

**Escuela Superior Politécnica del Litoral**

**Facultad de Ingeniería en Mecánica y Ciencias de la Producción**

Diseño de un robot autónomo recogedor de pelotas de tenis basado  
en visión por computador

INGE-2310

**Proyecto Integrador**

Previo la obtención del Título de:

**Ingeniero en Mecatrónica**

Presentado por:

Diego Estefano Vega Barquet

Alex Ricardo Mata Restrepo

Guayaquil - Ecuador

Año: 2023

## **Dedicatoria**

---

Dedico este proyecto a mis padres, Dick Vega y Mercedes Barquet, a mis hermanos Estefanía Vega y Ismael Vega, quienes me brindaron los valores y ética necesaria para llegar a este punto, además de su constante apoyo a lo largo de toda mi carrera universitaria

**Diego Estefano Vega Barquet**

## Dedicatoria

---

Dedico este proyecto a mis padres, Alexandro Mata y Elizabeth Restrepo, a mi abuela Rosa Restrepo, a mi hermano Xavier Mata. Gracias por creer en mí y por ser pilares fundamentales en mi camino hacia el éxito.

**Alex Ricardo Mata Restrepo**

## Agradecimientos

---

Agradezco a mis padres, Dick Vega y Mercedes Barquet, por siempre apoyarme en mis decisiones y darme la oportunidad de poder centrarme en mis estudios sin preocupación. Por haberme formado en la persona que soy actualmente y por su paciencia en todo momento

Agradezco a mis hermanos Ismael y Estefanía por siempre estar para mí en cualquier circunstancia y motivarme a seguir adelante.

A mis amigos más cercanos Xavier, Brian, Gianni, Rommyna y Tal sin ellos esto no sería posible.

Agradezco a mis modelos a seguir Cristiano Ronaldo y J.Cole , que siempre me enseñaron que nada es imposible si uno trabaja duro y da todo de sí mismo en su disciplina.

**Diego Estefano Vega Barquet**

## Agradecimientos

---

Quiero expresar mi profundo agradecimiento a Dios, por guiarme en todo este largo camino y permitirme culminar mi carrera universitaria. A mis padres, Alexandro Mata y Elizabeth Restrepo, cuyo apoyo incondicional ha sido mi mayor fortaleza a lo largo de esta travesía. Gracias por estar a mi lado, por alentarme en los momentos desafiantes y celebrar conmigo cada logro. A mi abuela Rosa Restrepo, quiero dedicarle un agradecimiento especial por su rol fundamental durante toda mi vida con su amor y apoyo, desde mi infancia hasta este momento donde logré cumplir uno de mis objetivos más importantes. Adicionalmente, quiero reconocer a mi hermano, Xavier Mata, por también formar parte de este proceso. Este logro no habría sido posible sin la combinación única de amor, apoyo y enseñanzas provenientes de mi familia.

**Alex Ricardo Mata Restrepo**

## Declaración Expresa

---

Nosotros Diego Estefano Vega Barquet y Alex Ricardo Mata Restrepo acordamos y reconocemos que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores. La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique a los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 30 de enero del 2024.



---

Diego Estefano Vega  
Barquet



---

Alex Ricardo Mata Restrepo

## **Evaluadores**

---

---

**Carlos Saldarriaga, Ph.D.**

Profesor de Materia

---

**Efraín Terán, M.Sc.**

Tutor de proyecto

## Resumen

En respuesta al creciente interés en la robótica y sus diversas aplicaciones, se destaca su presencia en el ámbito deportivo. En particular, en el tenis, la tarea de recoger las pelotas dispersas durante las prácticas se lleva a cabo manualmente, afectando la dinámica del entrenamiento y generando pérdida de tiempo y concentración para los jugadores. Este proyecto se propone abordar esta problemática mediante el diseño de un robot autónomo con un sistema de detección basado en visión por computadora.

La fase de desarrollo, realizada en Inventor 2023, incluyó la selección de materiales, como PLA y acrílico, priorizando la facilidad de transporte según los requisitos del cliente. La implementación se basó en una Raspberry Pi 3B para controlar sensores y actuadores, y se programó un algoritmo de visión por computadora utilizando la librería OpenCV en Python. Se incorporó un sistema de reconocimiento de pelotas y cálculo de distancias mediante redes neurales.

Los resultados obtenidos reflejaron un diseño ligero y fácil de transportar, con más de una hora de autonomía y un costo un 76% inferior a las alternativas comerciales. En conclusión, este proyecto demuestra la viabilidad y eficiencia de la robótica en el ámbito deportivo, ofreciendo soluciones innovadoras y económicas.

**Palabras Clave:** OpenCV en Python, Robótica deportiva, Robot autónomo, Recolección de pelotas



## **Abstract**

*In response to the growing interest in robotics and its various applications, its presence in the sports field stands out. Particularly, in tennis, the task of picking up the scattered balls during practices is carried out manually, affecting the dynamics of training and generating loss of time and concentration for the players. This project aims to address this problem by designing an autonomous robot with a detection system based on computer vision.*

*The development phase, carried out in Inventor 2023, included the selection of materials, such as PLA and acrylic, prioritizing the ease of transport according to customer requirements. The implementation was based on a Raspberry Pi 3B to control sensors and actuators, and a computer vision algorithm was programmed using the OpenCV library in Python. A ball recognition and distance calculation system were incorporated using neural networks.*

*The results obtained showed a lightweight and easy to transport design, with more than one hour of autonomy and a cost 76% lower than commercial alternatives. In conclusion, this project demonstrates the viability and efficiency of robotics in the sports field, offering innovative and economical solutions.*

**Keywords:** *OpenCV in Python, Sports Robotics, Autonomous Robot, Ball Retrieval*

## Índice general

Resumen.....	I
Abstract.....	II
Abreviaturas.....	VI
Simbología.....	VII
Índice de figuras.....	VIII
Índice de tablas.....	XI
Índice de planos.....	XII
Capítulo 1.....	1
1.    Introducción.....	2
1.1    Descripción del problema.....	5
1.2    Justificación del problema.....	6
1.3    Objetivos.....	7
1.3.1    Objetivo general.....	7
1.3.2    Objetivos específicos.....	7
1.4    Marco Teórico.....	7
1.4.1    Estado de arte (Robots en el deporte).....	7
1.4.2    Configuraciones.....	11
1.4.2.1    Ackerman.....	11
1.4.2.2    Triciclo clásico.....	12
1.4.2.3    Direccionamiento diferencial.....	13

1.4.2.4	Skid steer .....	15
1.4.2.5	Pistas de deslizamiento.....	15
1.4.3	Navegación.....	17
1.4.3.1	Mediciones de posición relativa .....	17
1.4.3.2	Medición de posición absoluta .....	18
1.4.4	Sensores para navegación por estimación.....	19
1.4.4.1	Encoder óptico incremental .....	20
1.4.4.2	Encoder óptico absoluto .....	20
1.4.5	Procesamiento de imágenes .....	21
Capítulo 2	.....	24
2.	Metodología .....	25
2.1	Requerimientos de Diseño .....	25
2.2	Selección de alternativas.....	26
2.3	Metodología de diseño.....	30
2.4	Diseño conceptual.....	32
2.5	Parámetros de diseño mecánico .....	33
2.5.1	Diseño mecánico del robot .....	33
2.5.1.1	Componentes del diseño mecánico del robot .....	33
2.5.2	Selección de motores.....	43
2.6	Parámetros de diseño electrónico.....	44

2.6.1	Selección de sensores para control del vehículo .....	44
2.7	Selección de fuente de energía.....	47
2.8	Parámetros de diseño de control .....	48
2.8.1	Control del vehículo .....	48
2.8.2	Control del mecanismo de recolección .....	50
Capítulo 3	.....	52
3. Resultados y análisis	.....	53
3.1	Diseño final del dispositivo .....	53
3.2	Funcionamiento del mecanismo de recolección .....	57
3.3	Simulación de movimiento .....	59
3.4	Simulación de detección de pelota de tenis desde robot.....	61
3.5	Simulación de detección de objetos de una cámara panorámica. ....	64
3.6	Análisis de Costo .....	68
Capítulo 4	.....	70
4. Conclusiones y recomendaciones	.....	71
4.1	Conclusiones .....	71
4.2	Recomendaciones .....	72
Referencias	.....	74
Apéndice.....	.....	78

## **Abreviaturas**

ESPOL Escuela Superior Politécnica del Litoral

D.C.L Diagrama de Cuerpo Libre

PLA Políácido láctico

3D Tres dimensiones

RPM Revoluciones por minuto

WD Wheel drive

## Simbología

mm	Milímetro
cm	Centímetro
m	Metro
mV	Milivoltios
V	Voltio
mA	Miliamperios
A	Amperio
N.m	Newton-metros
N	Newton
s	Segundos
h	Horas
MPa	Megapascal
DC	Corriente directa
kg	Kilogramos
°	Grado

## Índice de figuras

<b>Figura 1.1</b> Uso de la tecnología visual en el tenis .....	2
<b>Figura 1.2</b> Medición de un lado de la cancha de tenis .....	3
<b>Figura 1.3</b> Pelota de tenis en la cancha .....	4
<b>Figura 1.4</b> Robot móvil de alto presupuesto de la compañía Tennibot .....	5
<b>Figura 1.5</b> Entrenamiento de tenis y todas las pelotas que se usan .....	6
<b>Figura 1.6</b> Robot usado en el deporte de ping pong .....	8
<b>Figura 1.7</b> Funcionamiento del robot.....	8
<b>Figura 1.8</b> Robot usado para encestar canasta de basquetbol .....	9
<b>Figura 1.9</b> Vista del robot DONIC Robopong 3050 XL.....	10
<b>Figura 1.10</b> Ilustración de configuración Ackerman .....	11
<b>Figura 1.11</b> Vehículo autónomo Navlab 11 .....	12
<b>Figura 1.12</b> Ilustración de configuración Triciclo clásico .....	12
<b>Figura 1.13</b> Transpaleta autónoma T-MATIC .....	13
<b>Figura 1.14</b> Ilustración de Direccionamiento diferencial .....	14
<b>Figura 1.15</b> Robot Diferencial SR1 .....	14
<b>Figura 1.16</b> Robot TERRAGATOR .....	15
<b>Figura 1.17</b> Robot AURIGA.....	16
<b>Figura 1.18</b> Ruedas mecanum omnidireccionales marca KUKA .....	16
<b>Figura 1.19</b> Modelo de sistema de odometría de robot diferencial .....	17
<b>Figura 1.20</b> Esquema simple de sistema de navegación inercial.....	18

