuidando el volumen de trabajo que puede alcanzar el manipulador. En el diseño final de la estructura se consideró una arquitectura modular, se añadió el sistema de ventilación y una mejor distribución de las áreas de trabajo (Figura 21).

**Figura 21**

Para la celda de trabajo se tomarán en cuenta varias opciones de materiales, (Tabla 18).

**Tabla 18**

*Ventajas y desventajas de cada alternativa para la celda de trabajo*

<b>ALTERNATIVA</b>	<b>TIPO DE MATERIAL</b>	<b>VENTAJAS</b>	<b>DESVENTAJAS</b>
<b>ALTERNATIVA 1</b>	Aluminio	Peso bajo	Costo moderado No apto para exteriores Difícil Maquinabilidad No tiene variedad de colores Baja Durabilidad
<b>ALTERNATIVA 2</b>	MDF	Costo bajo Peso bajo Fácil maquinabilidad	No apto para exteriores No resiste golpes No tiene variedad de colores
<b>ALTERNATIVA 3</b>	Acrílico	Peso bajo Apto para exteriores	Costo moderado

	Fácil maquinabilidad
	Variedad de colores
	Durabilidad

### Selección de alternativa

Para la selección del material con el que se elaborará la celda de trabajo se definen los siguientes criterios:

- Costo bajo
- Apto para exteriores
- Durabilidad

**Tabla 19**

*Ponderaciones de cada criterio para el material de la celda.*

CRITERIO	COSTO	EXTERIOR	DURABILIDA D	$\Sigma+1$	%	POND.
<b>COSTO</b>	-	0.5	1	2.5	33.33	0.33
<b>EXTERIOR</b>	0.5	-	0.5	2	26.67	0.27
<b>DURABILIDAD</b>	1	1	-	3	40.00	0.4
<b>TOTAL</b>				7.5	100	1

**Tabla 20**

*Ponderaciones de cada alternativa respecto a cada criterio para el material de la celda.*

Costo	Alternativa 1	Alternativa 2	Alternativa 3	$\Sigma+1$	Pond.
<b>Alternativa 1</b>	-	0.5	0.5	2	0.31
<b>Alternativa 2</b>	1	-	1	3	0.46
<b>Alternativa 3</b>	0.5	0	-	1.5	0.23
<b>Total</b>				6.5	1
Exterior	Alternativa 1	Alternativa 2	Alternativa 3	$\Sigma+1$	Pond.
<b>Alternativa 1</b>	-	0.5	0	1.5	0.25
<b>Alternativa 2</b>	0.5	-	0	1.5	0.25

<b>Alternativa 3</b>	1	1	-	3	0.5
<b>Total</b>				6	1
<b>Durabilidad</b>	<b>Alternativa 1</b>	<b>Alternativa 2</b>	<b>Alternativa 3</b>	<b><math>\Sigma+1</math></b>	<b>Pond.</b>
<b>Alternativa 1</b>	-	0.5	0	1.5	0.25
<b>Alternativa 2</b>	0.5	-	0	1.5	0.25
<b>Alternativa 3</b>	1	1	-	3	0.5
<b>Total</b>				6	1

Tabla 21

Selección de alternativa para el material de la celda.

	<b>Costo</b>	<b>Exterior</b>	<b>Durabilidad</b>	<b><math>\Sigma</math></b>	<b>Pond.</b>
<b>Alternativa 1</b>	0.102	0.068	0.1	0.27	3
<b>Alternativa 2</b>	0.152	0.068	0.1	0.32	2
<b>Alternativa 3</b>	0.076	0.135	0.2	0.411	1

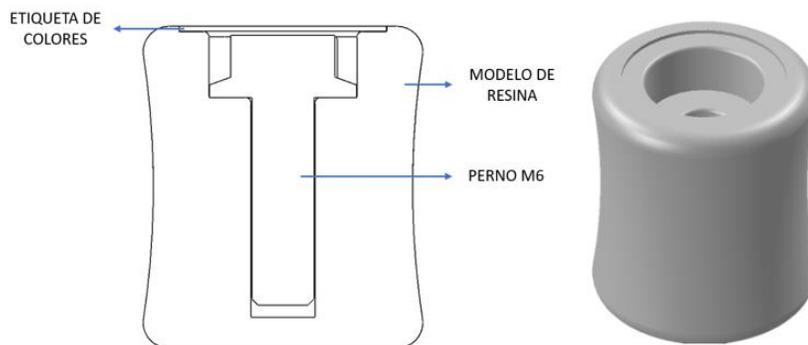
Como se determina, (Tabla 21), se escoge la alternativa 3, una celda de trabajo realizada en corte láser con material acrílico.

### **Subsistema 2. Elementos didácticos**

Para los elementos didácticos con lo que trabajará la celda robótica se optó por una forma cilíndrica evitando así la necesidad de orientación de las piezas, el diseño contempla la colocación de un perno metálico en su interior que será atraído por el electroimán, una etiqueta en la parte superior para la identificación (Figura 22) y se considera que la manufactura se realizará en impresión 3D en Resina, por las ventajas de acabado final y facilidad de costos.

Figura 22

Diseño de los cilindros



### Módulo 3. Potencia y Control

El módulo de potencia y control engloba a todos los componentes eléctricos, electrónicos y el software que hacen funcionar a la celda, divididos en subsistemas; estos son los sensores, la alimentación, los actuadores de las articulaciones, la tarjeta sobre la cual se programa para el control de los movimientos, planificación de las acciones del robot y la detección de los elementos por medio de colores. Cada uno de estos subsistemas se detallan a continuación:

#### ***Subsistema 1. Potencia del Robot***

##### **Selección de Actuadores**

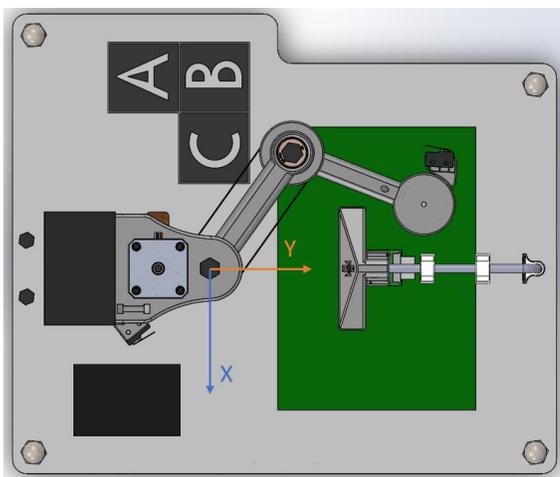
Los motores son los encargados de dar el movimiento al manipulador por medio de las tres articulaciones. Con el torque máximo calculado que requiere cada motor, se escogen los motores adecuados para el funcionamiento del sistema. Para el proyecto se ocuparán motores a pasos debido principalmente al área de trabajo que se obtiene como se verá a continuación y por las ventajas que presentan como una buena precisión, buen rendimiento en la aceleración, arranque y parada, existe un error de paso menor al 5%, tienen un ángulo de rotación proporcional a los pulsos de entrada y las velocidades de rotación son amplias por la frecuencia de pulsos de entrada digitales como menciona la Compañía Levantina de Reductores (2019).

##### ***Comparación del área de trabajo***

Como se vio en el capítulo 2 el área de trabajo de este tipo de manipulador, es decir, todas las ubicaciones a las que puede llegar el efector final están relacionado directamente con el ángulo que los motores pueden girar para dar la posición en el plano X, Y, (Figura 23).

**Figura 23**

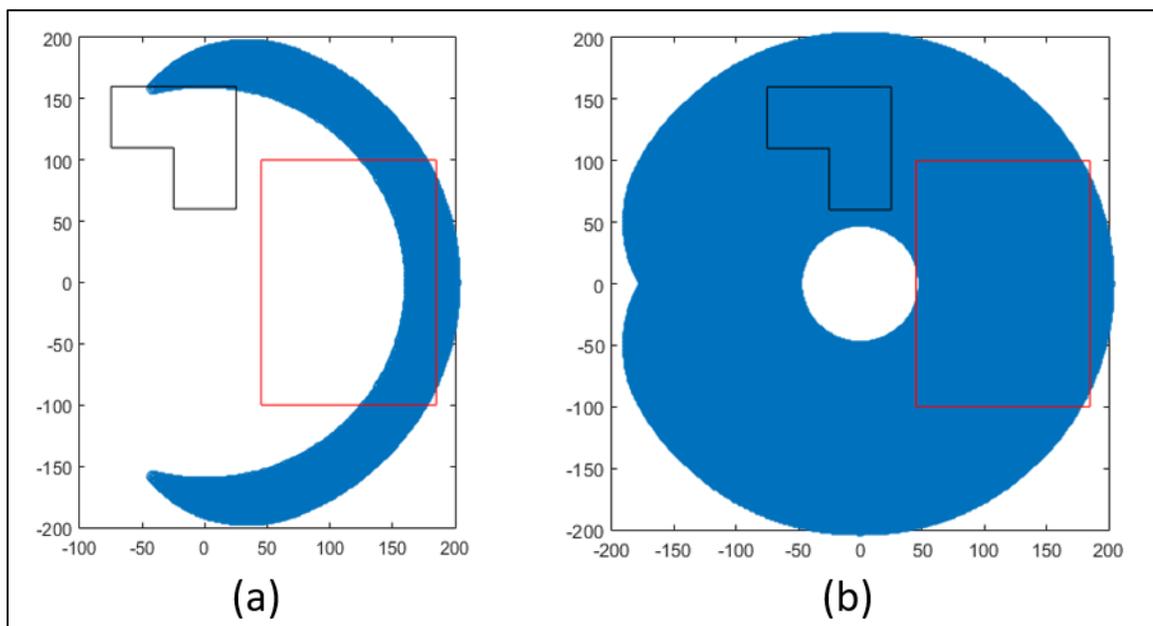
*Manipulador vista desde el eje X,Y*



Si bien es cierto los servomotores son los más utilizados en la robótica para dar movimiento a articulaciones rotacionales porque poseen un control de posición de lazo cerrado, sin embargo, comercialmente en el mercado solo se encuentran servomotores con un giro de hasta  $180^\circ$  teóricos pero que en la práctica solo alcanzan un giro de hasta  $140^\circ$ . Como se puede observar en la Figura 24 el área marcada con azul representa a todas las posiciones que puede llegar el efector final usando los dos tipos de motores, el área marcada en rojo a la zona de inicio de los elementos y la zona en negro la zona objetivo donde son ordenados. En el gráfico (a) se simuló el uso de un servomotor en cada articulación rotacional del robot y, tomando en cuenta que el máximo giro que se puede obtener es de  $140^\circ$ ; mientras que el gráfico (b) representa el mismo caso, pero con el uso de motores a pasos.

**Figura 24**

*Representación del área de trabajo obtenida al usar (a) servomotores y (b) motores a paso.*



Si bien es cierto en el mercado se pueden encontrar servomotores con un giro continuo de  $360^\circ$  y control de posición su costo es bastante elevado y para fines del proyecto los resultados que se pueden conseguir con los motores a pasos son bastante buenos.

Seleccionamos los motores a utilizar por el torque necesario. Para el motor 1, el cual se determinó anteriormente un torque máximo de  $0.18\text{Nm}$ , aplicando un factor de seguridad de 1.6 se obtiene un torque de  $0.288\text{Nm}$ . Con respecto a los motores 2 y 3 de las articulaciones

rotacionales, el torque máximo que necesitan, así mismo, aplicando el factor de seguridad es de 0.224 Nm y 0.176 Nm respectivamente. Con estos datos podemos elegir los motores.

**Tabla 22**

*Características de los motores seleccionados.*

<b>NOMBRE DEL MOTOR</b>	<b>MOTOR 1 PRISMÁTICO</b>	<b>MOTOR 2 Y 3 ROTACIONALES</b>
<b>FLANJE</b>	NEMA 17	NEMA 17
<b>TORQUE MÁXIMO</b>	0.45Nm	0.28 Nm
<b>INERCIA DEL ROTOR</b>	54 gcm <sup>2</sup>	34 gcm <sup>2</sup>
<b>CORRIENTE</b>	1.2A	0.4A
<b>VOLTAJE</b>	12 VDC	12 VDC
<b>ÁNGULO DE PASO</b>	1.8°	1.8°

*Nota.* Tomado de AliExpress: Myers Store (2019), AliExpress: CIOOT Official Store (2016)

El control de los motores a pasos se lo realiza mediante el uso de un driver, el driver A4988, que genera los pulsos en cada bobina del motor para su funcionamiento, además de las protecciones de corriente y temperatura, las características del driver, (Tabla 23).

**Tabla 23**

*Características del driver A4988 para los motores a pasos.*

<b>MODELO</b>	<b>A4988</b>
<b>COLOR</b>	Verde o Rojo
<b>CORRIENTE MÁXIMA</b>	2A
<b>VOLTAJE MÁXIMO</b>	35V
<b>MICROSTEPS</b>	16

*Nota.* Tomado de García (2020)

### **Selección de Alimentación**

Para la alimentación del sistema se debe tomar en cuenta el consumo total energético de todos los componentes del sistema, (Tabla 24), el consumo de cada elemento y el total de todo el sistema. El voltaje que requerimos de la fuente son 12VDC para la alimentación de los motores, ventiladores luz y electroimán, para lo cual se presentan varias opciones, (Tabla 32).

Tabla 24

*Consumo energético de cada elemento del sistema.*

ELEMENTO	CONSUMO ENERGÍA
MOTOR 1	9.6 [W]
MOTOR 2	5 [W]
MOTOR 3	5 [W]
ILUMINACIÓN LED	0.55 [W]
ELECTROIMÁN	5 [W]
VENTILADORES	7.2 [W]
<b>TOTAL</b>	<b>32.35 [W]</b>

El consumo total de todos los elementos es de 32.35 [W], (Tabla 24), pero multiplicado por un factor de seguridad de 1.6 el consumo total es de 51.76 [W], tomando en cuenta la adición de otros elementos, se debe elegir una fuente que proporcione más de 50 [W].

Tabla 25

*Ventajas y desventajas de cada alternativa para la fuente de alimentación.*

ALTERNATIVA	TIPO DE FUENTE	VENTAJAS	DESVENTAJAS
<b>ALTERNATIVA 1</b>	Baterías de plomo	Recargables	Peso Alto Costo Alto Ciclo de vida limitado Necesita mantenimiento Tiempo de uso reducido
<b>ALTERNATIVA 2</b>	Baterías Lipo	Peso bajo Mejor adaptación	Costo Alto Deterioro irreversible No soporta altas temperatura Período de tiempo de uso limitado

<b>ALTERNATIVA 3</b>	Fuente de poder	Costo bajo	Peso moderado
	AC/DC	Eficiencia alta	Requiere de un cable
		Temperatura alta de funcionamiento	externo que proporcione AC
		Período de tiempo de uso ilimitado	
		No requiere ser autónoma	

### Selección de alternativa

Para la selección de la fuente de alimentación se definen los siguientes criterios:

- Costo
- Peso
- Tiempo de uso

Tabla 26

*Ponderaciones de cada criterio para la fuente de alimentación*

CRITERIO	COSTO	PESO	TIEMPO DE USO	$\Sigma+1$	%	POND.
<b>COSTO</b>	-	0.5	0.5	2	28.57	0.29
<b>PESO</b>	0.5	-	1	2.5	35.71	0.36
<b>TIEMPO DE USO</b>	0.5	1	-	2.5	35.71	0.36
<b>TOTAL</b>				7	100	1

Tabla 27

*Ponderaciones de cada alternativa respecto a cada criterio para la fuente de alimentación.*

COSTO	ALTERNATIVA 1	ALTERNATIVA 2	ALTERNATIVA 3	$\Sigma+1$	POND.
	1	2	3		
<b>ALTERNATIVA 1</b>	-	0.5	0	1.5	0.25
<b>ALTERNATIVA 2</b>	0.5	-	0	1.5	0.25

<b>ALTERNATIVA 3</b>	1	1	-	3	0.5
<b>TOTAL</b>				6	1
<b>PESO</b>	<b>Alternativa 1</b>	<b>Alternativa 2</b>	<b>Alternativa 3</b>	<b>Σ+1</b>	<b>Pond.</b>
<b>ALTERNATIVA 1</b>	-	0	0	0	0
<b>ALTERNATIVA 2</b>	1	-	1	3	0.6
<b>ALTERNATIVA 3</b>	1	0	-	2	0.4
<b>TOTAL</b>				5	1
<b>TIEMPO DE USO</b>	<b>Alternativa 1</b>	<b>Alternativa 2</b>	<b>Alternativa 3</b>	<b>Σ+1</b>	<b>Pond.</b>
<b>ALTERNATIVA 1</b>	-	0.5	0	1.5	0.25
<b>ALTERNATIVA 2</b>	0.5	-	0	1.5	0.25
<b>ALTERNATIVA 3</b>	1	1	-	3	0.5
<b>TOTAL</b>				6	1

Tabla 28

*Selección de alternativa para la fuente de alimentación.*

	<b>COSTO</b>	<b>PESO</b>	<b>TIEMPO DE USO</b>	<b>Σ</b>	<b>POND.</b>
<b>ALTERNATIVA 1</b>	0.073	0	0.09	0.163	3
<b>ALTERNATIVA 2</b>	0.073	0.216	0.09	0.379	2
<b>ALTERNATIVA 3</b>	0.145	0.144	0.18	0.469	1

Se escoge la alternativa 3, (Tabla 28), una fuente de poder AC/DC, como fuente de alimentación que proporcionará la energía necesaria a todos los elementos eléctricos y electrónicos del sistema, (Tabla 29), se muestran sus características.

Tabla 29

*Características de la fuente de alimentación.*

<b>TENSIÓN DE ENTRADA</b>	<b>100-240V AC</b>
<b>TENSIÓN DE SALIDA</b>	11.4-13.2VDC
<b>CORRIENTE MÁXIMA</b>	5A
<b>POTENCIA</b>	60W
<b>TEMPERATURA DE FUNCIONAMIENTO</b>	70°C

*Nota.* Tomado de Tecnología informática (2020)

### Selección de Sensores

Se usan finales de carrera como sensores de posición, que se colocan en los brazos del robot, como señalización cuando el brazo ha llegado a la parte superior o inferior del movimiento, en este caso, cuando llega al punto antes de chocar contra la estructura, esto sirve de retroalimentación cuando ocurre algún evento y obtener una posición determinada.

### Subsistema 2. Control del Robot

Para el control del Robot lo más recomendable es usar un microcontrolador o microprocesador que permita realizarlo, las opciones más comunes es el uso de un Arduino o una Raspberry pi, (Tabla 30), se muestran las características de cada uno y a continuación la selección de la mejor alternativa.

**Tabla 30**

*Características de Arduino vs Raspberry pi.*

ALTERNATIVA	NOMBRE	VENTAJAS	DESVENTAJAS
<b>ALTERNATIVA 1</b>	Arduino	Aplicaciones sencillas para uso concreto Costo bajo	Requiere procesador como una PC No tiene conectores para internet Requiere alimentación serial.
<b>ALTERNATIVA 2</b>	Raspberry pi	No requiere un procesador Puerto Wifi y Ethernet Control remoto Acople para pantalla Cargador AC	Arranca solo con un sistema operativo Costo alto

### Selección de alternativa

Para la selección del controlador se definen los siguientes criterios:

- Costo
- Procesador

Tabla 31

*Ponderaciones de cada criterio para la Raspberry pi4.*

CRITERIO	COSTO	PROCESADOR	$\Sigma+1$	%	POND.
<b>COSTO</b>	-	0.5	1.5	42.86	0.43
<b>PROCESADOR</b>	1	-	2	57.14	0.57
<b>TOTAL</b>			3.5	100	1

Tabla 32

*Ponderaciones de cada alternativa respecto a cada criterio para la Raspberry pi4.*

COSTO	ALTERNATIVA 1	ALTERNATIVA 2	$\Sigma+1$	POND.
<b>ALTERNATIVA 1</b>	-	1	2	0.67
<b>ALTERNATIVA 2</b>	0	-	1	0.33
<b>TOTAL</b>			3	1
PROCESADOR	Alternativa 1	Alternativa 2	$\Sigma+1$	Pond.
<b>ALTERNATIVA 1</b>	-	0	1	0.33
<b>ALTERNATIVA 2</b>	1	-	2	0.67
<b>TOTAL</b>			3	1

Tabla 33

*Selección de alternativa para la Raspberry pi4.*

	COSTO	PROCESADOR	$\Sigma$	POND.
<b>ALTERNATIVA 1</b>	0.288	0.188	0.476	2
<b>ALTERNATIVA 2</b>	0.142	0.382	0.524	1

Como se determina, (Tabla 33), se escoge la alternativa 2, ya que la Raspberry pi al contar con un procesador integrado, mediante la instalación de un sistema operativo se puede instalar también el sistema ROS, que permite el control completo del robot, el control de los motores también se puede realizar con la Raspberry en vez de un Arduino, así mismo la integración de una pantalla que permita operar una interfaz de manera fácil. Se muestran las características de la Raspberry pi, (Tabla 34).

Tabla 34

Características de la Raspberry Pi4

<b>CONECTIVIDAD</b>	2.4 GHZ AND 5.0 GHZ LAN, BLUETOOTH 5.0, BLE GIGABIT ETHERNET 2 × USB 3.0 PORTS 2 × USB 2.0 PORTS
<b>GPIO</b>	Standard 40-pin GPIO header
<b>VIDEO Y SONIDO</b>	2 puertos × micro HDMI 2-lane MIPI DSI puerto display 2-lane MIPI CSI puerto cámara 4-pole puertos stereo audio and composite video
<b>MULTIMEDIA</b>	H.265 (4Kp60 decode) H.264 (1080p60 decode, 1080p30 encode) OpenGL ES, 3.0 graphics
<b>ENTRADA</b>	5V DC via USB-C connector (minimum 3A) 5V DC via GPIO header (minimum 3A) Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
<b>TEMPERATURA DE OPERACIÓN</b>	0-50°C

*Nota.* Tomado de APM, Raspberry pi 4 Model B 2GB RAM (2020)

#### Módulo 4. Interacción

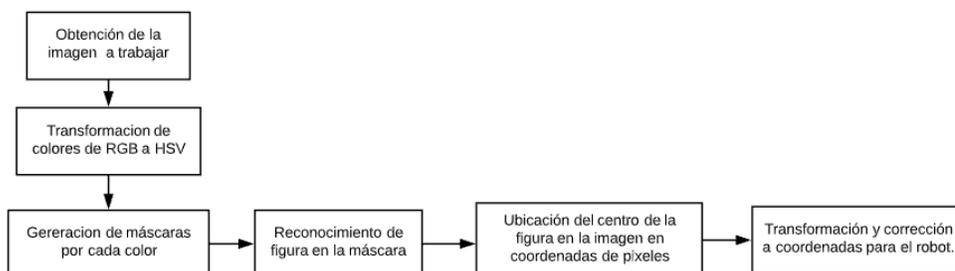
##### ***Subsistema 1. Visión Artificial del Robot***

La visión artificial de la celda robótica realiza la función de reconocimiento de colores y ubicación de los elementos didácticos que se van a ordenar sobre el área de trabajo, para esto se usa una cámara compatible con Raspberry (piCamera) y algoritmos de procesamiento de imágenes programados mediante OpenCV (biblioteca libre de visión artificial originalmente desarrollada por Intel) que siguen una secuencia como se muestra en la Figura 25. El primer paso es la obtención de la imagen con la que se va a trabajar mediante la cámara instalada, después transformar esa imagen en un rango de color HSV, que permite la detección del color de la etiqueta que identifica a cada elemento, la etiqueta se refiere a un círculo de un determinado

color que está colocado en la parte superior de cada cilindro (elemento didáctico de la celda) de esta manera se puede usar la cámara de la celda como un sensor de presencia de un determinado cilindro y además se determina la posición respecto al espacio de trabajo del manipulador robótico para posicionar el actuador.

**Figura 25**

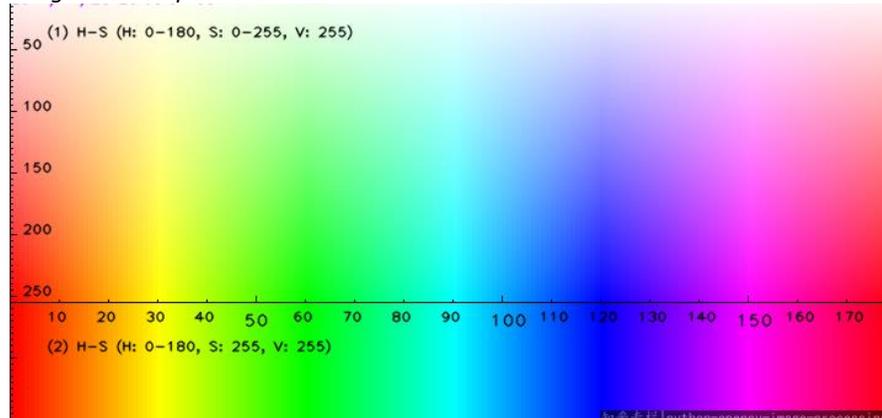
*Pasos para procesar la imagen con la cámara*



La detección de colores se realiza usando el modelo de color HSV por su facilidad descrito en el capítulo anterior, OpenCV y la cámara con la que se trabaja identifica los colores en un modelo RGB por lo que para trabajar sobre una imagen transformamos este modelo al modelo HSV. La detección de colores se realiza obteniendo una máscara que no es más que filtrar una característica específica de la imagen que en este caso sería un rango de valores en HSV para cada color con el que se trabaja. Los rangos de valores que se utilizan para el filtrado deben ser calibrados específicamente para los colores con los que se va a trabajar, se trata de un proceso de prueba y error que depende en una gran parte, no solo al color en sí de cada etiqueta, si no, a la iluminación del momento que se obtuvo la imagen.

**Figura 26**

*Rangos de HSV para cada color*



Como se mencionó una parte crítica de la que dependen los resultados de la aplicación de visión artificial para cualquier campo es la iluminación del área de la que se va a obtener la imagen y esta celda robótica no es la excepción. La iluminación y el fondo sobre el cual van a estar colocados los cilindros ayudan a que los resultados sean óptimos y al final conllevan a una buena repetibilidad del sistema completo. Dado que la iluminación de toda la celda de trabajo, como se vio en el diseño, se ve totalmente influida por la iluminación del ambiente en el que se encuentre se colocó una fuente de iluminación LED propia de la celda a un lado de la cámara para corregir en gran parte esta dependencia (Figura 26).