<b>Figura 1.21</b> Encoder óptico incremental herramienta.....	20
<b>Figura 1.22</b> Encoder óptico absoluto herramienta.....	21
<b>Figura 1.23</b> Ilustración de procesamiento de imágenes.....	22
<b>Figura 1.24</b> Ilustración de proceso de procesamiento de imágenes.....	23
<b>Figura 2.1</b> Boceto de primera alternativa a la problemática.....	26
<b>Figura 2.2</b> Boceto de segunda alternativa a la problemática.....	27
<b>Figura 2.3</b> Boceto de tercera alternativa a la problemática.....	27
<b>Figura 2.4</b> Diagrama de proceso de metodología.....	32
<b>Figura 2.5</b> Ilustración de diseño en software Inventor.....	33
<b>Figura 2.6</b> Vista superior en perspectiva del ensamble del chasis del robot.....	36
<b>Figura 2.7</b> Vista inferior en perspectiva del ensamble del chasis del robot.....	36
<b>Figura 2.8</b> Vista inferior en perspectiva del chasis del robot ya ensamblado.....	37
<b>Figura 2.9</b> Vista superior en perspectiva del depósito de pelotas.....	38
<b>Figura 2.10</b> Vista inferior en perspectiva del depósito de pelotas.....	38
<b>Figura 2.11</b> Vista superior depósito ensamblado.....	39
<b>Figura 2.12</b> Vista inferior depósito ensamblado.....	39
<b>Figura 2.13</b> Vista en perspectiva de la rampa.....	40
<b>Figura 2.14</b> Vista en perspectiva de la rampa ensamblada.....	40
<b>Figura 2.15</b> Vista en perspectiva de las paredes laterales ensambladas.....	41
<b>Figura 2.16</b> Mecanismo de paletas para recolección de pelotas.....	42
<b>Figura 2.17</b> Vista isométrica del diseño completo.....	43
<b>Figura 2.18</b> Dispositivos de detección del diseño.....	49



<b>Figura 2.19</b> Funcionamiento del sistema de control .....	50
<b>Figura 3.1</b> Vista lateral del robot indicando el centro de gravedad del diseño.....	53
<b>Figura 3.2</b> Acople de caster wheel en parte posterior del depósito con pernos de sujeción.....	54
<b>Figura 3.3</b> Vista en perspectiva de la bisagra y las agarraderas. ....	55
<b>Figura 3.4</b> Vista en perspectiva de pared lateral modificada.....	56
<b>Figura 3.5</b> Vista en perspectiva del diseño final realizado. ....	56
<b>Figura 3.6</b> Vista lateral del sistema de recolección .....	57
<b>Figura 3.7</b> Vista lateral, sistema de recolección golpeando la pelota de tenis.....	58
<b>Figura 3.8</b> Vista lateral pelota de tenis subiendo la rampa .....	58
<b>Figura 3.9</b> Vista lateral pelota ingresando al depósito.....	59
<b>Figura 3.10</b> Vista de herramienta rviz .....	60
<b>Figura 3.11</b> Vista de robot móvil simulado en Gazebo .....	61
<b>Figura 3.12</b> Vista de funcionamiento de código de detección de color por visión por computador desde la Raspberry .....	62
<b>Figura 3.13</b> Identificación de pelotas en una cancha de tenis utilizando visión por computadora .....	63
<b>Figura 3.14</b> Entrenamiento en Google Colab .....	64
<b>Figura 3.15</b> Funcionamiento de primera fase del código.....	65
<b>Figura 3.16</b> Funcionamiento del código con distancias.....	66
<b>Figura 3.17</b> Funcionamiento del código con medidas de la cancha y distancias en metros .....	67

## Índice de tablas

<b>Tabla 2.1</b> Requerimientos del diseño .....	25
<b>Tabla 2.2</b> Criterios y sus relevancias .....	29
<b>Tabla 2.3</b> Matriz de decisión para seleccionar la mejor alternativa.....	30
<b>Tabla 2.4</b> Peso de componentes del diseño.....	34
<b>Tabla 2.5</b> Características de motores de chasis .....	44
<b>Tabla 2.6</b> Características de sensor ultrasónico .....	45
<b>Tabla 2.7</b> Características de sensor LM393 .....	46
<b>Tabla 2.8</b> Características de cámara.....	46
<b>Tabla 2.9</b> Características de Raspberry Pi 3B.....	47
<b>Tabla 2.10</b> Características de batería LiPo.....	48
<b>Tabla 3.1</b> Costos de inversión para el proyecto para el proyecto .....	69

## Índice de planos

- PLANO 1 Robot autónomo recolector de pelotas
- PLANO 2 Medidas de robot autónomo recolector de pelotas
- PLANO 3 Rampa de recolección
- PLANO 4 Base de Webcam
- PLANO 5 Depósito de pelotas
- PLANO 6 Barra de soporte superior
- PLANO 7 Paletas de recolección
- PLANO 8 Base superior
- PLANO 9 Soporte lateral derecho
- PLANO 10 Soporte lateral izquierdo
- PLANO 11 Tapa trasera
- PLANO 12 Soporte de motor
- PLANO 13 Acople de matrimonio

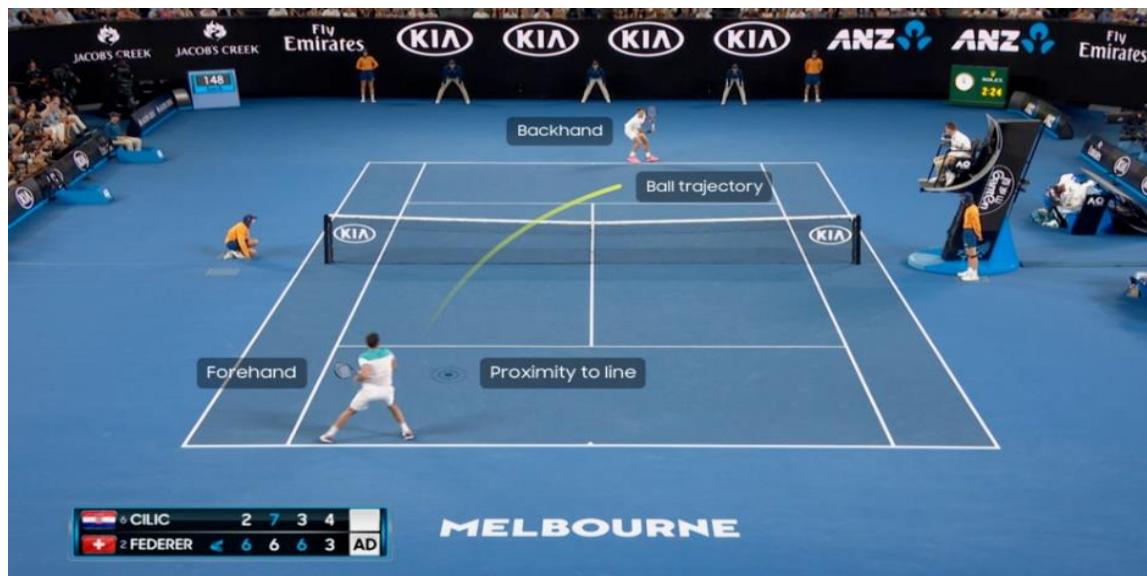
# Capítulo 1

## 1. Introducción

Hoy en día el deporte es una actividad muy importante para tanto personas jóvenes y adultas, uno de los deportes más grande alrededor del mundo es el tenis. La gente que le gusta esta actividad desempeña mucho tiempo en ella desde jugar, entrenar y ver los partidos. Existen una variedad de tecnología que ayuda en los deportes, desde tecnología visual, de transporte, entre otras.

**Figura 1.1**

*Uso de la tecnología visual en el tenis*

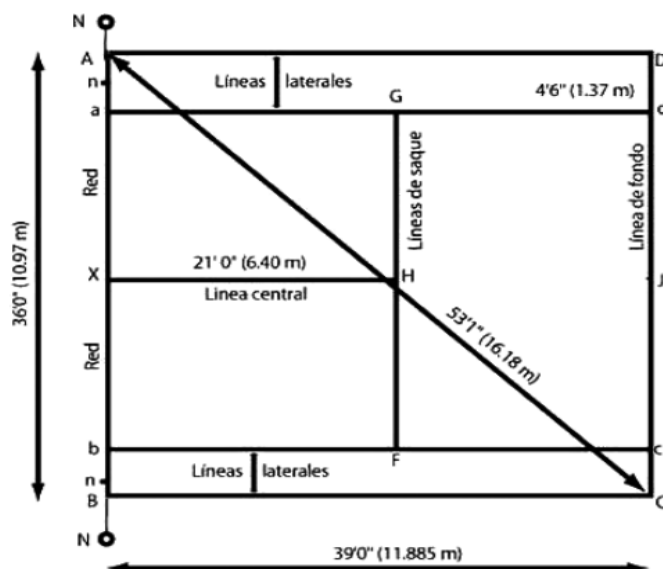


*Nota. Imagen tomada de [1].*

Lo que se busca en el deporte conocido como el tenis, donde el área de juego será un rectángulo que mida 23.77 metros de longitud y 8.23 metros de ancho. En los partidos de dobles, el ancho de la pista será de 10.97 metros. La pista estará dividida por la mitad mediante una red suspendida por una cuerda o un cable metálico. Los extremos de la red estarán fijados a la parte superior de dos postes o pasarán por encima de dos postes a una altura de 1.07 metros. [2]

**Figura 1.2**

*Medición de un lado de la cancha de tenis*



*Nota. Imagen tomada de [2].*

Se tiene una cancha ya establecida con medidas la cual es rectangular donde se juega partidos individuales o en pareja, es realizar la actividad de recolección de las pelotas de tenis, durante los entrenamientos las pelotas se esparcen en varios lugares de la cancha, debido a que se usan un gran número de estas y hacer esa acción toma tiempo al jugador o a la persona que lo está entrenando, de esta manera perdiendo la concentración en la actividad que se está realizando y haciéndola pesada.