**Tabla 35**

*Características de la cámara para la Raspberry Pi4*

<b>DISPARO</b>	<b>1/4, 5M</b>
<b>LONGITUD FOCAL</b>	3.29
<b>NÚMERO DE PÍXELES</b>	2592x1944
<b>TAMAÑO DEL PÍXEL</b>	1.4 x 1.4 $\mu\text{m}$
<b>LENTE</b>	$f=3.6 \text{ mm}$
<b>ÁNGULO DE VISIÓN</b>	$54 \times 41^\circ$
<b>CAMPO DE VISIÓN</b>	2 x 1.33 m a 2 m
<b>VÍDEO</b>	1080p a 30fps con codec H.264 (AVC) Hasta 90fps vídeo en VGA
<b>TAMAÑO DE LA PLACA</b>	25 x 24 mm

*Nota.* Tomado de APM (2020)

### ***Subsistema 2. Planificador***

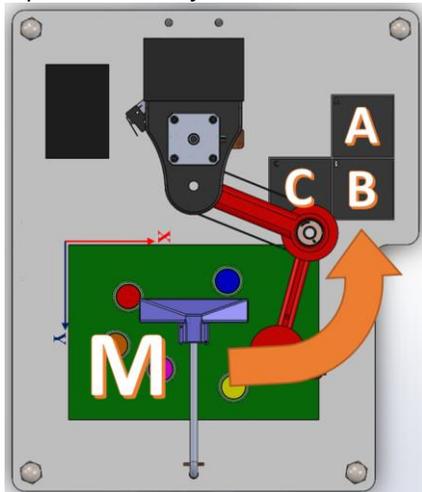
La función del sistema de celda Robótica es ordenar cinco cilindros de colores de la forma en la que se desee, llevándolos desde el espacio *M* hasta las posiciones marcadas con las letras *A*, *B* y *C* como se ve en la Figura 27. Los cilindros pueden estar apilados de dos en el espacio *M* y de igual forma pueden estar apilados hasta de tres en las posiciones *A*, *B* y *C*; lo que evidentemente requiere de un plan para conseguirlo.

Para que el manipulador robótico que se diseñó en este proyecto genere un plan de acciones que se van a ejecutar para ordenar los elementos, se va a utilizar un planificador de tipo STRIPS. Este planificador consta de dos partes indispensables que trabajan juntas para generar una

solución. La primera parte es la descripción del sistema total, como se vio en la teoría, plantear un problema de planificación conlleva, en términos simples, describir cual es el estado de cada elemento que entra en juego en el sistema, describir de igual forma el estado final que se desea alcanzar y las acciones que puede realizar el manipulador para llegar del estado inicial al objetivo. Toda esta descripción se la realiza en el lenguaje STRIPS en un archivo de texto plano.

**Figura 27**

*Espacios de trabajo de la Celda Robótica*



Teniendo el problema planteado la segunda parte consiste en encontrar el plan resolviendo el problema para lo cual, y por razones de que desarrollar un algoritmo de este tipo no es un objetivo del proyecto, se utilizará el algoritmo de uso libre desarrollado por Wesley Tansey en el 2001 en lenguaje Python, un solucionador lineal STRIPS para problemas simples como este es el caso.

El solucionador STRIPS de Tansey, como su autor y la teoría lo indican, se trata de un solucionador de orden total, ya que asume que todos los pasos pueden planificarse uno tras otro con un orden estricto. Trabaja hacia atrás comenzando en el estado de la meta y buscando una acción fundamentada (una con todas las variables reemplazadas por literales) que satisfaga una precondición del estado de la meta. Cuando encuentra una acción aceptable, la añade al plan y la acción recién añadida se convierte en el nuevo estado objetivo. Si se añade una acción que entra en conflicto con otro objetivo ya satisfecho, se vuelve a introducir el objetivo bloqueado y se satisface de nuevo. El algoritmo sigue trabajando hacia atrás, apilando submetas unas sobre otras mediante una búsqueda en profundidad hasta que satisface todos los objetivos o agota todos los planes posibles y se da por vencido.

El algoritmo solucionador trabaja con el lenguaje de descripción de problemas STRIPS que lo generemos en un archivo de texto plano en el cual la primera línea es el estado inicial del sistema como ejemplo de la siguiente forma:

*Initial state: enMesa(Ba), enMesa(Bb), Libre(Ba), Libre(Bb), Vacio(A)*

Cada argumento del estado inicial debe ser un literal. La segunda línea describe el estado objetivo al que se quiere que llegue el sistema, como ejemplo:

*Goal state: enA(Ba), On(Bb,Ba)*

A partir de la tercera línea se declaran las acciones que el sistema puede realizar con las precondiciones y postcondiciones de cada acción como se vio en la teoría, como ejemplo:

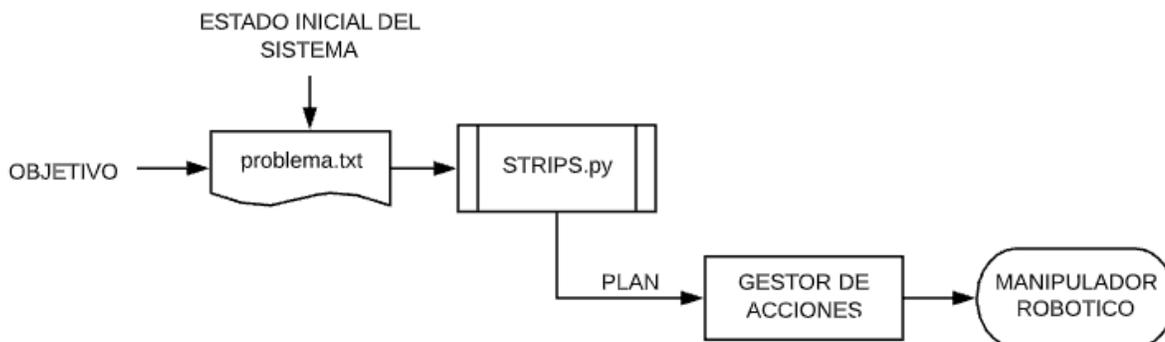
*MoveraA(Bx)*

*Preconditions: Vacio(A), Libre(Bx), EnMesa(Bx)*

*Postconditions: EnA(Bx), !Vacio(A), !EnMesa(Bx)*

**Figura 28**

*Diagrama general del Subsistema Planificador*



### **Subsistema 3. Interfaz Gráfica**

La interfaz gráfica es el medio por el cual el usuario u operador de la celda robótica interactúa con el sistema. La interfaz gráfica se desarrollará mediante la librería Tkinter con la que cuenta Python en todas sus versiones y se presentará mediante una pantalla táctil compatible con la Raspberry pi4, que permite no solo visualizar la HMI (Interfaz humano-maquina) si no que interactuar también con ella, se muestra el modelo de la pantalla a usar en la Tabla 36 y todas las características de la misma.

Tabla 36

*Características de la pantalla para la Raspberry Pi4*

<b>TAMAÑO</b>	<b>85.42 X 55.60 MM</b>
<b>RESOLUCIÓN</b>	320 x 480 píxeles
<b>PESO</b>	49 g
<b>TIPO DE LCD</b>	TFT
<b>INTERFAZ LCD</b>	SPI
<b>TIPO DE PANTALLA TÁCTIL</b>	Resistiva
<b>MEJOR VISUALIZACIÓN CON RELACIÓN DE ASPECTO</b>	8:5
<b>CONTROLADOR DE PANTALLA TÁCTIL</b>	XPT2046

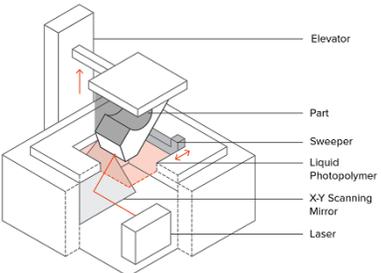
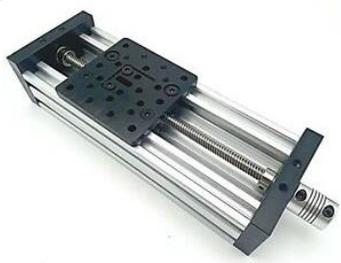
*Nota.* Tomado de Electronics (2020)

### Resumen del capítulo

El capítulo terminado detalló el diseño general de la celda robótica, a continuación, (Tabla 46), se detallan los componentes, métodos y algoritmos elegidos y diseñados para todos los subsistemas definidos, (Tabla 7), la construcción e implementación de cada subcapítulo para la celda robótica se realizará en el siguiente capítulo.

Tabla 37

*Selección final para los subsistemas del manipulador.*

<b>Estructural</b>	Estructura física	Manufactura Aditiva: Resina	
Transmisión	Tornillo de posicionamiento		

---

		Correa dentada y polea	
Eector Final		Electroimán	
Eslabones		Eslabón 1	
		Eslabón 2	
		Eslabón 3	
<b>Espacio de trabajo</b>	Celda de trabajo	Corte laser: Acrílico	

---

**Potencia y control**    Actuadores y control    Motores a pasos



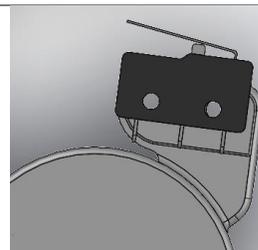
Módulo driver  
A4988



Alimentación    Fuente de poder  
AC/DC



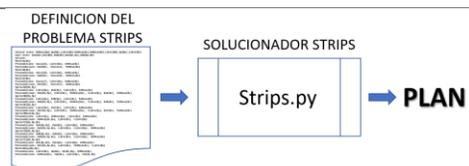
Sensores    Final de carrera



Control    Raspberry pi4

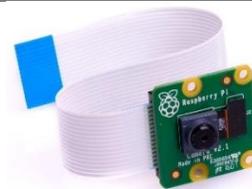


**Interacción**    Planificación    Planificador  
STRIPS



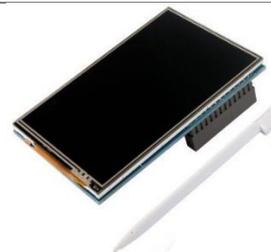
---

Visión            Cámara CAM 4  
Artificial        Raspberry pi



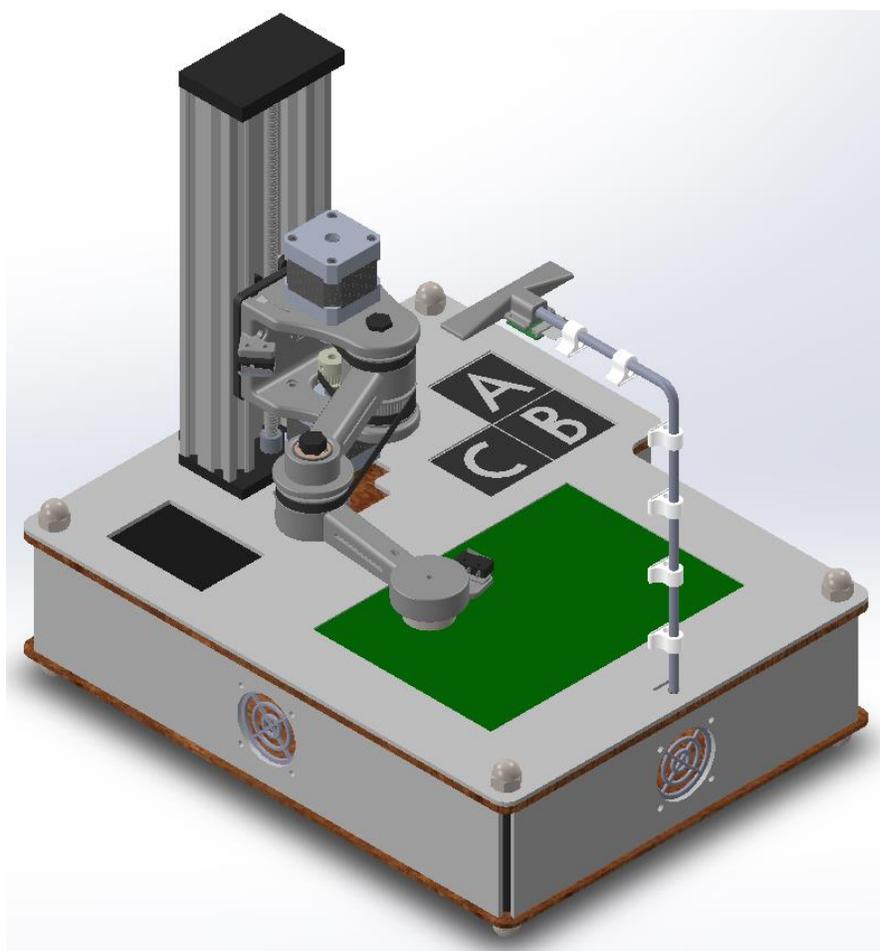
---

Interfaz        Pantalla 3.5"  
gráfica         Raspberry pi



---

Estructura final de la Celda Robótica.



## Capítulo 4

### Construcción, implementación y evaluación

En el capítulo se detalla el proceso de construcción del manipulador con la implementación de cada componente electrónico, mecánico, programación y el ensamblaje con la celda robótica, así mismo, una evaluación del funcionamiento del sistema con pruebas y resultados de la celda.

#### Módulo 1. Sistema Mecánico

##### *Construcción y ensamblaje del manipulador*

Las partes del manipulador, como los eslabones, el acople para los motores rotacionales, los elementos a clasificar (cilindros) y unos acoples para la visión artificial diseñados, fueron impresos mediante SLA, en una impresora de Resina marca ANYCUBIC Photon Mono X tecnología LCD cuyas especificaciones técnicas, (Tabla 42) y los parámetros de impresión de las piezas, (Tabla 43).

**Tabla 38**

*Especificaciones técnicas de la impresora.*

PARÁMETRO	VALOR
RESOLUCIÓN PANTALLA	4K, 3840 x 2400 px, 0.05 [mm] en XY
VOLUMEN DE IMPRESIÓN	192 x 125 x 245 [mm]
VELOCIDAD MÁXIMA DE IMPRESIÓN	60 [mm/h] *con una altura de capa de 0,05[mm]
ALTURA DE CAPA	0.01 – 0.15 [mm]

**Tabla 39**

*Parámetros de impresión.*

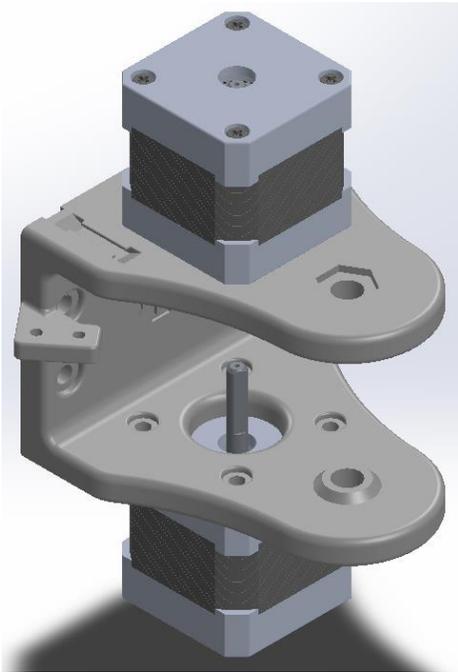
	MATERIAL 1	MATERIAL 2
MATERIAL	Resina UV 405 nm	Resina UV 405 nm
COLOR	Blanco	Gris
MARCA	ANYCUBIC	ANYCUBIC
ALTURA DE CAPA	0.02 [mm]	0.02 [mm]
TIEMPO DE CURADO POR CAPA	1.4 [s]	1.8 [s]

## Motores

Por el diseño del manipulador, los dos motores que accionan las articulaciones rotacionales deben ir montados sobre el eslabón 1 que se diseñó en el capítulo anterior (Figura 29), mientras que el motor que acciona la junta prismática va montado en el acople del tornillo de posicionamiento (Figura 30), y a su vez, el eslabón 1 se fija a la placa móvil. Todo este conjunto es considerado la consideración prismática del manipulador como se ve en la Figura 31.

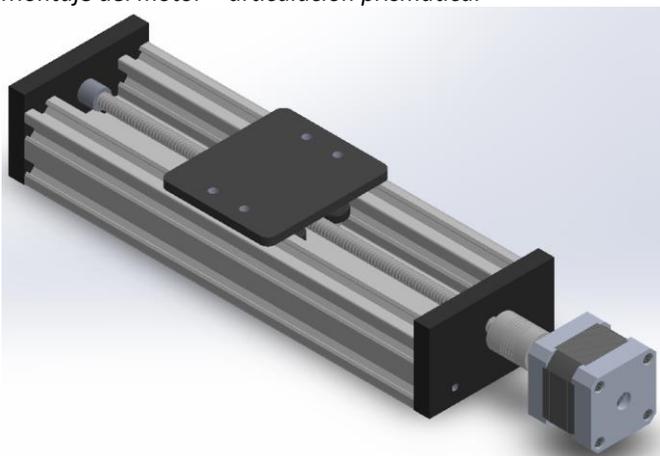
**Figura 29**

*Montaje de los motores – articulaciones rotacionales*



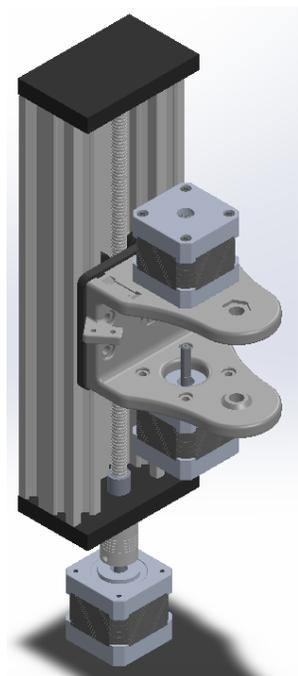
**Figura 30**

*Montaje del motor – articulación prismática.*



**Figura 31**

*Conjunto articulación prismática del manipulador*

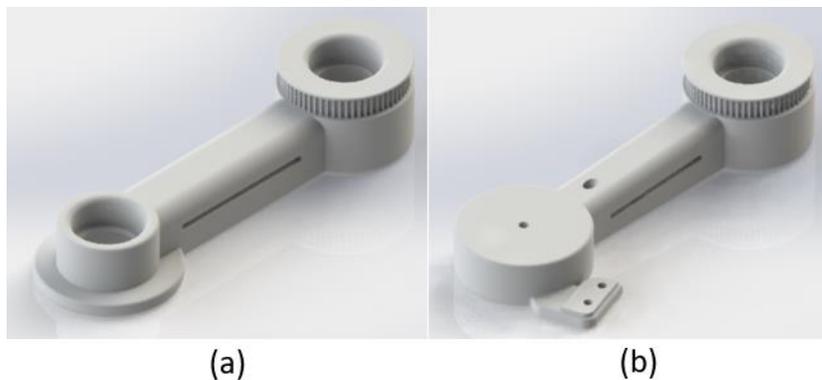


### **Eslabones y Sistema de trasmisión**

Para completar la configuración PRR del manipulador se requiere de dos eslabones o brazos para generar las juntas rotacionales. La manufactura de los eslabones se realizó, de igual forma, en impresión 3D (SLA) en el material de Resina como se determinó en el capítulo anterior, los eslabones tienen acoplados un engrane para el sistema de transmisión por correa con el acople de cada motor, (Figura 32).

**Figura 32**

*Eslabones impresos mediante SLA (a)Eslabón 'brazo 1' y (b) Eslabón 'brazo 2'.*



La transmisión de movimiento angular de los motores hacia cada articulación se consigue mediante el conjunto correa-polea. La transmisión entre los engranajes se realiza mediante las

correas dentadas, (Figura 33), las correas síncronas como se las conoce comercialmente, en Budynas & Nisbett (2008), mencionan que las bandas de sincronización son fabricadas de una tela impregnada con caucho y alambre de acero, el núcleo de sus dientes ofrece rigidez por lo que es la parte que soporta la mayoría de los esfuerzos y así descargar las tensiones del resto de la correa.

**Figura 33**

*Banda dentada de sincronización de perfiles curvilíneos.*



*Nota.* Tomado de KALARM (2018)

Este tipo de banda no se estiran ni se deslizan por lo que la potencia se transmite a una velocidad angular regular, debido a que son bandas dentadas no necesita una tensión inicial en transmisiones con ejes fijos, como es el caso de la aplicación, se escoge una banda dentada de sincronización con perfiles curvilíneos ya que la zona crítica de concentración de las tensiones se ubica en el centro del diente aumentando la duración de la banda.

Las poleas usadas, (Figura34), son acoples directos a los motores a pasos, son poleas GT2 de aluminio que se encuentran fácilmente en el mercado, tienen un diámetro de 16 mm, 20 dientes, el paso es 2, y debido a que va acoplado al motor toma la velocidad de este.

**Figura 34**

*Polea de aluminio para motor a pasos.*



*Nota.* Tomado de KALARM (2018)

Tomando en cuenta los datos de torque en cada motor y 100rpm como velocidad crítica por motivos de cálculos se realiza el diseño de los engranes rectos que van acoplados en los eslabones.

$$\text{Par motor 2: } T2 = 0.014 \text{ kgf.m} \quad (60)$$

$$\text{Par motor 3: } T3 = 0.011 \text{ kgf.m} \quad (61)$$

Potencia transmitida:

$$P2 = \frac{100 \text{ rpm} \times T2}{973.8} = 0.0014 \text{ [kW]} \quad (62)$$

$$P3 = \frac{100 \text{ rpm} \times T3}{973.8} = 0.0011 \text{ [kW]} \quad (63)$$

Se calcula la potencia corregida, la cual es la potencia que tendrá la banda, tomando en cuenta un coeficiente de corrección determinado por el tipo de motor y el tipo de funcionamiento, de acuerdo a las especificaciones para un funcionamiento normal el coeficiente es 1.1.

$$\text{Potencia corregida: } Pc2 = P2 \times 1.1 = 0.00216 \text{ HP} \quad (64)$$

$$Pc3 = P3 \times 1.1 = 0.00169 \text{ HP} \quad (65)$$

Teniendo las poleas para los motores a pasos de 20 dientes, este dato es una constante para los demás cálculos. Por razones de facilidad en el diseño buscamos un diámetro aproximado de 40mm o menor para el engrane de los eslabones, asumimos un numero de dientes del engrane mayor de 62 dientes y teniendo en cuenta que por las características de los motores a pasos utilizados cada paso representa 1.8° resulta que:

$$i = \frac{z_1}{z_2} = \frac{20}{62} = 0.323 \quad (66)$$

$$i = 0.323 \quad (67)$$

Para el diámetro del engranaje, teniendo un diámetro exterior de 12.2 mm de la polea:

$$i = \frac{d}{D} = \frac{12.2}{D} = 0.323 \quad (68)$$

$$D = 37.77 \text{ mm} \quad (69)$$

Distancia entre ejes (E) para una relación de transmisión, la distancia entre ejes debe ser mayor que el diámetro del engrane, por lo que se toma un valor de 36.5 mm y de 100 mm con una

relación de 1 por cuestiones de diseño. La longitud de las correas se realiza mediante la siguiente ecuación.

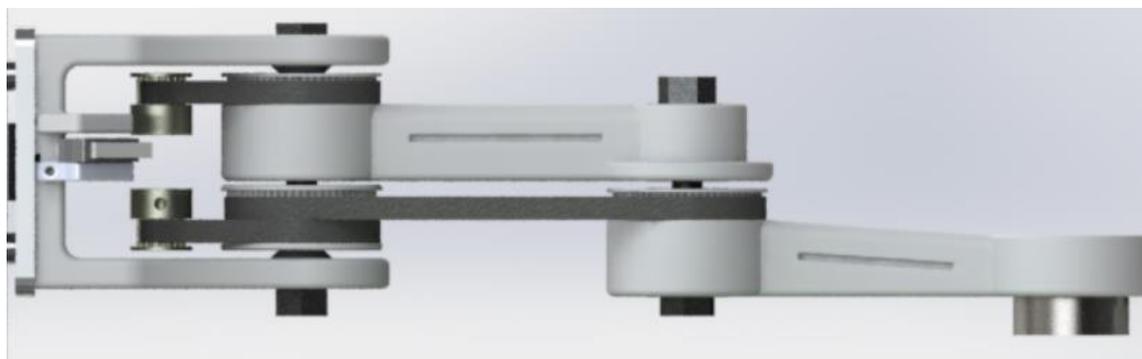
$$Lp1 = Lp2 = 2E + \frac{\pi}{2}(D + d) + \frac{(D-d)^2}{4E} \approx 160 \text{ mm} \quad (70)$$

$$Lp3 \approx 320 \text{ mm} \quad (71)$$

En la Figura 35 se puede observar los tres sistemas de transmisión de movimiento para las articulaciones rotacionales.

**Figura 35**

*Implementación del sistema de transmisión por correa y poleas.*

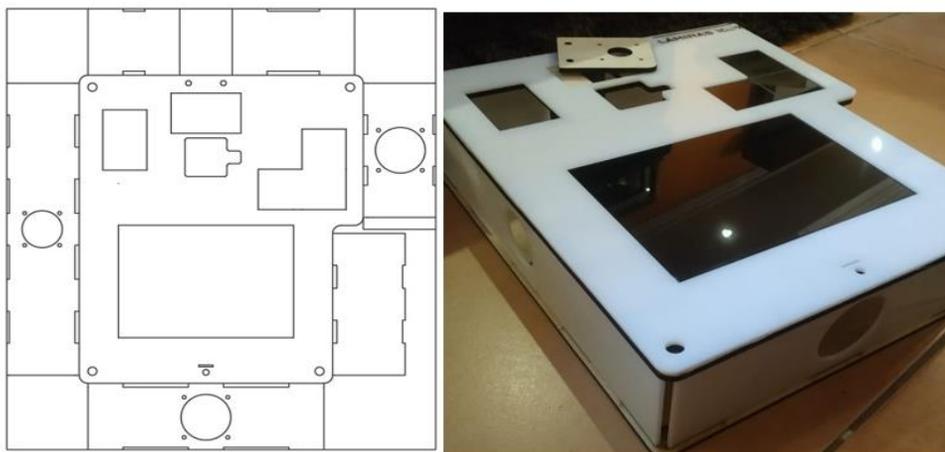


**Construcción y ensamblaje de la celda de trabajo**

Para la implementación del manipulador en la celda robótica, se construyó la estructura modular de la celda de trabajo, como se muestra en la Figura 36 (a) su diseño para el corte en acrílico y construcción de la celda de trabajo, (Figura 36(b)).