**Figura 1.3**

*Pelota de tenis en la cancha*



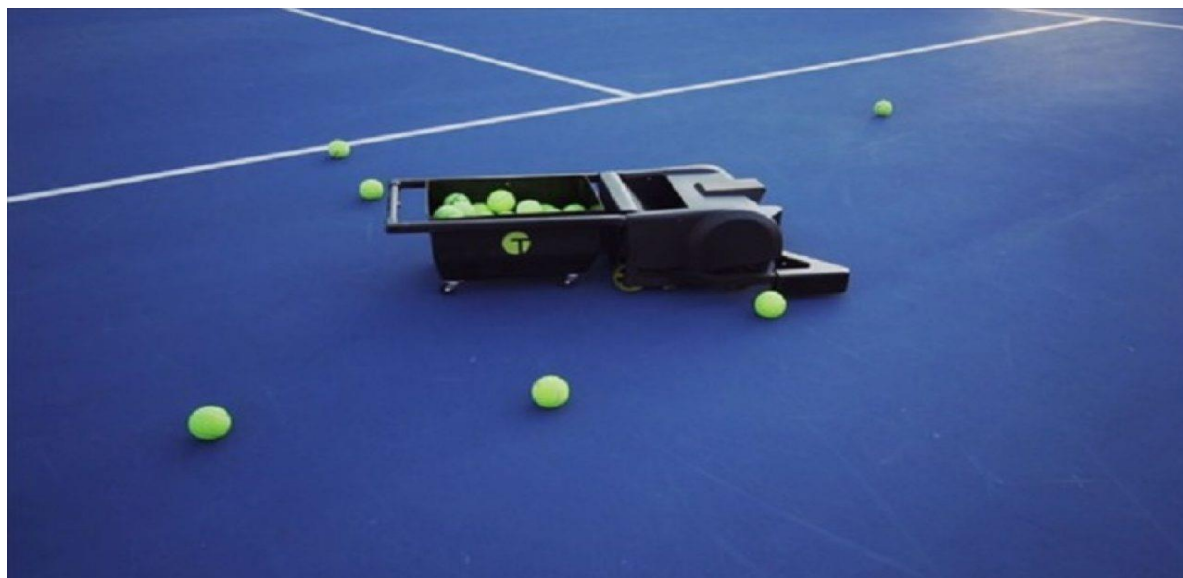
*Nota. Imagen tomada de [3].*

Los robots móviles requieren un método de locomoción específico para su aplicación, determinando su capacidad de movimiento, superficie de contacto y otras características operativas. En el diseño, es esencial considerar propiedades físicas como peso, tamaño y modo de transporte, asegurando la estabilidad durante su operación. [3]

Estos robots han encontrado aplicación en diversos campos, desde la industria y el entretenimiento hasta la seguridad, la atención médica y la investigación, incluso en formas humanoides. Su utilidad diaria abarca desde la recolección de basura hasta el transporte eficiente de materiales. Sin embargo, para tareas más complejas, como el reconocimiento de situaciones y la ejecución de comandos programados, los robots también dependen de ayudas como sensores, sistemas de reconocimiento de imágenes y procesamiento de datos.

**Figura 1.4**

*Robot móvil de alto presupuesto de la compañía Tennibot*



*Nota. Imagen tomada de [4].*

En este contexto lo que se busca en el proyecto es diseñar un robot móvil que realice de manera autónoma la recolección de pelotas durante el entrenamiento de una persona, haciendo este procedimiento más rápido que un humano.

### **1.1 Descripción del problema**

El constante avance en el ámbito de la robótica ha generado beneficios significativos en diversas esferas del mundo laboral. Sin embargo, es intrigante observar por qué este avance tecnológico no ha sido tan ampliamente adoptado en el ámbito deportivo. Esta aparente brecha puede atribuirse a varias razones, siendo dos de las principales el costo elevado de los productos robóticos y la complejidad de las tareas que se espera que los robots realicen en el contexto deportivo.

En el ámbito específico del tenis, las personas que desean entrenar enfrentan dilemas significativos. Por un lado, tienen la opción de contratar a un entrenador o recolector humano cada



vez que deseen practicar, lo que puede limitar su capacidad para entrenar con la frecuencia deseada debido a los costos asociados. Por otro lado, si eligen entrenar solos, deben invertir tiempo considerable en la recolección manual de pelotas, lo que puede ser una tarea tediosa y consume mucho tiempo, especialmente cuando se enfrentan a recuperar un número considerable de pelotas, que varían de 10 a 100 en una sesión de entrenamiento.

**Figura 1.5**

*Entrenamiento de tenis y todas las pelotas que se usan*



*Nota. Imagen tomada de [5].*

## **1.2 Justificación del problema**

La introducción de robots en deportes como el tenis podría abordar desafíos como la recolección de pelotas durante los entrenamientos, una tarea normalmente realizada por personas contratadas o por los propios jugadores. Aunque los robots diseñados para esta tarea podrían ofrecer una solución eficaz, las barreras financieras y técnicas han limitado su adopción generalizada. Estos robots podrían permitir a los jugadores practicar sin limitaciones de tiempo y costo, optimizando su rendimiento. Aunque la implementación en el ámbito deportivo es

prometedora, superar estas barreras requiere mayor innovación y colaboración entre la comunidad científica y la industria del deporte.

### **1.3 Objetivos**

#### ***1.3.1 Objetivo general***

Diseñar un robot autónomo equipado con sensores, actuadores y un sistema de programación basado en visión por computador, para la recolección automatizada de pelotas de tenis en una cancha durante las sesiones de entrenamiento

#### ***1.3.2 Objetivos específicos***

1. Desarrollar un mecanismo que realice la recolección de las pelotas de tenis de una forma rápida y efectiva.
2. Seleccionar la plataforma móvil adecuada para mover el mecanismo para recoger pelotas
3. Desarrollar un sistema de visión por computadora junto con reconocimiento de imagen y sensores, que pueda detectar la ubicación de las pelotas de tenis.

### **1.4 Marco Teórico**

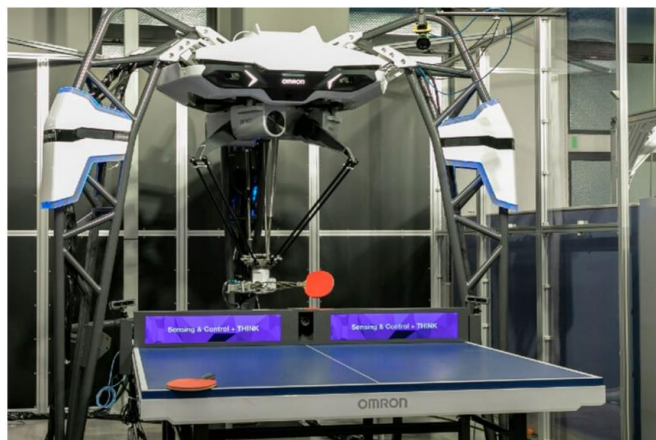
#### **1.4.1 Estado de arte (Robots en el deporte)**

Durante los últimos años ha habido múltiples avances e incursiones con el objetivo de desarrollar robots que permiten mejorar el rendimiento de atletas en diversos deportes, entre los robots más representativos e innovadores que se han desarrollado están los siguientes.

FORPHEUS desarrollado por OMRON, es un robot inteligente de ping-pong que aprende de sus rivales y los motiva para mejorar. En su séptima generación, cuenta con tres pies fijos, un módulo central aéreo que alberga la inteligencia artificial y tres brazos articulados para el movimiento y el agarre de la pala. [6]

**Figura 1.6**

*Robot usado en el deporte de ping pong*

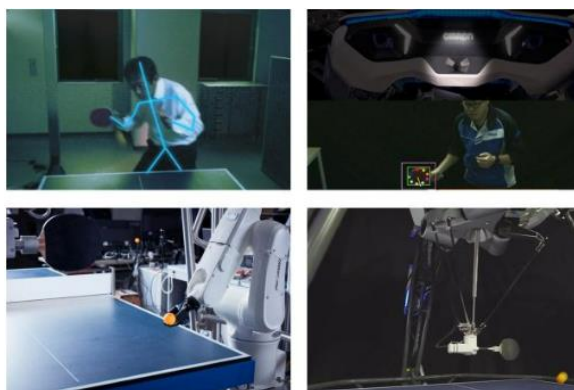


*Nota. Imagen tomada de [6].*

Este robot está equipado con sensores que miden la distancia hasta la bola y predicen su rotación y trayectoria. Dispone de tres cámaras para observar los movimientos del contrincante y la pala. Además, se le han incorporado algoritmos para mejorar su rendimiento en ambientes con poca iluminación, permitiendo a la inteligencia artificial prever los movimientos del cuerpo del oponente. [6]

**Figura 1.7**

*Funcionamiento del robot*

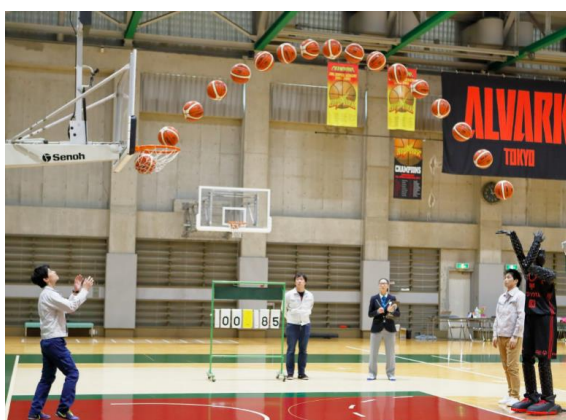


*Nota. Imagen tomada de [6].*

CUE3, desarrollado por ingenieros de Toyota en 2017, es un robot capaz de jugar baloncesto con brazos, rodillas y manos que imitan los movimientos humanos al lanzar una pelota. Equipado con ruedas para moverse en el campo, utiliza inteligencia artificial para calcular la anotación, considerando parámetros como ángulo, velocidad y distancia al aro antes de cada lanzamiento. [7]

### Figura 1.8

*Robot usado para encestar canasta de basquetbol*



*Nota. Imagen tomada de [8].*

DONIC Robopong 3050 XL es un robot de entrenamiento de ping-pong que simula a la perfección el comportamiento de jugadores de diferentes niveles. Su configuración se realiza mediante Bluetooth con la aplicación Newgy. Equipado con dos ruedas de lanzamiento y motores programados independientemente, puede simular diversas combinaciones de efectos de juego. [9]

**Figura 1.9**

*Vista del robot DONIC Robopong 3050 XL*



*Nota. Imagen tomada de [9].*

El desarrollo de robots móviles surgió para ampliar el alcance de la robótica, inicialmente limitada a estructuras mecánicas ancladas. La autonomía de estos robots se fundamenta en sistemas de navegación automática, que involucran planificación, percepción y control. Al diseñar un robot móvil, es crucial considerar la configuración, el tipo de navegación, los sensores utilizados y el sistema de visión computarizada. [10]

## 1.4.2 Configuraciones

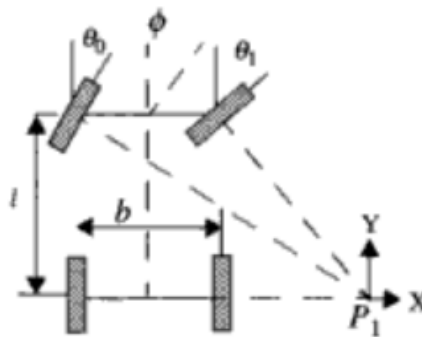
Los vehículos con ruedas son la solución más simple y eficiente para conseguir movilidad en terrenos suficientemente duros y libres de obstáculos. [10] Algunas de las configuraciones más empleadas son las descritas a continuación.