**Figura 36**

*(a)Diseño para corte (b)Construcción de la celda de trabajo*

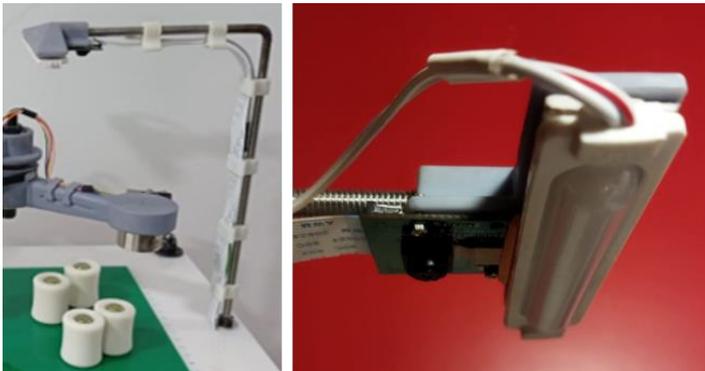


La ubicación de la cámara para la visión artificial se posicionó de la manera más práctica y óptima para el sistema enfocando la totalidad del área de trabajo y en la misma dirección se coloca la iluminación, (Figura 37), de esta manera se comprobó que se reducen las sombras y minimiza el reflejo que pueda causar en los objetos.

Este tipo de iluminación frontal, según Nogué & Antiga (2012), es muy usado para una correcta detección de colores en superficies, sin embargo, la desventaja que presenta es que puede haber reflejos en ciertas superficies, por lo que también se realizaron pruebas en diferentes superficies con diferentes colores, eligiéndose finalmente una superficie mate de color verde que no interviene en el reflejo ni en la detección de los demás colores de los cilindros.

**Figura 37**

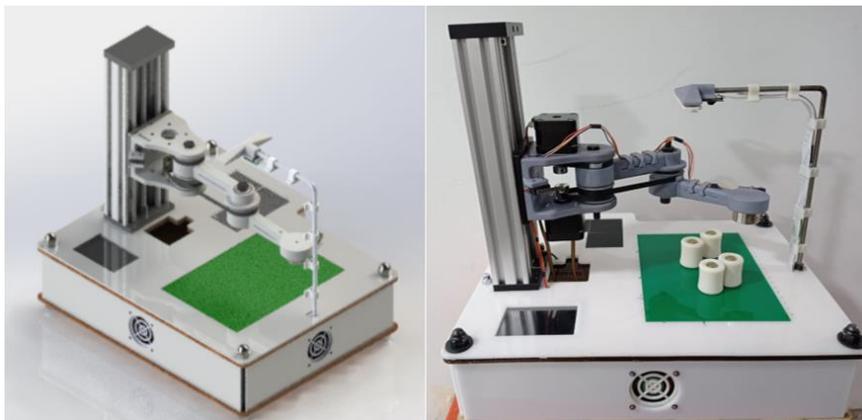
*Sistema de Visión e Iluminación de la celda*



Se muestra la implementación completa de la celda de trabajo, (Figura 38), con todos los elementos mecánicos, motores, sistema de transmisión, área de trabajo, cámara e iluminación.

**Figura 38**

*Implementación total de la celda de trabajo (a) en Software CAD y (b) Real.*



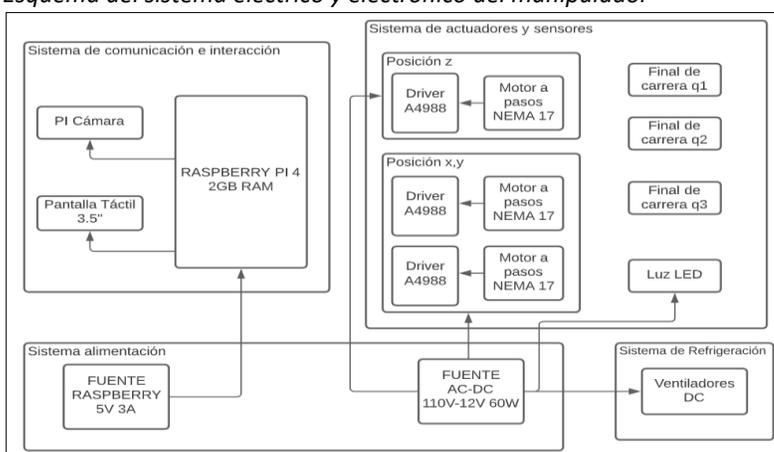
## Módulo 2. Sistema Eléctrico

### *Implementación componentes electrónicos*

El sistema eléctrico, los elementos y las conexiones generales entre ellos para el sistema eléctrico y electrónico del manipulador se pueden observar en la Figura 39. Todo el conjunto lo componen cuatro sistemas: Sistema de comunicación e interacción, sistema de actuadores y sensores, sistema de ventilación y sistema de alimentación. Las características técnicas de cada componente se detallaron en el capítulo anterior.

**Figura 39**

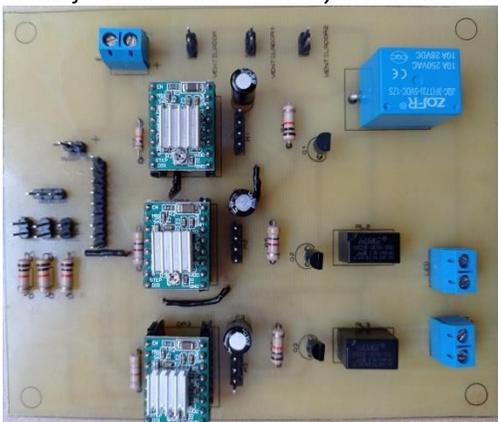
*Esquema del sistema eléctrico y electrónico del manipulador*



Después de toda la implementación mecánica se realiza el montaje de los elementos electrónicos en la parte interna de la celda de trabajo, utilizando una placa PCB, (Figura 40), tomando en cuenta el espacio para evitar inconvenientes en el mantenimiento o reemplazo de ciertos elementos o de que exista alguna interferencia eléctrica y de lectura de señales.

**Figura 40**

*Montaje del sistema eléctrico y electrónico del manipulador*



### ***Consumo energético de los motores***

El consumo energético que generan los motores se mide por el tiempo que están en funcionamiento y la potencia que consume en ese tiempo, teniendo el consumo en vatios por hora. Para este proyecto, este dato no representa mayor importancia por el hecho de que se trata de una celda robótica didáctica y no de un manipulador que requiera unas determinadas horas de trabajo al día, pero por motivos informativos se realizará el cálculo aproximado.

Para el motor de la junta prismática que tiene un consumo aproximado de 10W, se toma en cuenta que el funcionamiento de los actuadores es intermitente dentro de las horas de funcionamiento dirías que podría tener la celda, se podría decir que su consumo energético es de 60Wh por día, y de los motores de las juntas rotacionales que consumen 5W tendrán un consumo energético aproximado de 30Wh.

### **Módulo 3. Programación de la Celda**

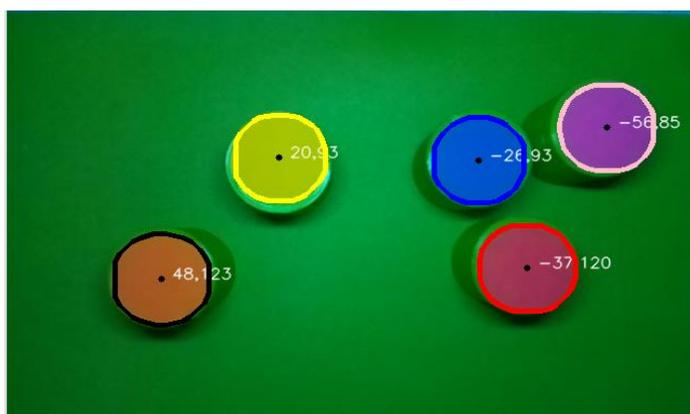
#### ***Sistema Visión artificial***

Para el sistema de visión artificial se realiza una calibración en los colores, determinando el rango de máximos y mínimos de cada color en la escala HSV que junto con la iluminación adecuada se logra reconocer el color de la etiqueta (Figura 41), de la misma manera en la imagen se realiza un reconocimiento de contornos para determinar los centros de masa de las figuras reconocidas de cada elemento que se va a clasificar y así tener la posición de cada uno de ellos por color.

Como se puede observar en la Figura 41, de igual forma, la mejora de la imagen con una buena iluminación y un fondo adecuado es evidente.

**Figura 41**

*Prueba de reconocimiento de colores*



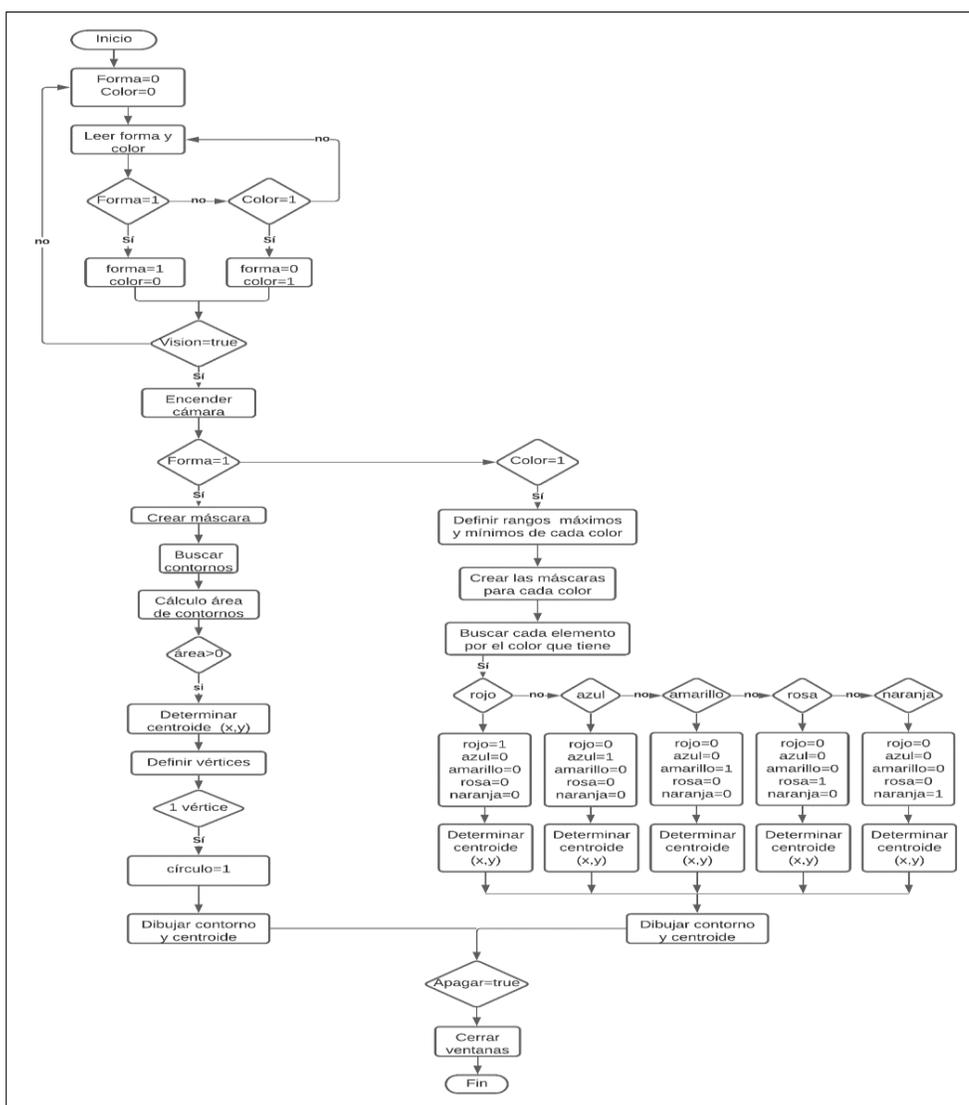
## Algoritmo visión artificial

El sistema debe reconocer los elementos que se encuentran en el área de trabajo y realizar la clasificación y acciones con cada uno de ellos de acuerdo a los criterios que se presenten, la única forma que reconoce el sistema es un círculo y los colores que reconoce el sistema son: rojo, amarillo, naranja, rosa y azul.

Después de reconocer el color de los objetos el algoritmo determina el centro de masa de cada uno de ellos obteniendo las coordenadas de la posición que será enviado al planificador como se muestra en el algoritmo del flujograma (Figura 42).

**Figura 42**

*Flujograma del sistema de visión artificial.*



### ***Integración del planificador STRIPS***

El control inteligente de acciones de la celda robótica lo realiza el planificador STRIPS, todos los sistemas de la celda aportan la información y en control de cada parte de la celda para poder generar y resolver un problema de planificación.

Programar en ROS nos da la ventaja de que la tarjeta controladora realice acciones simultaneas, programando por nodos nos permite que un nodo sea el encargado de recibir la información del estado inicial y objetivo del sistema, por medio de una HMI como se verá más adelante, y genere el problema descrito con las acciones que puede realizar el manipulador para cambiar los estados. (Figura 43). Como se ve en la figura cada acción requiere de condiciones que se deben cumplir y consecuencias de la acción sobre el mundo, como se vio en la teoría. En las acciones cada bloque está representado por la variable 'Bx' y los 3 espacios para ordenar los elementos como 'A', 'B' y 'C'.

**Figura 43**

*Acciones del sistema descritas en STRIPS*

```

Actions:
MoveraA(Bx)
Preconditions: Vacio(A), Libre(Bx), EnMesa(Bx)
Postconditions: EnA(Bx), !Vacio(A), !EnMesa(Bx)
MoveraB(Bx)
Preconditions: Vacio(B), Libre(Bx), EnMesa(Bx)
Postconditions: EnB(Bx), !Vacio(B), !EnMesa(Bx)
MoveraC(Bx)
Preconditions: Vacio(C), Libre(Bx), EnMesa(Bx)
Postconditions: EnC(Bx), !Vacio(C), !EnMesa(Bx)
ApilarDA(Bx,By)
Preconditions: Libre(By), EnA(By), Libre(Bx), EnMesa(Bx)
Postconditions: OnA(Bx,By), Libre(Bx), !EnMesa(Bx), !Libre(By), EnA(Bx), !EnMesa(Bx)
ApilarDB(Bx,By)
Preconditions: Libre(By), EnB(By), Libre(Bx), EnMesa(Bx)
Postconditions: OnB(Bx,By), Libre(Bx), !EnMesa(Bx), !Libre(By), EnB(Bx), !EnMesa(Bx)
ApilarDC(Bx,By)
Preconditions: Libre(By), EnC(By), Libre(Bx), EnMesa(Bx)
Postconditions: OnC(Bx,By), Libre(Bx), !EnMesa(Bx), !Libre(By), EnC(Bx), !EnMesa(Bx)
ApilarDMesa(Bx,By)
Preconditions: Libre(By), EnMesa(By), Libre(Bx), EnMesa(Bx)
Postconditions: OnMesa(Bx,By), Libre(Bx), !Libre(By)
ApilarTA(Bx,By,Bz)
Preconditions: OnA(By,Bz), EnA(Bz), Libre(By), EnMesa(Bx)
Postconditions: OnA(Bx,By,Bz), Libre(Bx), !Libre(By), !EnMesa(Bx)
ApilarTB(Bx,By,Bz)
Preconditions: OnB(By,Bz), EnB(Bz), Libre(By), EnMesa(Bx)
Postconditions: OnB(Bx,By,Bz), Libre(Bx), !Libre(By), !EnMesa(Bx)
ApilarTC(Bx,By,Bz)
Preconditions: OnC(By,Bz), EnC(Bz), Libre(By), EnMesa(Bx)
Postconditions: OnC(Bx,By,Bz), Libre(Bx), !EnMesa(Bx), !Libre(By)
MoveraMesa(Bx,By)
Preconditions: Libre(Bx), Up(Bx), On(Bx,By), EnMesa(By)
Postconditions: EnMesa(Bx), !Up(Bx), Libre(By), !On(Bx,By)

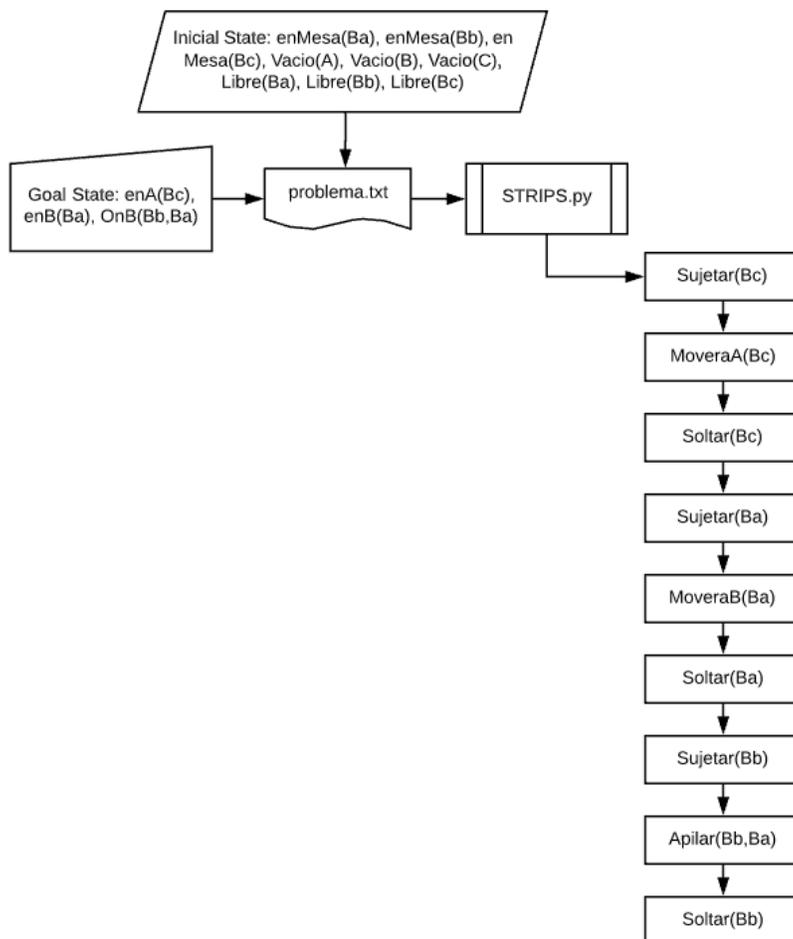
```

Este archivo de texto debe incluir el estado inicial y final que es necesario que el usuario ingrese, usando la HMI con las opciones que presta, que después utilizara el solucionar ocupado

para generar el plan de acciones, pasando así, al gestor de acciones, a continuación, en la Figura 44, se muestra un ejemplo diagramado de la resolución de un problema de planificación:

**Figura 44**

*Ejemplo de planificación*



El planificador genera la secuencia de acciones para llegar al estado final, pero estas acciones, aun requieren de procesamiento para transformarlas en acciones más pequeñas o simples que debe realizar el manipulador para generar el cambio del estado. El sistema requiere aun de más nodos que interpreten y traduzcan las acciones.

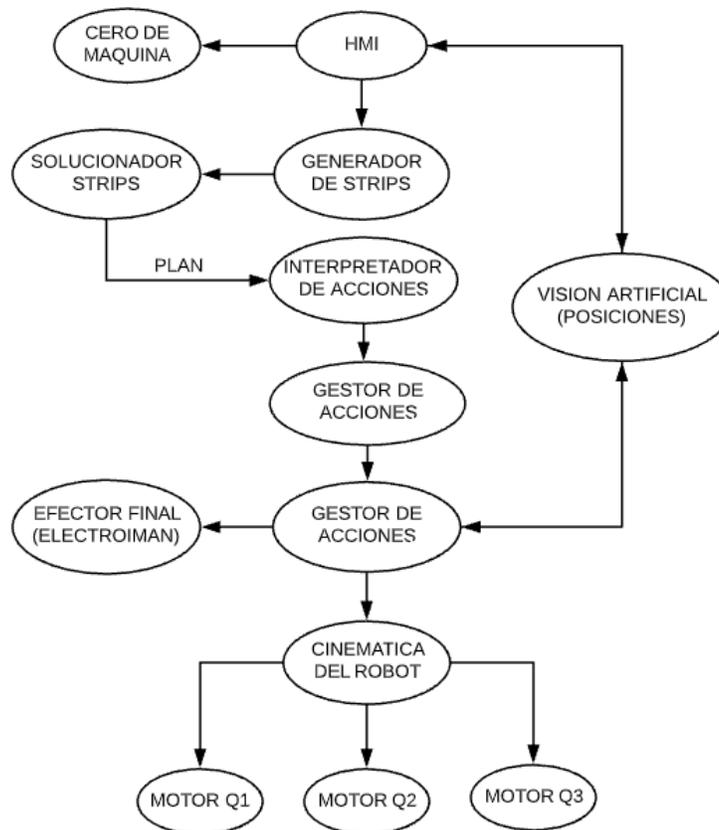
### ***Programación de la Celda Robótica***

La programación e integración de todos los sistemas de la celda robótica se lo realizó mediante programación en ROS en lenguaje Python, ROS permite y la celda requiere la programación por nodos y la comunicación entre nodos se lo realiza mediante la característica de publicador y

suscriptor de cada nodo. Cada sistema de la celda está representado por un nodo como se ve en la Figura 45, el nodo 'HMI' es el nodo principal desde el que se generan las órdenes hacia los demás nodos, desde este nodo el usuario puede escoger las opciones de funcionamiento y generar un problema para ordenar los elementos como se verá en el apartado de la interfaz gráfica. Una vez que se escoge el modo y se han ingresado las condiciones iniciales y finales del sistema el nodo 'HMI' envía estos datos al nodo 'Generador' se genera el archivo de texto del problema actual y este es con el que trabaja el nodo de 'Solucionador STRIPS', el solucionador genera una lista de acciones o el plan que requiere de una interpretación para traducirlo a acciones que puede ejecutar el manipulador y esto lo hace el nodo 'Interpretador', ahora si las acciones se dividieron en sub-acciones o pasos que debe realizar el manipulador y para esto el nodo 'Gestor de acciones' se comunica con el nodo que controla los movimientos de cada motor 'Cinemática' que a su vez hace que cada nodo de cada motor y del efector posicione, sujete o suelte cada elemento.

**Figura 45**

*Diagrama de los nodos de la Celda Robótica.*



**Programación HMI**

Para la programación de la interfaz gráfica se toma en cuenta los criterios de la guía GEDIS, se realiza en un programa que sea compatible con Python y que se pueda conectar con ROS, que detalla los criterios de interés. En la Figura 46 se puede observar la navegación entre las distintas pantallas de la interfaz. En la pantalla principal nos encontramos con dos menús, que permite que la celda opere resolviendo un problema de planificación o de modo libre ubicando para cambiar la posición del robot manualmente (Figura 47).

**Figura 46**

*Navegación entre pantallas de la Interfaz*

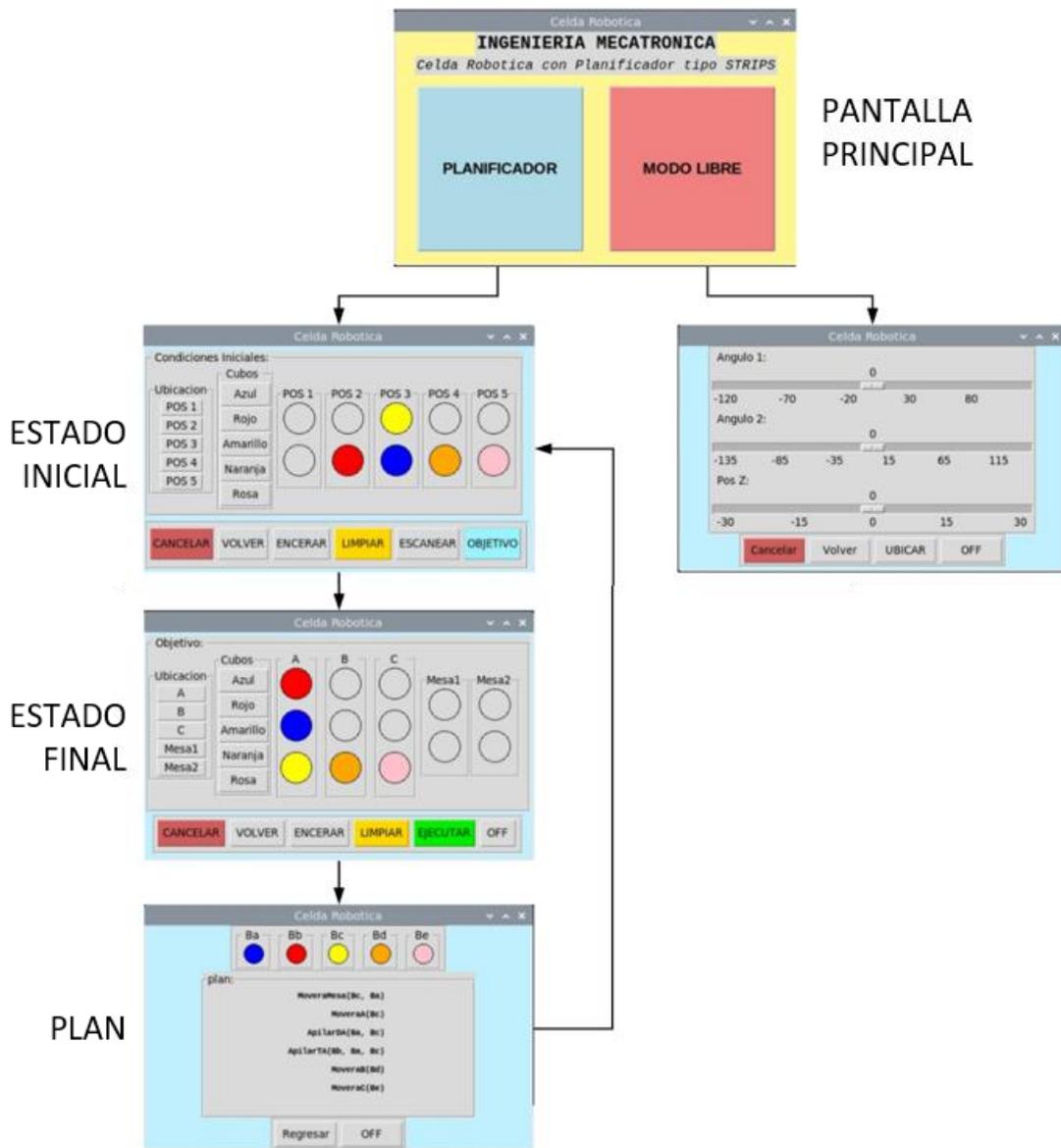


Figura 47

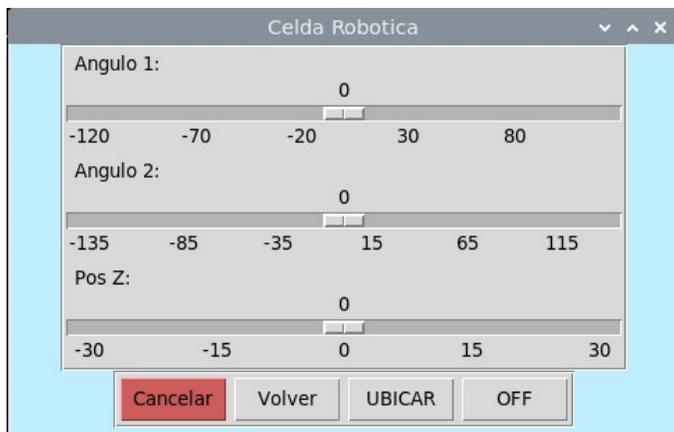
Pantalla principal.