### 1.4.2.1 Ackerman

Este sistema, utilizado en vehículos de cuatro ruedas, ajusta el ángulo de la rueda delantera interior para evitar el deslizamiento. Las ruedas delanteras se cruzan en un punto sobre el eje trasero, y los puntos trazados por los centros de los neumáticos forman circunferencias concéntricas alrededor del eje de rotación. [10]

**Figura 1.10**

*Ilustración de configuración Ackerman*



*Nota. Imagen tomada de [10].*

Un ejemplo de vehículo autónomo que se ha venido desarrollando con este tipo de configuración es el NavLab el cual se ha venido empleándose en el Robotics Institute de Carnegie Mellon University desde 1986 para realizar experimentos de navegación autónoma en exteriores,

encontrando así que el principal problema de la locomoción Ackerman es la limitación de la maniobrabilidad. [10]

**Figura 1.11**

*Vehículo autónomo Navlab 11*



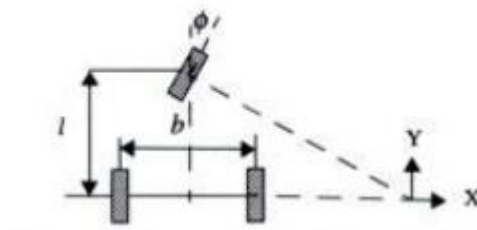
*Nota. Imagen tomada de [11].*

#### **1.4.2.2 Triciclo clásico**

Este sistema de locomoción utiliza una rueda delantera para tracción y dirección, mientras que el eje trasero, con dos ruedas laterales, es pasivo. Aunque ofrece mayor maniobrabilidad, puede tener problemas de estabilidad en terrenos difíciles, especialmente en pendientes donde el centro de gravedad puede desplazarse, resultando en pérdida de tracción. [10]

**Figura 1.12**

*Ilustración de configuración Triciclo clásico*



*Nota. Imagen tomada de [10].*

Un ejemplo de robot con esta configuración es el T-MATIC, un robot industrial desarrollado por Linde. Diseñado como un transpaleta autónomo, puede transportar pallets de hasta 3000 kilogramos y también puede ser controlado manualmente. Equipado con sensores ópticos, el T-MATIC reacciona de manera confiable ante obstáculos o movimientos en la zona de trabajo. [12]

**Figura 1.13**

*Transpaleta autónoma T-MATIC*



*Nota. Imagen tomada de [12].*

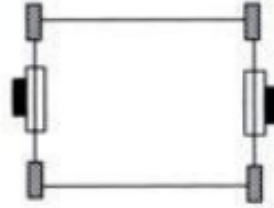
#### ***1.4.2.3 Direccionamiento diferencial***

El direccionamiento viene dado por la diferencia de velocidades de las ruedas laterales. La tracción se consigue también con estas mismas ruedas. Adicionalmente, existen una o más ruedas para soporte. Esta configuración es la más frecuente en robots para interiores. [10]



**Figura 1.14**

*Ilustración de Direccionamiento diferencial*



*Nota. Imagen tomada de [10].*

El SR1 es un ejemplo de robot diferencial didáctico con múltiples sensores, incluyendo cámara inalámbrica en su versión completa. Puede ser controlado remotamente vía radio mediante un computador. [13]

**Figura 1.15**

*Robot Diferencial SR1*



*Nota. Imagen tomada de [13].*

#### ***1.4.2.4 Skid steer***

Se disponen varias ruedas en cada lado del vehículo que actúan de forma simultánea. El movimiento es el resultado de combinar las velocidades de las ruedas de la izquierda con las de la derecha. Un ejemplo de esta configuración es el robot Terragator, desarrollado por el Robotics Institute de la Carnegie Mellon University para aplicaciones como la minería [10]

**Figura 1.16**

*Robot TERRAGATOR*



*Nota. Imagen tomada de [10].*

#### ***1.4.2.5 Pistas de deslizamiento***

Los vehículos tipo oruga utilizan pistas de deslizamiento para lograr tanto la impulsión como el direccionamiento, funcionando de manera similar a ruedas de gran diámetro. Este sistema es eficaz en terrenos irregulares, donde muestra un rendimiento destacado debido a una menor limitación por deslizamiento y una mayor resistencia al desgaste. Un ejemplo de esta locomoción es el robot AURIGA, desarrollado en la Universidad de Málaga. [10]

**Figura 1.17**

*Robot AURIGA*



*Nota. Imagen tomada de [14].*

Finalmente existen otros tipos de configuraciones como el empleo de ruedas especiales, este el caso de las denominadas “ruedas secas”, que permiten conseguir el movimiento omnidireccional de un vehículo con cuatro ruedas de este tipo adecuadamente controladas. [10]

**Figura 1.18**

*Ruedas mecanum omnidireccionales marca KUKA*



*Nota. Imagen tomada de [15].*

### 1.4.3 Navegación

De manera general los problemas de navegación de un robot móvil a través de las preguntas: “¿Dónde estoy?”, “¿A dónde estoy yendo?” y “¿Cómo debo llegar ahí?”. A pesar de no existir una única solución para el problema de navegación, las soluciones se pueden categorizar en dos grupos: mediciones de posición relativa y mediciones de posición absoluta.[16]

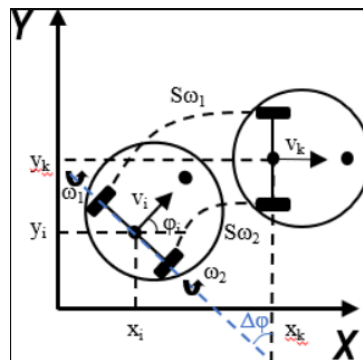
#### 1.4.3.1 Mediciones de posición relativa

##### *Odometría*

Este método utiliza encoders para medir la rotación de las ruedas y/o la orientación del volante. La odometría tiene la ventaja de ser completamente autónoma y siempre puede proporcionar una estimación de la posición del vehículo. La desventaja de la odometría es que el error de posición aumenta indefinidamente a menos que se utilice periódicamente una referencia independiente para reducir el error. [16]

**Figura 1.19**

*Modelo de sistema de odometría de robot diferencial*



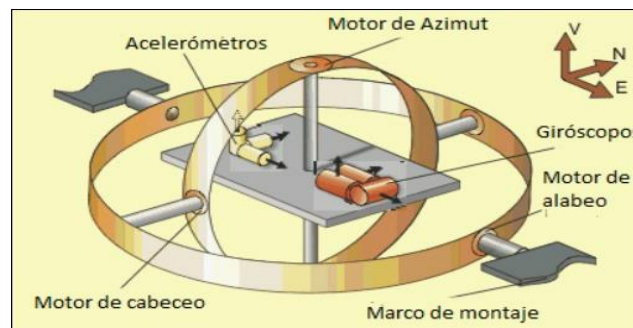
*Nota. Imagen tomada de [17].*

### ***Navegación Inercial***

Este método utiliza giroscopios y, a veces, acelerómetros para medir la velocidad de rotación y la aceleración, integrando estas mediciones para obtener la posición. Aunque los sistemas de navegación inercial son autónomos, enfrentan desafíos, como la desviación gradual de los datos con el tiempo debido a la integración, lo que limita su precisión a largo plazo. Además, el alto costo del equipo es un inconveniente. [16]

**Figura 1.20**

*Esquema simple de sistema de navegación inercial*



*Nota. Imagen tomada de [18].*

#### ***1.4.3.2 Medición de posición absoluta***

##### ***Balizas Activas***

Este método calcula la posición absoluta del robot midiendo la dirección de incidencia de tres o más balizas transmitidas activamente. Los transmisores, generalmente utilizando luz o frecuencias de radio, deben estar ubicados en sitios conocidos en el entorno. [16]

##### ***Reconocimiento de Marcas Artificiales***

En este método, se colocan marcadores artificiales distintivos en lugares conocidos en el entorno. La ventaja de las marcas artificiales es que se pueden diseñar para una detección óptima

incluso en condiciones ambientales adversas. El posicionamiento de las marcas tiene la ventaja de que los errores de posición están limitados, pero la detección de marcas externas y la fijación de la posición en tiempo real no siempre son posibles. [16]

### ***Reconocimiento de Marcas Naturales***

En este método, los puntos de referencia son características distintivas en el entorno. No es necesario preparar el entorno, pero se debe conocer el entorno de antemano. La fiabilidad de este método no es tan alta como la de las marcas artificiales. [16]

### **Coincidencia de Modelos**

Este método compara la información de los sensores del robot con un mapa del entorno. Si las características coinciden, se puede estimar la ubicación absoluta del vehículo. Los dos tipos principales de mapas utilizados son geométricos (representan el mundo en coordenadas globales) y topológicos (lo representan como una red de nodos y arcos). [16]

## **1.4.4 Sensores para navegación por estimación**

El cálculo de posición por estimación es un método matemático para determinar la ubicación actual de un vehículo extrapolando desde una posición previa mediante datos conocidos sobre rumbo y velocidad durante un período determinado. La mayoría de los robots móviles actuales se basan en esta técnica, que corrige periódicamente los errores acumulados mediante "correcciones" de varios dispositivos de navegación. [16]

La implementación más básica de este cálculo, a veces llamada odometría, implica que el desplazamiento del vehículo a lo largo de su trayectoria se deriva directamente de algún "odómetro" a bordo. Un método común de instrumentación de la odometría utiliza encoders ópticos acoplados directamente a las armaduras del motor o a los ejes de las ruedas. [16]

#### 1.4.4.1 Encoder óptico incremental

Los encoders incrementales miden la velocidad de rotación y pueden inferir la posición relativa. Si la información de posición no volátil no es crucial, los codificadores incrementales son fáciles de interconectar y ofrecen una buena resolución a un costo bajo. Un tipo simple es el codificador de tacómetro de canal único, que produce pulsos de onda sinusoidal o cuadrada para cada revolución del eje. Sin embargo, presenta inestabilidades a baja velocidad y no puede detectar la dirección de rotación, lo que limita su uso como sensor de posición. [16]

Los codificadores incrementales de fase cuadratura superan estos problemas al incorporar un segundo canal desplazado 90 grados del primero. Esto permite determinar la dirección de rotación y proporciona una mayor resolución, mejorando así su utilidad. [16]

**Figura 1.21**

*Encoder óptico incremental herramienta*



*Nota. Imagen tomada de [19].*

#### 1.4.4.2 Encoder óptico absoluto

Los codificadores absolutos suelen utilizarse en aplicaciones de rotación más lenta que requieren información de posición cuando no se puede tolerar la pérdida de referencia debido a

una interrupción de energía. Los encoders absolutos miden directamente la posición angular e infieren la velocidad. [16]

En lugar de las secuencias de bits en serie de los diseños incrementales, los codificadores ópticos absolutos proporcionan una salida de palabras en paralelo con un patrón de código único para cada posición cuantizada del eje. Los esquemas de codificación más comunes son el código Gray, el binario natural y el decimal codificado en binario. [16]

**Figura 1.22**

*Encoder óptico absoluto herramienta*



*Nota. Imagen tomada de [19].*

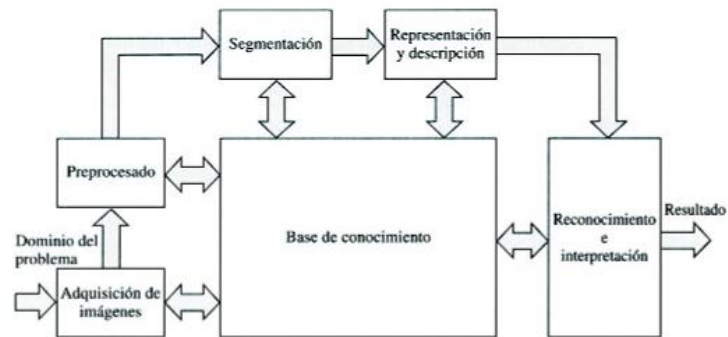
### **1.4.5 Procesamiento de imágenes**

El procesamiento de imágenes implica modificar la información visual para mejorar resultados o aislar características específicas. Una imagen digital se representa como una matriz, donde los índices indican un punto en la imagen y el valor correspondiente refleja el nivel de gris en ese punto.[20]



**Figura 1.23**

*Ilustración de procesamiento de imágenes*



*Nota. Imagen tomada de [20].*

Se puede definir la visión artificial como los procesos de obtención, caracterización e interpretación de información de imágenes tomadas de un mundo tridimensional. El proceso de procesamiento de imágenes se puede dividir en cinco etapas principales:

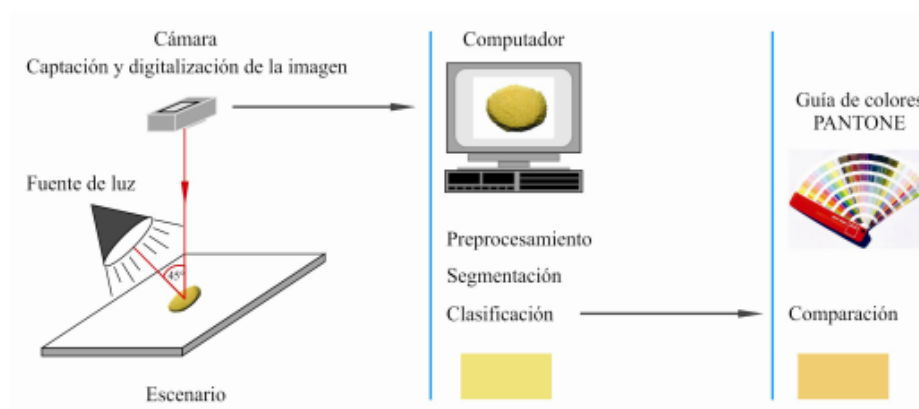
- Adquisición
- Preprocesado
- Segmentación
- Representación y descripción
- Reconocimiento e interpretación

La visión por computadora permite realizar ya sea segmentación de imágenes, reconocimiento de patrones, clasificación de objetos, seguimiento de objetos y detección de objetos, además de tareas más precisas como el reconocimiento facial que es una forma avanzada de detección de objetos.

La visión por computadora emplea redes neuronales para clasificar cantidades masivas de datos hasta que logra comprender lo que está mirando. Estos sistemas funcionan interpretando una imagen o video como una serie de píxeles, cada uno etiquetado con un valor de color. [21]

**Figura 1.24**

*Ilustración de proceso de procesamiento de imágenes*



*Nota. Imagen tomada de [22].*

## **Capítulo 2**

## 2. Metodología

En el presente capítulo se expone la metodología implementada para alcanzar los objetivos propuestos. Se abordan conceptos que abarcan desde los requisitos de diseño, las posibles soluciones propuestas, hasta los parámetros de diseño desde las perspectivas mecánica, electrónica y de control.

### 2.1 Requerimientos de Diseño

Para realizar el diseño de un robot autónomo basado en visión por computador, se siguieron ciertos requerimientos dados por el cliente, se aprecian en la **tabla 2.1**:

**Tabla 2.1**

*Requerimientos del diseño*

Tipo de requerimiento	Detalles del requerimiento
Mecánicos	<ul style="list-style-type: none"> <li>• Se tendrían motores para el movimiento de las llantas del robot, y otro motor para el movimiento del mecanismo de recolección.</li> <li>• Depósito con fácil acceso a las pelotas recolectadas.</li> <li>• Ligero y fácil de transportar manualmente. (peso máximo 15 Kg)</li> </ul>
Electrónico	<ul style="list-style-type: none"> <li>• Colocación del microcontrolador, dispositivos de control y actuadores.</li> <li>• Autonomía de al menos 1 hora.</li> </ul>
De Control	<ul style="list-style-type: none"> <li>• Uso de sensores como fuente de apoyo para detectar el acercamiento a las pelotas.</li> <li>• Un código de reconocimiento de color.</li> </ul>

## 2.2 Selección de alternativas

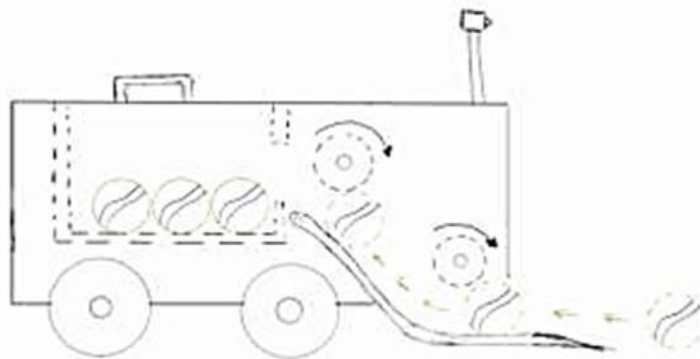
En vista del problema del primer capítulo, se desarrollaron tres alternativas de solución focalizadas en el mecanismo de recolección del robot móvil, para recoger las pelotas de tenis durante un entrenamiento eficiente.

- **Alternativa uno**

En esta propuesta, el diseño incorporará un mecanismo de recolección con dos rodillos, que pueden ser de diversos materiales, como pintura o caucho. Además, se incluirá una rampa en la parte interna del robot para facilitar el ascenso de las pelotas hacia el almacenamiento.

**Figura 2.1**

*Boceto de primera alternativa a la problemática*

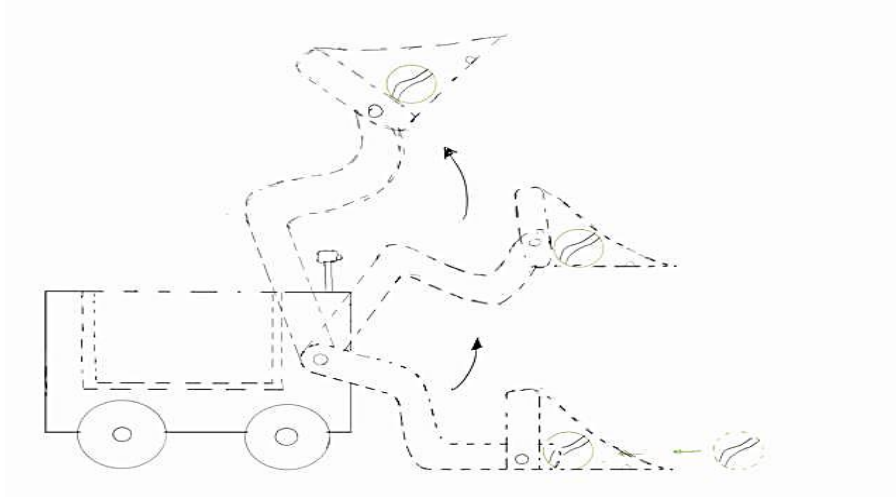


- **Alternativa dos**

Diseño del mecanismo de recolección sería mediante una garra como las máquinas de construcción, la cual recogería una cantidad de pelotas haciendo un movimiento elevándose el mecanismo y poniéndolas en un almacenamiento.

**Figura 2.2**

*Boceto de segunda alternativa a la problemática*

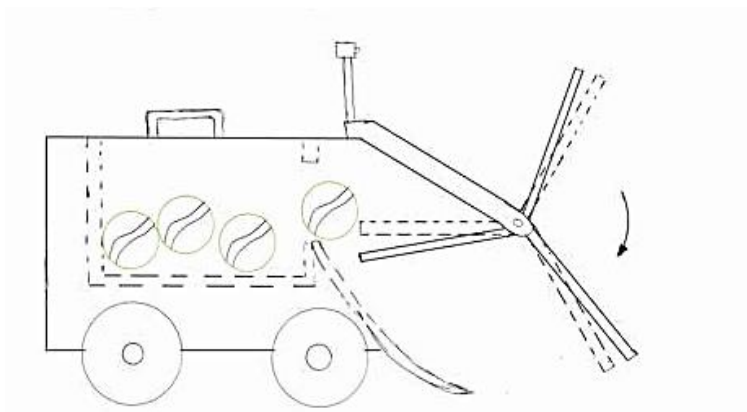


- **Alternativa tres**

Diseño del mecanismo de recolección es hecho mediante el uso de paletas las cuales tendrán un giro de 360° en su propio eje, todo este movimiento para poder poner las pelotas de tenis en su adecuado almacenamiento.

**Figura 2.3**

*Boceto de tercera alternativa a la problemática*



Con estas tres alternativas en mente, se evaluaron según diversos criterios para una perspectiva que determine la solución más viable frente al problema definido. A continuación, se describen los criterios considerados en dicha evaluación:

- **Precisión:** El mecanismo que sea elegido debe realizar el acto de recolección de una manera precisa tal que ninguna pelota que este a su alcance no se les escape y pueda ser puesta en el almacenamiento.
- **Rapidez del diseño:** qué tan ágil sería el mecanismo de recolección al hacer el movimiento de observar la pelota y recolectarlas en su interior.
- **Complejidad del sistema:** La construcción de la alternativa, qué tan difícil sería traerla a la realidad con la menor cantidad de fallos y cambios a realizar.
- **Visualmente agradable:** El agrado que tenga la gente para comprarlo y usarlo en una cancha de tenis.
- **Costo Inicial:** El valor de los elementos y dispositivos necesarios para la creación de las propuestas.

En la **tabla 2.2** a continuación se puede observar la ponderación que se le da a cada criterio previamente mencionados.

**Tabla 2.2***Crterios y sus relevancias*

<b>Criterio</b>	<b>Relevancia</b>	<b>Peso</b>
Precisión	1	7
Rapidez	2	5
Complejidad del sistema	3	4
Visualmente agradable	4	2.4
Costo final	5	1.6

Observando los valores totales para cada alternativa se determina que la alternativa número 3 sería la más viable en realizar con respecto a la problemática planteada, tomando en cuenta la precisión del mecanismo, la rapidez del robot con sus variables, la complejidad de realizar el sistema de la alternativa, si es visualmente agradable al público y su costo a realizar la implementación.