El modo libre lleva al robot a un estado de cero de máquina que le permite mediante sliders ingresar el Angulo de cada junta rotacional que queremos que llegue el manipulador y la altura, se tienen valores máximos y mínimos que se permiten ingresar para el robot se posicione como se ve en la Figura 48, la interfaz permite salir del modo con el botón 'Volver' y hacer que el manipulador llegue a la ubicación ingresada con el botón 'Ubicar'.

Figura 48

Pantalla Modo Libre.



Las condiciones iniciales, es decir, el estado en el cual se encuentran los elementos a ordenar se ingresa mediante la pantalla de 'Condiciones Iniciales' en la cual mediante los botones etiquetados con cada color y posición en la mesa podemos ingresar manualmente las condiciones

de cada cilindro. La interfaz nos da la opción de ‘Escanear’ mediante la cámara de la celda los cilindros que puede se pueden identificar, pero si debajo de un cilindro se encuentra otro se lo debe especificar manualmente. El botón ‘Limpiar’ nos permite corregir algún dato mal ingresado borrando todos los datos actuales. Una vez ingresadas correctamente las condiciones iniciales se oprime el botón de ‘Objetivo’ que verifica que los cubos ingresados se encuentren sobre la mesa y nos lleva a la siguiente pantalla para el ingreso del estado objetivo.

**Figura 49**

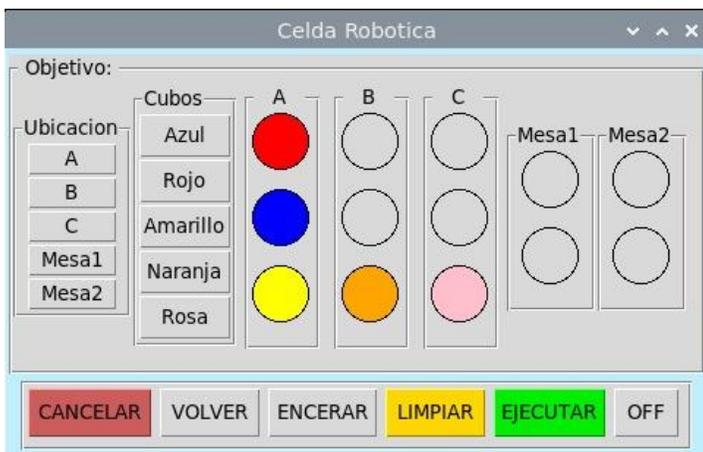
*Interfaz gráfica: ingreso de las condiciones iniciales.*



En la pantalla de ingreso del objetivo (Figura 50), de la misma manera se representan los cilindros del color y en qué espacio de clasificación se requiere que estén y se ingresan los datos mediante los botones del lado izquierdo. De la misma manera se tienen botones para ‘Encerar’, ‘Limpiar’ los datos ingresados y ‘Ejecutar’ para que el manipulador ejecute el plan.

**Figura 50**

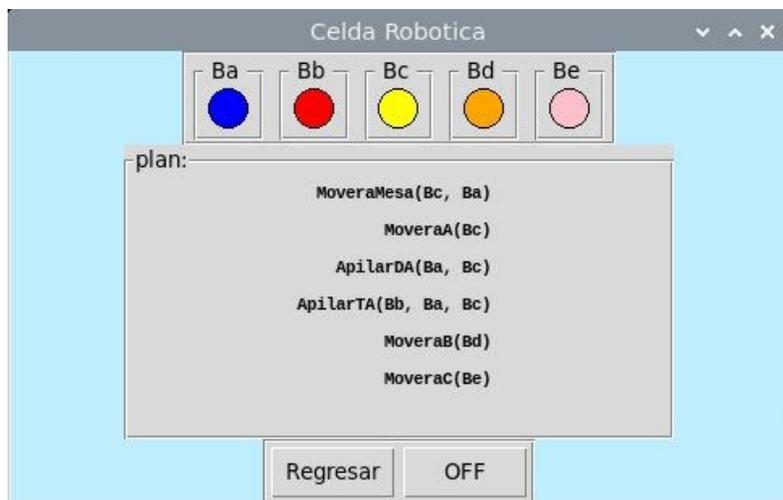
*Interfaz gráfica: ingreso del objetivo del sistema.*



En la pantalla de plan (Figura 51), se muestra el plan generado con STRIPS, es decir, las acciones que va a realizar el manipulador para llegar al objetivo. La representación de cada color en letras se visualiza en la parte superior de la pantalla.

**Figura 51**

*Interfaz gráfica: Plan elaborado con STRIPS.*



#### **Módulo 4. Pruebas de Funcionamiento**

Las pruebas de funcionamiento en robots se realizan para considerar un aumento de la productividad y calidad en los productos a desarrollarse, en las funciones que cumplen, etc., los parámetros que se deben tomar en cuenta para determinar la calidad y el buen funcionamiento de un robot son la precisión y repetibilidad. Con lo cual también se puede calibrar o mejorar la celda, en este caso, para el funcionamiento adecuado de la misma.

##### ***Prueba de Precisión Manipulador***

La precisión según (Universidad Politécnica de Madrid, 2017) es la distancia que existe entre el punto que se programa, en este caso una posición teórica, y el valor medio de los puntos a los que llegaron realmente, al repetir varias veces el movimiento del manipulador bajo las mismas condiciones de temperatura y carga nominal.

Tomando en cuenta que, los motores a paso no tienen una precisión exacta de movimiento, para la aplicación de este proyecto se puede considerar un error máximo de 3 mm en la posición.

Se muestra la posición (x,y) teórica que se dió al manipulador para que se posicione en dicho punto, y una posición (x,y) real, que es el punto al que llegó realmente el manipulador,

medido con los sensores de posición sin el planificador, obteniendo los valores de las variables articulares (Tabla 44).

**Tabla 40**

*Prueba de precisión de posicionamiento del robot sin planificador.*

Prueba	Posición teórica (x,y) mm	Posición real (x,y) mm	Error absoluto
1	(70,120)	(70,121)	(0,1)
2	(10,80)	(9,82)	(1,2)
3	(65,95)	(66,93)	(1,2)
4	(50,130)	(52,130)	(2,0)
5	(-30,70)	(-29,70)	(1,0)
6	(20,100)	(20,102)	(0,2)
7	(-35,150)	(-34,150)	(1,0)
8	(-48,110)	(-48,109)	(0,1)
9	(-25,135)	(-26,134)	(1,1)
10	(-80,180)	(-82,179)	(2,1)
<b>Promedio</b>			(0.9,1)

Como se puede apreciar en la tabla anterior, al final se obtiene un error promedio de la medida de la posición de  $\pm 0.95$  mm, en la coordenada x:  $\pm 0.9$  mm y en la coordenada y:  $\pm 1$  mm, un error aceptable para el posicionamiento del manipulador.

#### ***Prueba de Repetibilidad del Manipulador***

La repetibilidad según (Madera Coronel & López Parra, 2007), es la medida que el robot regresa a una misma posición y orientación, para este proyecto solo se necesita prueba de posicionamiento más no de orientación, para lo cual se tomó 10 mediciones de posicionamiento de un mismo punto sin planificador (Tabla 45).

**Tabla 41**

*Prueba de repetibilidad de la posición del robot sin planificador.*

Prueba	Posición teórica (x,y) mm	Posición real (x,y) mm	Error absoluto
1	(85,120)	(85,121)	(0,1)
2	(85,120)	(85,120)	(0,0)

3	(85,120)	(85,122)	(0,2)
4	(85,120)	(84,120)	(1,0)
5	(85,120)	(84,121)	(1,1)
6	(85,120)	(83,120)	(2,0)
7	(85,120)	(85,121)	(0,1)
8	(85,120)	(86,120)	(1,0)
9	(85,120)	(85,121)	(0,1)
10	(85,120)	(86,120)	(1,0)
<b>Promedio</b>			(0,6,0.6)

Como se puede apreciar en la tabla anterior, al final se obtiene un error promedio de la medida de la posición de  $\pm 0.6$  mm, en la coordenada x:  $\pm 0.6$  mm y en la coordenada y:  $\pm 0.6$  mm, un error aceptable de repetibilidad de posición.

### ***Prueba de repetibilidad de la Celda de Trabajo***

Para la prueba de repetibilidad de la celda robótica se toman 30 mediciones con el planificador, se realiza el mismo plan (Figura 52) para cada medición, y se determina si se realizó el plan correctamente (Tabla 46).

### **Figura 52**

*Plan usado para la prueba del robot con el planificador.*

```
may@ubuntu:~/STRIPS$ python strips.py bloques.txt
Goal already solved? False
Solving...
Solved!
Plan: MoveraB(Bc) -> LiberarA(Ba) -> ApilarDB(Ba, Bc) -> LiberarC(Be) -> MoveraC(Bb) ->
ApilarDC(Bd, Bb) -> MoveraA(Be)
may@ubuntu:~/STRIPS$
```

### **Tabla 42**

*Prueba de repetibilidad del plan del robot con el planificador.*

<b>Prueba</b>	<b>Cumple el plan</b>	<b>Observaciones</b>
1	SI	57 segundos
2	SI	58 segundos
3	SI	59 segundos
4	SI	58 segundos

5	NO	Pieza caída
6	SI	57 segundos
7	SI	59 segundos
8	SI	60 segundos
9	SI	60 segundos
10	SI	60 segundos
11	SI	61 segundos
12	SI	59 segundos
13	SI	58 segundos
14	SI	59 segundos
15	SI	60 segundos
16	SI	61 segundos
17	SI	58 segundos
18	SI	57 segundos
19	SI	59 segundos
20	SI	58 segundos
21	SI	54 segundos
22	SI	59 segundos
23	SI	60 segundos
24	SI	60 segundos
25	SI	60 segundos
26	SI	58 segundos
27	SI	58 segundos
28	SI	60 segundos
29	SI	60 segundos
30	SI	59 segundos
<b>Promedio</b>		58.83 segundos

Como se observa en la tabla anterior de las 30 pruebas, una no cumplió el objetivo, debido a que se cayó una pieza antes de llegar al espacio de resolución, es decir, que el 96.67% de las pruebas cumplieron con el plan.

Los errores de repetibilidad se dan generalmente por problemas de rozamiento en los sistemas de transmisión, en este caso en el sistema de bandas y poleas, problemas con la histéresis y zona muerta de los motores, los errores de precisión, en cambio se dan por no tener una buena calibración del manipulador, también, como ya se había mencionado que los motores a pasos no son tan precisos, otra razón se debe por redondeos en los cálculos de transformación cinemática o en dimensiones reales y teóricas del robot.

## Capítulo 5

### Conclusiones, recomendaciones y trabajos futuros

El capítulo desarrolla las conclusiones a las que se llegaron tras la implementación de una celda robótica basada en un planificador de tipo “STRIPS”, así también, se redactan recomendaciones para aumentar la vida útil de la celda, finalmente, trabajos futuros que permitan a los interesados tener una pauta para un próximo proyecto.

#### Conclusiones

Se diseñó y construyó un manipulador PRR de 3 grados de libertad, basado en la cinemática y dinámica robótica para el dimensionamiento de los motores que son controlados con un procesador que asegura el movimiento de cada articulación al punto establecido.

Se implementó una celda robótica de trabajo para clasificación por colores, a través de un manipulador PRR (3 grados de libertad), la misma que consta de un planificador de tipo STRIPS para la optimización de la distribución del tiempo y llegar a cumplir el objetivo, usando para esto la visión artificial y el controlador-procesador Raspberry Pi.

Se programó la planificación basada en STRIPS, con un sistema operativo robótico (ROS), para la elaboración del plan que realiza la celda cumpliendo con la meta establecida.

Se realizó una programación de nodos en ROS en lenguaje Python, con un gestor principal de acciones, para establecer la secuencia de las mismas a ejecutarse por cada parte de la celda a través de una comunicación de mensajería publicador/subscriptor, evitando de esta manera posibles errores en el funcionamiento de cada etapa.