**Tabla 2.3***Matriz de decisión para seleccionar la mejor alternativa*

	Criterio 1	Criterio 2	Criterio 3	Criterio 4	Criterio 5	
<b>Descripción</b>	Precisión	Rapidez	Complejidad del sistema	Visualmente agradable	Costo inicial	
	Criterio 1	Criterio 2	Criterio 3	Criterio 4	Criterio 5	Total
<b>Peso</b>	7	5	4	2.4	1.6	16
<b>%</b>	35%	25%	20%	12%	8%	100%
	Calificación	Calificación	Calificación	Calificación	Calificación	
Solución 1	8	8	7	8	7	7.72
Solución 2	3	5	2	5	6	3.78
Solución 3	8	8	7	8	8	7.8

### 2.3 Metodología de diseño

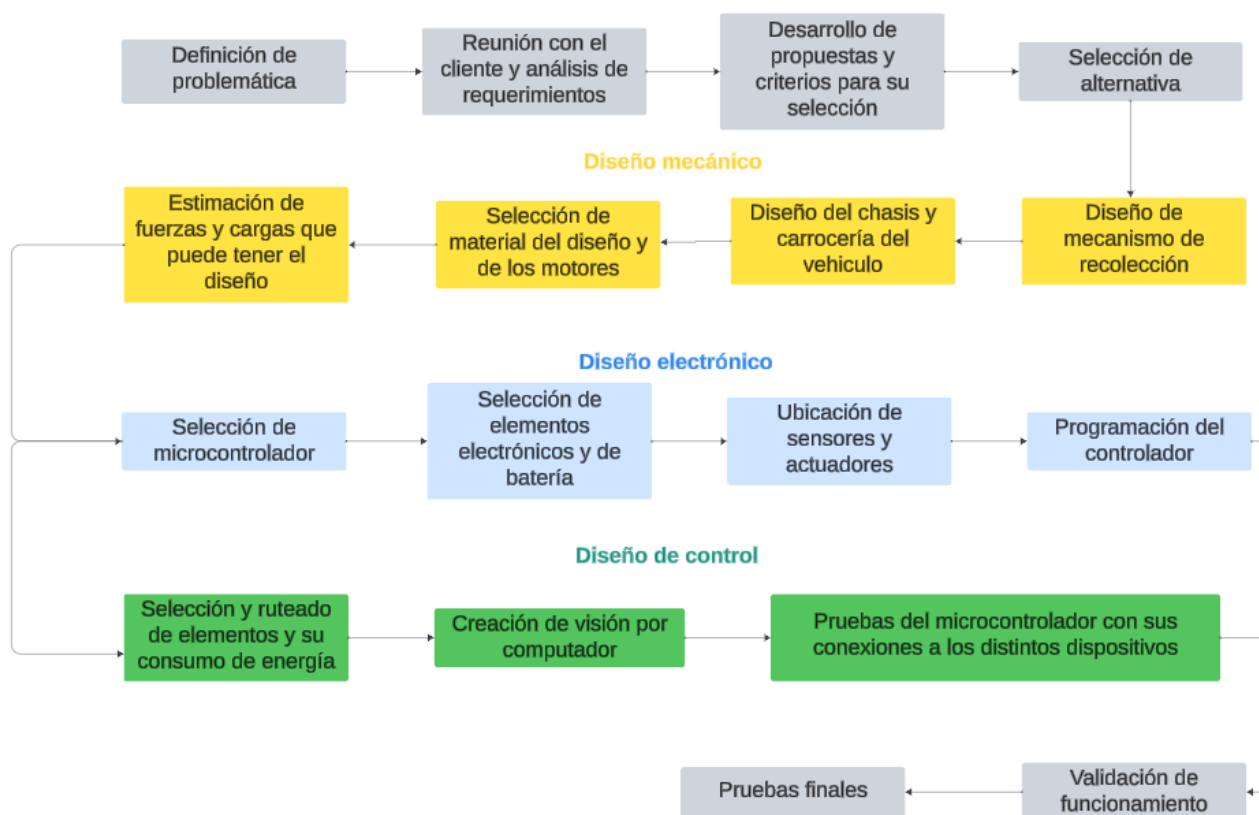
Para la elaboración del diseño final, se llevaron a cabo diversas etapas, cada una enfocada en aspectos específicos del proyecto. La **figura 2.4** presenta estas etapas junto con las acciones correspondientes. Inicialmente, en la fase inicial, se definió el problema mediante la interacción con el cliente y la recopilación de sus inquietudes, lo que condujo al desarrollo de una lista de alternativas para la solución. A continuación, se establecieron criterios específicos para la selección de la solución óptima.

El proceso de diseño se inició con el enfoque en la parte mecánica, abordando las dimensiones y seleccionando los materiales para conceptualizar las piezas aplicables al chasis del vehículo a escala. Posteriormente, considerando las dimensiones del mecanismo, se eligieron los elementos y se determinó su posición en función del microcontrolador seleccionado. Una vez seleccionados todos los dispositivos, se procedió con la programación.

Con el diseño mecánico y eléctrico establecido, se llevó a cabo la fase de diseño de control, sometiéndola a un proceso de validación para asegurar que el funcionamiento se alinearía con los objetivos previstos.

Figura 2.4

Diagrama de proceso de metodología

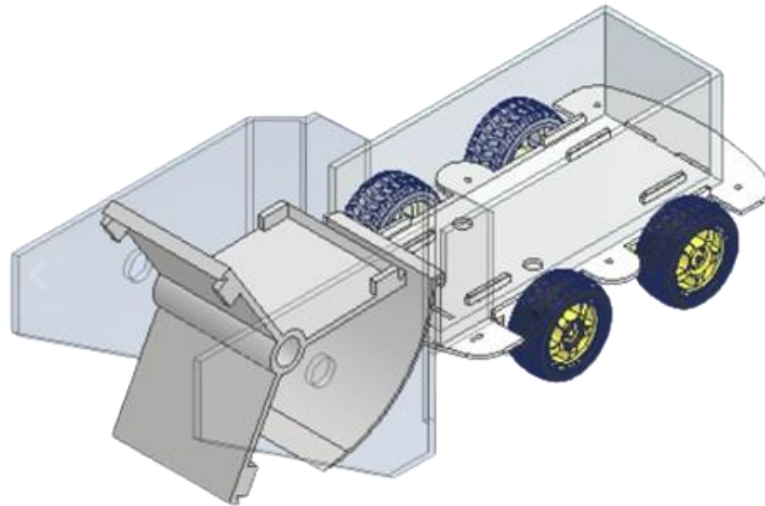


## 2.4 Diseño conceptual

En la **figura 2.5** se muestra la versión inicial de la solución propuesta después de la fase de creación. Se identifican elementos clave, como el mecanismo de recolección, motores para las ruedas y el espacio para componentes electrónicos, todos dimensionados adecuadamente en el chasis seleccionado. El diseño garantiza que el cliente pueda extraer las pelotas del almacenamiento y que el mecanismo continúe su operación sin interrupciones. La estructura ensamblada se ha optimizado para eficiencia y seguridad, evitando desarmes y manteniendo un sistema de cables organizado.

**Figura 2.5**

*Ilustración de diseño en software Inventor*



## **2.5 Parámetros de diseño mecánico**

Para representar un diseño final que satisfaga lo pedido por el cliente y se pueda realizar con los estándares del diseño de la ingeniería, se tomaron en cuenta simulaciones para saber datos para elegir elementos, considerando varias restricciones de elementos.

### **2.5.1 Diseño mecánico del robot**

#### **2.5.1.1 Componentes del diseño mecánico del robot**

Para la selección de ciertos componentes del diseño se realizaron diversos cálculos para determinar el funcionamiento de este equipo, desde los motores para el movimiento del robot hasta el tamaño del eje, se puede encontrar en el **Apéndice C** esto se hizo con ayuda de la **tabla 2.4** que contiene todos los pesos de los componentes del diseño.

**Tabla 2.4***Peso de componentes del diseño*

<b>Material</b>	<b>Densidad</b>	<b>Unidad</b>	<b>Espesor</b>	<b>Unidad</b>	
<b>PLA</b>	1.25	gr/cm <sup>3</sup>	0.25	cm	
<b>Acrílico</b>	1.18	gr/cm <sup>3</sup>	1.00	cm	

<b>Parte</b>	<b>Área</b>	<b>Unidad</b>	<b>Peso</b>	<b>Unidad</b>	<b>Cantidad</b>
<b>Depósito</b>	2905.01	cm <sup>2</sup>	907.82	gr	1
<b>Tapa posterior</b>	358.41	cm <sup>2</sup>	112.00	gr	2
<b>Rampa</b>	168.89	cm <sup>2</sup>	52.78	gr	1
<b>Paletas Recolección</b>	371.31	cm <sup>2</sup>	464.14	gr	1
<b>Paredes laterales</b>	569.13	cm <sup>2</sup>	1343.15	gr	2
<b>Soporte superior</b>	230.00	cm <sup>2</sup>	271.40	gr	1
<b>Base Webcam</b>	372.34	cm <sup>2</sup>	116.36	gr	1
<b>Kit Arduino</b>	-	-	500.00	gr	1
<b>Rodamientos</b>	-	-	18.14	gr	2
<b>Caster Wheel</b>	-	-	192.00	gr	1
<b>Pernos M6x20</b>	-	-	56.00	gr	8
<b>Pernos M8x30</b>	-	-	40.00	gr	2
<b>Pernos M6x90</b>	-	-	100.00	gr	2
<b>Pernos M3x10</b>	-	-	103.50	gr	15
<b>Pernos M8x20</b>	-	-	69.00	gr	6
<b>Pernos M6x10</b>	-	-	16.32	gr	4
<b>Pernos M4x40</b>	-	-	3.40	gr	1
<b>Tuercas M3</b>	-	-	29.55	gr	15
<b>Tuercas M6</b>	-	-	25.00	gr	10
<b>Tuercas M8</b>	-	-	37.60	gr	8
<b>Pelotas de tenis</b>	-	-	175.50	gr	3
<b>Cámara</b>	-	-	1.00	gr	1
<b>Motores</b>	-	-	100.00	gr	4
<b>Circuitería</b>	-	-	188.00	gr	1
<b>Raspberry</b>	-	-	500.00	gr	1
<b>Batería</b>	-	-	51.00	gr	1
<b>Peso total</b>	-	-	5473.66	gr	
			5.47	kg	

Continuando con el diseño mecánico, se consideraron los requerimientos de diseño sugeridos por el cliente, como que el robot debía ser ligero para desplazarlo hacia la cancha de tenis, para esto se empleó PLA y acrílico para el diseño de las piezas, así se garantizaba un diseño robusto, pero ligero y fácil de maniobrar. Otro de los requerimientos solicitados fue que el depósito de las pelotas de tenis se pueda vaciar de una forma fácil y rápida.

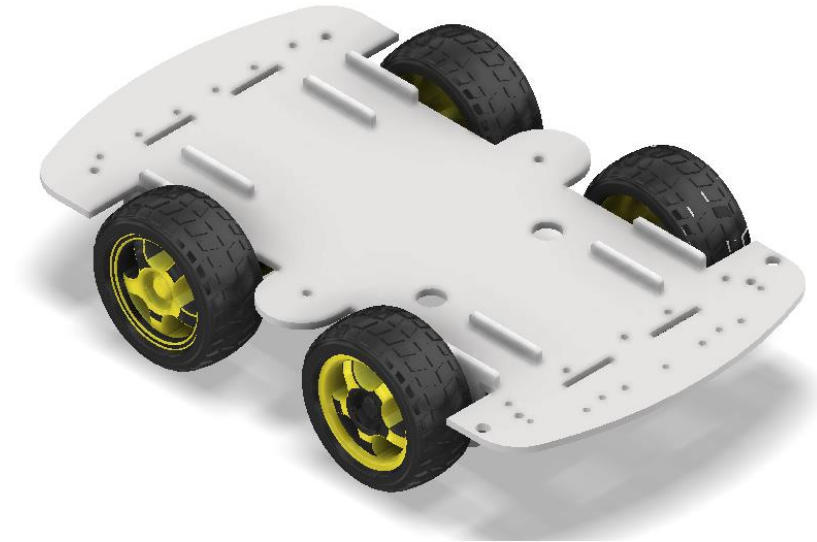
El diseño propuesto consta de las partes descritas a continuación:

- Chasis
- Depósito de pelotas
- Rampa
- Paredes de soporte lateral
- Paletas de recolección
- Base superior
- Soporte de cámara

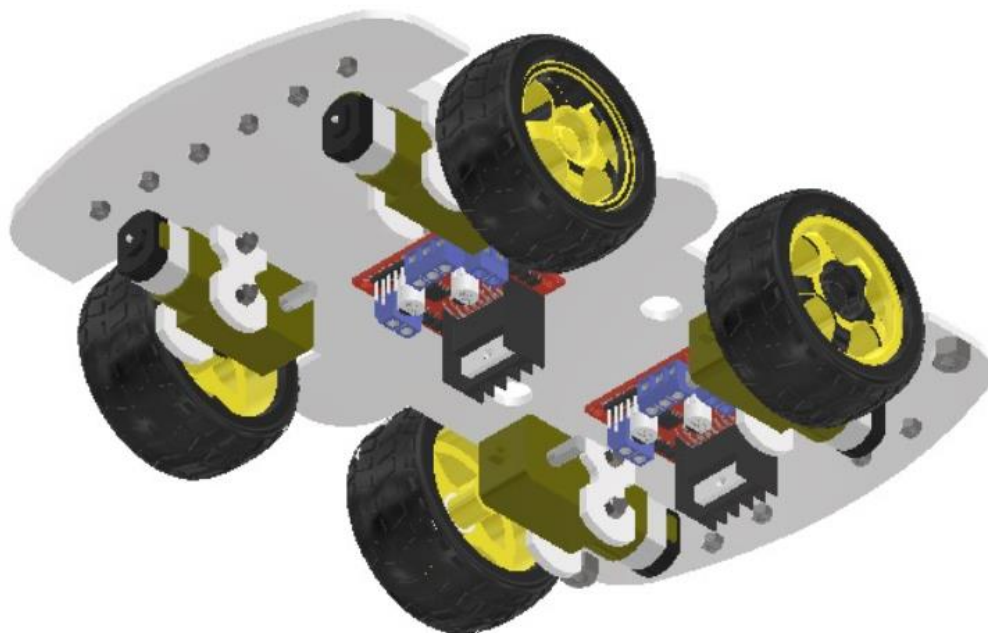
El chasis, con un espesor de 3 mm y fabricado en acrílico, presenta orificios estratégicos para acoplar los soportes y pernos de los motores. Tanto el chasis como los motores, soportes y ruedas fueron seleccionados de un kit de Arduino disponible en el mercado. Importante destacar que el kit incluye cuatro motores con ruedas, donde las ruedas derechas y las izquierdas operan de manera sincronizada para lograr la cinemática de un robot diferencial. Este control se realiza mediante puentes H que gestionan los motores de las ruedas.

**Figura 2.6**

*Vista superior en perspectiva del ensamblaje del chasis del robot*

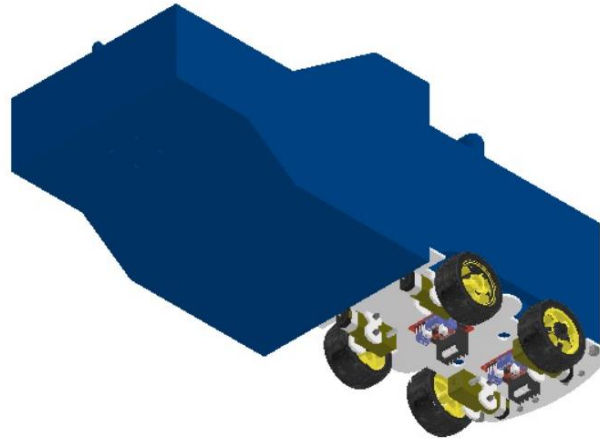
**Figura 2.7**

*Vista inferior en perspectiva del ensamblaje del chasis del robot*



**Figura 2.8**

*Vista inferior en perspectiva del chasis del robot ya ensamblado*

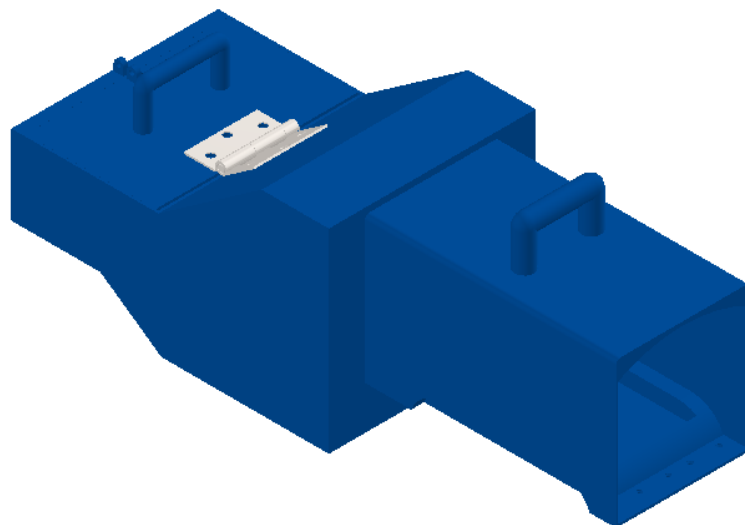


Para el depósito de pelotas se empleó PLA de 2.5 mm de espesor, el dimensionamiento se hizo considerando el espacio disponible encima del chasis y en la parte posterior del robot para almacenar por lo menos 15 pelotas de tenis en simultáneo, además se tuvieron que modificar para no interferir con el funcionamiento de las llantas del robot o los soportes de los motores del chasis. Finalmente, se emplearon juntas empernadas para la respectiva sujeción entre el depósito y el chasis del robot.



**Figura 2.9**

*Vista superior en perspectiva del depósito de pelotas*

**Figura 2.10**

*Vista inferior en perspectiva del depósito de pelotas*



**Figura 2.11**

*Vista superior depósito ensamblado*

**Figura 2.12**

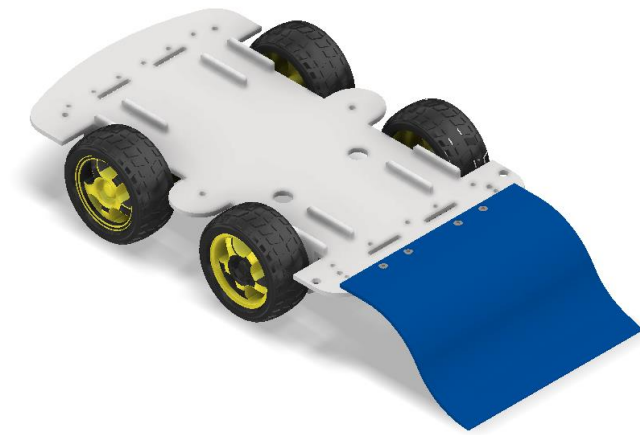
*Vista inferior depósito ensamblado*



El diseño de la rampa también se lo hizo en PLA de 2.5 mm de espesor con una altura dada por el chasis del robot y una curvatura pertinente para que las pelotas de tenis subas por ella sin ningún inconveniente y puedan ser almacenadas en el depósito, se emplearon pernos para juntar la rampa con el chasis del robot.

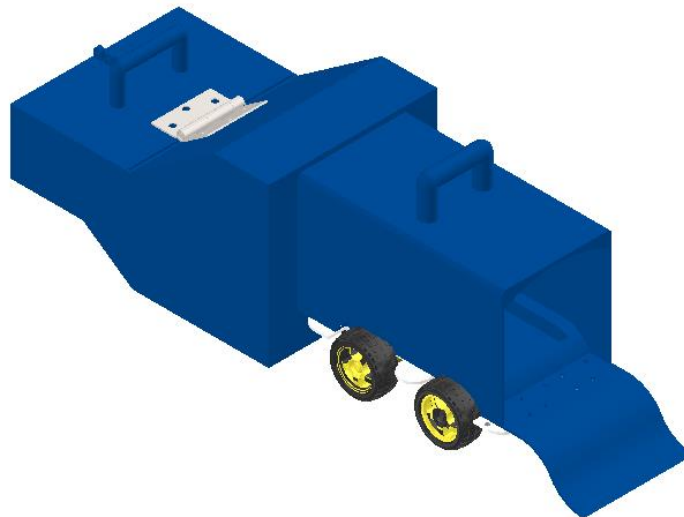
**Figura 2.13**

*Vista en perspectiva de la rampa*



**Figura 2.14**

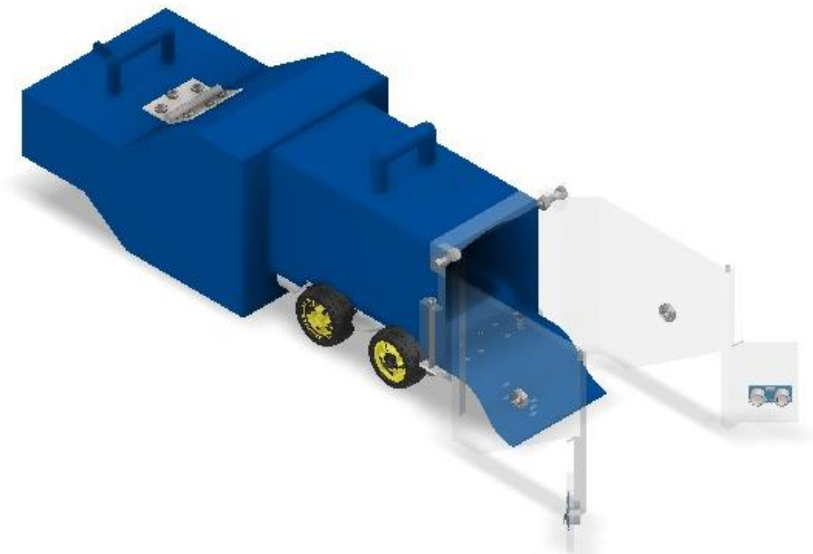
*Vista en perspectiva de la rampa ensamblada*



Las paredes de soporte lateral fueron diseñadas en acrílico de 10 mm de espesor, dichas paredes cuentan con un agujero de 20 mm de diámetro a una profundidad de 8 mm para acoplar los rodamientos que permitirán la rotación de las paletas de recolección, en los otros 2 mm habrá un agujero de 9 mm para que pase el eje de las paletas de recolección y a su vez evitar el deslizamiento del rodamiento, estas paredes también estarán aseguradas al chasis mediante pernos de sujeción, además existe una distancia de 210 mm entre ambas paredes para posibilitar el paso de 2 pelotas en simultáneo y permitir la recolección aun cuando el robot no se aproxime de forma precisa a la posición de la pelota. Además, se implementó una barra de soporte en la parte superior, dándole mayor estabilidad al diseño al momento de realizar el proceso de recolección, y evitando la separación de las paredes laterales. Finalmente, la distancia entre el suelo y la parte inferior de las paredes es de 10 mm para evitar inconvenientes de fricción entre la superficie y las mismas.