Se realizaron pruebas de funcionamiento sin el planificador, en base al posicionamiento, se obtuvo un error promedio de la medida de la posición de  $\pm 0.95$  mm, en la coordenada x:  $\pm 0.9$  mm y en la coordenada y:  $\pm 1$  mm, un error aceptable para el posicionamiento del manipulado

En la prueba de funcionamiento sin el planificador, en base a la repetibilidad, se registró un error promedio de la medida de la posición de  $\pm 0.6$  mm, en la coordenada x  $\pm 0.6$  mm y en la coordenada y:  $\pm 0.6$  mm, un error aceptable de repetibilidad de posición.

En la prueba de funcionamiento con el planificador, en base a la repetibilidad del plan, se determinó que el 96.67% de las pruebas si cumplieron con el plan, mientras que el 3.33% de estas no cumplió con el objetivo establecido en el plan, debido a que una pieza se soltó del efector final.

Los errores de repetibilidad se dan generalmente por problemas de rozamiento en los sistemas de transmisión, en este caso en el sistema de bandas y poleas, problemas con la histéresis y zona muerta de los motores, los errores de precisión, en cambio se dan por no tener una buena calibración del manipulador, ya que los motores a pasos no son tan precisos, y/o por redondeos en los cálculos de transformación cinemática en dimensiones reales y teóricas del manipulador.

### **Recomendaciones**

Se recomienda el desarrollo de un manipulador con otra configuración (SCARA, RRR, entre otros) y con más grados de libertad, para generar una comparación de resultados entre los tipos de robot.

Se recomienda el uso de otros lenguajes de programación que permitan llegar al objetivo con mayor optimización de recursos, más la implementación de una simulación previa en tiempo real de la ejecución del plan de la celda.

Se recomienda tomar en cuenta la variación del volumen de trabajo en la celda para mayor captación del manipulador y así obtener una capacidad de cumplir con más objetivos en menor tiempo.

Se puede considerar la alternativa del cambio en la visión artificial, para una clasificación por formas o por códigos QR que generen dinamismo en la gama de reconocimiento de acuerdo a la necesidad.

Se recomienda realizar una comparación de respuesta de la celda con el uso de varios tipos de controladores que permitan llegar al mismo objetivo.

Se recomienda realizar al menos dos pruebas con/sin el planificador para considerar un rendimiento óptimo de la celda y el manipulador, generando así un análisis de los resultados que den paso a acciones de corrección a posibles errores.

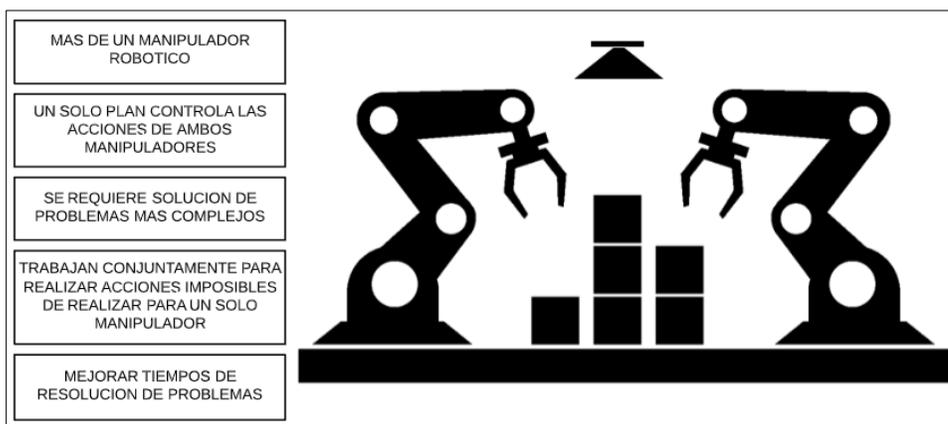
Antes de la puesta en marcha de la celda robótica, considerar las recomendaciones planteadas como medidas de seguridad en los manuales de operación y mantenimiento.

## Trabajos Futuros

Una posible adecuación futura del proyecto presentado, es la implementación de otro manipulador gemelo que mantenga la misma planificación STRIPS del manipulador original, para que la información compartida, mediante una comunicación maestro-esclavo, produzca una solución más óptima en cuestión de tiempo de resolución y de ejecución.

**Figura 53**

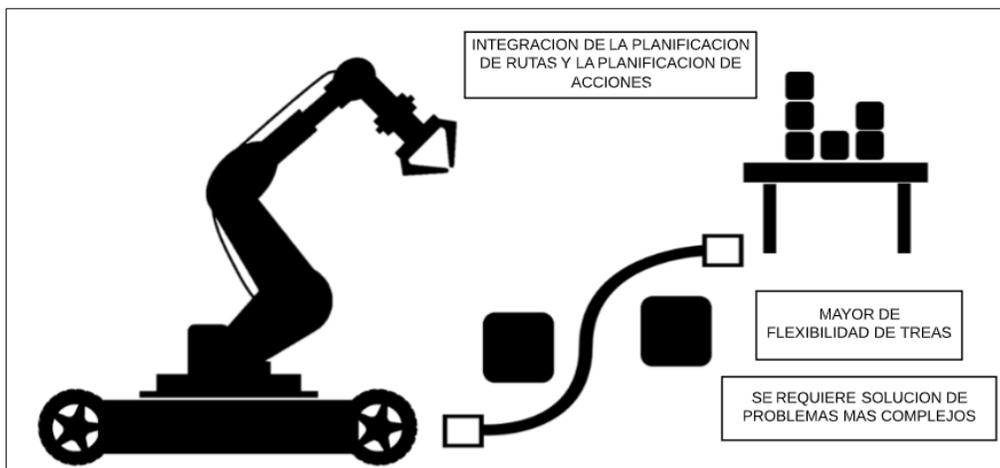
*Esquema de trabajo conjunto de manipuladores con Planificación*



Otra investigación futura es la implementación de la planificación STRIPS, en un robot móvil que genere dinamismo en la ejecución de tareas, donde exista la interacción de un punto central a un punto objetivo mediante la integración de la planificación de trayectorias y planificación de tareas del mismo.

**Figura 54**

*Esquema de integración de planificaciones en un Robot móvil.*



### Bibliografía

- Alcero. (2007). *Plotter Router Fresadora CNC: cálculo del par motor en sistema lineal*. Alcero.
- AliExpress: CIOOT Official Store. (30 de Agosto de 2016). Stepper motor NEMA17 42BYGH34 . Estados Unidos.
- AliExpress: Myers Store. (09 de Enero de 2019). Motor paso a paso HYBRID OK42HB40-1204A-TJ750. Estados Unidos.
- APM. (2020). Cámara 5MP para Raspberry Pi 4. Ecuador.
- APM. (2020). Raspberry pi 4 Model B 2GB RAM. Ecuador.
- Arias, F., Jiménez, J., & Ovalle, D. (2009). *Modelo de planificación instruccional en sistemas tutoriales inteligentes* . Medellín: Revista Avances en Sistemas e Informática, Vol.6(No.1),155-164.
- Berzal, F. (2013). *Planificación*. Granada: Universidad de Granada.
- Budynas, R., & Nisbett, K. (2008). *Diseño en ingeniería mecánica de Shigley*. México D.F.: McGraw-Hill 8th Edición .
- Castillo, L., Fernández, J., & González, A. (2013). *Diseño Interactivo de Programas de Control Industrial Basado en Técnicas de Inteligencia Artificial*. Granada: Universidad de Granada.
- Compañía Levantina de Reductores . (2019). *Motor paso a paso: cuándo y por qué utilizarlo*. Alicante: Compañía Levantina de Reductores .
- Córdova Arias, A. F., & Quimbiamba Ayol, W. P. (2018). *Diseño, control e implementación de una celda robotizada para clasificación de productos mediante el trabajo colaborativo de dos robots scara*. Sangolquí : Universidad de las Fuerzas Armadas ESPE.
- E.T.S. de Ingenieros Industriales. (2016). *Morfología del robot: estructura mecánica*. Valladolid: Universidad de Valladolid.
- Eckroth, J. (2015). PDDL. *PDDL*. Creative Commons Attribution-ShareAlike 3.0 Unported License.
- Electronics. (2020). Pantalla 3.5" para Raspberry pi4. Ecuador.

- García, V. (05 de Marzo de 2020). *Electrónica Práctica Aplicada: Descripción del driver A4988*. España.
- Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated Planning Theory and Practice*. Trento: Elsevier.
- González, A. (2014). *Quality Function Deployment: Una Herramienta para establecer los requerimientos técnicos de un edificio en México*. México D.F. : Universidad Politécnica de Valencia .
- González, E. A. (2015). *Desarrollo de un algoritmo planificador de rutas con capacidad de implementación en diversas aplicaciones de la robótica móvil*. Pereira: Universidad Tecnológica de Pereira.
- Guzmán , J., & Gutiérrez, A. (2012). *Planificación abstracta en los sistemas de robots autónomos*. Medellín: Revista Colombiana de Tecnologías Avanzadas Vol.2 No.20.
- Guzmán, J. (2003). Técnicas de planificación para la generación automática de programas de control. *Revista Ingeniería UDistrital Vol. 8 No.2, 25-31*.
- Guzmán, J. (2012). *Aplicación del planificador STRIPS para la construcción de un programa de control*. Medellín: Universidad Nacional de Colombia.
- Guzmán, J. (2015). *Aplicación del planificador STRIPS para la construcción de un programa de control*. Medellín: Revista científica UDistrital.
- Guzmán, J., & Gutiérrez , A. (2012). *Planificación abstracta en los sistemas de robots autónomos* . Medellín: Revista Colombiana de Tecnologías de Avanzada .
- KALARM. (2018). *Correa de distribución* . Estados Unidos .
- KALARM. (2018). *Polea GT2 de aluminio* . Estados Unidos.
- Kuipers, B., Feigenbaum, E., Hart, P., & Nilsson, N. (2017). *Shakey: From Conception to History*. AI Magazine.
- Lagos, C., Buitrago, J., & Timarán, R. (2010). *Metodología para el desarrollo de un planificador de tareas para una celda flexible de manufactura utilizando búsqueda heurística* . Pasto: Revista Colombiana de Tecnologías Avanzadas .

- Megatronica. (2020). Final de carrera . Quito , Ecuador .
- MGSYSTEM. (2020). Electroiman 12vdc 100N . Quito, Ecuador.
- MOONS´moving in better ways. (2018). *NEMA 17 Standard Hybrid Stepper Motors*. Shanghai: MOONS´moving in better ways.
- Moya, M., & Vaucheret, C. (2008). *Planificador continuo como controlador de agentes robots*. Neuquén : Universidad Nacional del Comahue.
- Moya, M., & Vaucheret, C. (2008). *Planificador continuo como controlador de agentes robots*. Buenos Aires: Universidad Nacional del Comahue .
- Nava , N., & Ceccarelli, M. (2005). Región posible para el espacio de trabajo de un manipulador genérico de tres pares de revolución . *Revista Iberoamericana de Ingeniería Mecánica Vol.11, N° 2, 67-80*.
- Peña , M., López, I., Corona, J., & Ordaz , K. (2003). *Visión para Robots en Tareas de Ensamble*. México: Research Gate.
- Rainer , C., Muñoz , R., & Hernández, Á. (2017). *Manufactura Aditiva*. San Salvador: Realidad y Reflexión N°43.
- Rey, A., & Hernández, G. (1991). *Robot*. Bogotá: Sección Publicaciones SENA.
- Rodas Benalcázar, A. V. (1999). *Control desde un PC de la pinza de un brazo de robot*. Quito: Escuela Politécnica Nacional: Facultad de Ingeniería Eléctrica.
- Russell, S., & Norvig, P. (2010). Classical Planning. En S. Russell, & P. Norvig, *Artificial Intelligence: A Modern Approach Third Edition* (págs. 366-396). New Jersey: Prentice Hall, Pearson Education, Inc.
- Sánchez, C. (2017). *Shakey, el primer robot inteligente de la historia y el abuelo del coche autónomo*. España: El diario.
- Sapiensman: Oficios Técnicos. (2020). *Componentes mecánicos en un sistema de control de movimiento* . Sapiensman.
- Sistemas Expertos e Inteligencia Artificial. Guía No. 11. (2015). *Planificación en Inteligencia Artificial. Agentes Planificadores*. Sistemas Expertos e Inteligencia Artificial.

SolidWorks. (2018). SolidWorks.

Tapia, C. (2017). *Diseño e implementación de un planificador para un agente autónomo*. Madrid: Universidad Politécnica de Madrid.

Tecnología informática. (2020). Fuente de Alimentación 12v - 5A - 60W. Quito, Ecuador.

Valenzuela, O., Moreno, V., Cedeño, Y., & Valenzuela, R. (2014). *Planificador para la navegación en la robótica móvil basada en agentes en entornos estructurados*. La Habana : Serie Científica de la Universidad de las Ciencias Informáticas Vol. 7, No. 1, Pág. 39-53.

Van Loenen, B., & Mroczkowski, M. (2010). QFD for the building and construction industry. *International Journal of Design Sciences and Technology*, 91-105.

Vassos, S. (2012). *Introduction to AI STRIPS planning*. Grecia: University of Athens.

**Anexos**