**Figura 2.15**

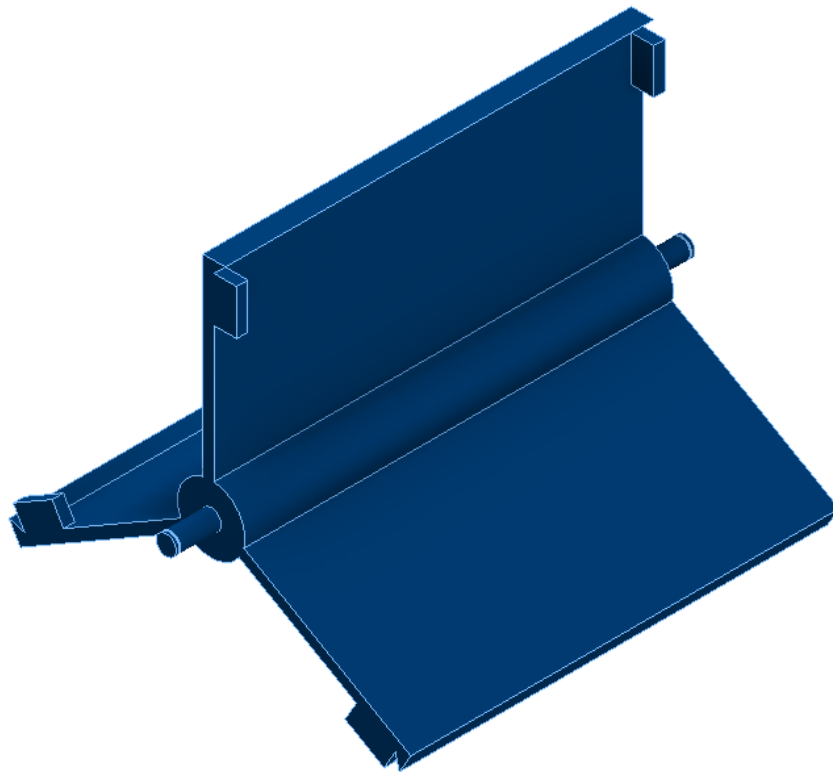
*Vista en perspectiva de las paredes laterales ensambladas*



El sistema de paletas de recolección está diseñado en PLA de 5 mm de espesor, es un sistema macizo que cuenta tanto con las paletas como con el eje que permitirá la rotación de las mismas, las paletas están ubicadas a  $120^\circ$  entre sí, y tienen una longitud de 90 mm para garantizar que puedan empujar las pelotas de tenis, además el eje tiene una longitud de 230 mm y a 2 mm de cada uno de sus extremos cuenta con un canal para vinchas para evitar el deslizamiento de los rodamientos.

**Figura 2.16**

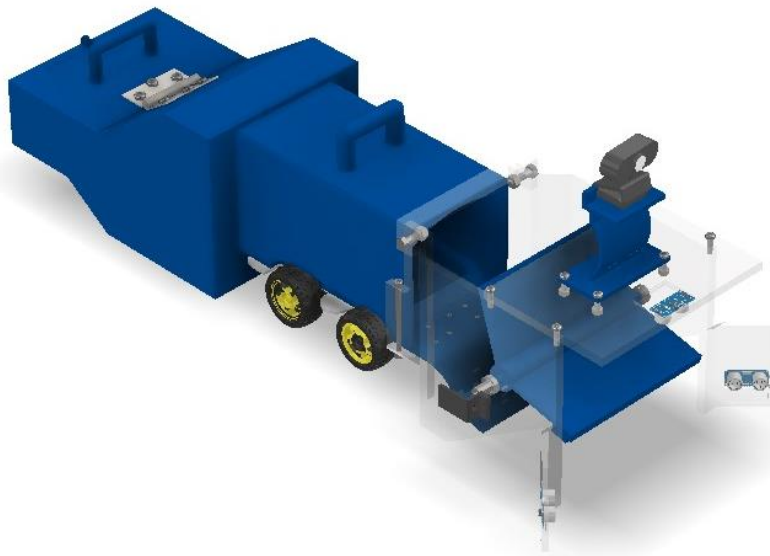
*Mecanismo de paletas para recolección de pelotas*



La base superior es de acrílico de 10 mm de espesor y está el soporte de la cámara web, asegurada al resto del robot mediante pernos de sujeción.

**Figura 2.17**

*Vista isométrica del diseño completo*

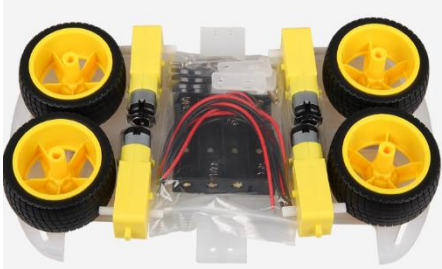


### ***2.5.2 Selección de motores***

Para la selección de motores se debe tener en cuenta el torque deseado y el dimensionamiento para ver si es viable para el diseño.

En primer lugar, se seleccionaron los motores integrados para el chasis 4WD del robot, los cuales posibilitan su movimiento. A continuación, se describen las especificaciones correspondientes:

**Tabla 2.5***Características de motores de chasis*

Figura	Características
	<p data-bbox="824 449 1101 485">Voltaje: 4.5-12 V DC</p> <hr/> <p data-bbox="824 520 1268 556">Torque: 800 – 1200(GF/CM MIN)</p> <hr/> <p data-bbox="824 592 1317 627">Dimensiones del chasis: 280 x 156 x 3</p> <hr/> <p data-bbox="824 663 1263 699">Velocidad de motor: 90 - 300 rpm</p>

*Nota. La tabla de información sobre el chasis es sacada de la página de origen Joy-it [24]*

Para el mecanismo de recolección, se usa un motor similar al mencionado, siendo estos motores de engranajes DC para Arduino, que proporciona la fuerza de torque necesaria para accionar las paletas adecuadamente.

## **2.6 Parámetros de diseño electrónico**

El diseño electrónico es una parte importante, ya que aquí se selecciona los sensores y cualquier otro tipo de aparato que permita navegar de forma autónoma y el poder realizar la acción de localización y planificación.

### **2.6.1 Selección de sensores para control del vehículo**

Para realizar la acción de localización y detectar donde mismo estará el objeto se dispondrá de sensores de detención y una cámara para la visión.

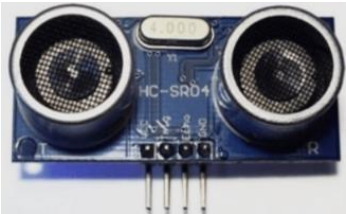
Para determinar la presencia de un objeto en alguna dirección cercana al mecanismo, se empleó un sensor con las siguientes características:

- **Precisión:** Dado que estará en una cancha de tenis debe tener mínimo distancia de 20 cm hasta 5 metros.
- **Eficiencia de detección:** Debe estar hecho para detectar todo tipo de materiales, independientemente de su transparencia, forma o color.
- **Montaje:** Tamaño no tan grande para poder posicionarse fácilmente y peso leve para no perjudicar el movimiento del robot.
- **Compatibilidad:** Con el microcontrolador raspberry.

El sensor ultrasónico que se muestra a continuación tiene las siguientes características.

**Tabla 2.6**

*Características de sensor ultrasónico*

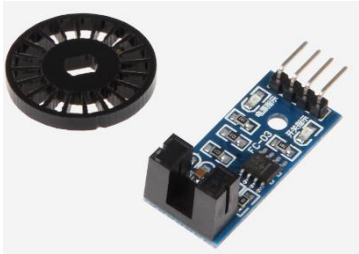
Figura	Características
	<p>Voltaie: 5 VDC</p> <hr/> <p>Ángulo de detección de no más de 15°</p> <hr/> <p>Distancia de detección de 2 cm hasta 450 cm</p> <hr/> <p>Intensidad de menos de 2 mA</p>

*Nota. La tabla de información sobre el sensor sacado de la página Ultra-lab [25]*

Para permitir que el robot, con la asistencia de este sensor y una cámara, ajuste el movimiento de sus ruedas y determine la distancia necesaria para detenerse, es esencial utilizar un sensor de velocidad compatible con la Raspberry Pi. Esto garantizará un proceso de análisis de datos más eficiente.




**Tabla 2.7***Características de sensor LM393*

Figura	Características
	Voltaje: 3.3 - 5 V
	Dimensiones: 32 x 14 mm
	Compuesto por dos componentes el sensor y un disco
	Peso: 4 g

*Nota. La tabla de información sobre el chasis es sacada de la página de origen Joy-it [26]*

La cámara que se usará para localizar tendrá la importancia de detectar el color, es su mayor virtud para poder detectar el objeto y saber su posición.

**Tabla 2.8***Características de cámara*

Figura	Características
	Rango de detección: 5 m
	Rango de fotograma en RGB: 30 fps
	Resolución máxima de video: 1080p

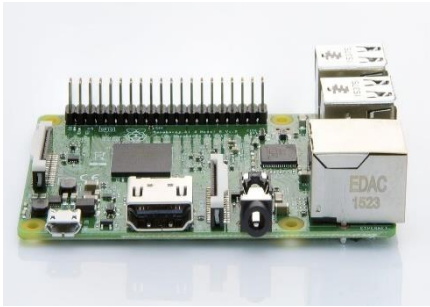
*Nota. La tabla de información de la cámara web es de Amazon. [27]*

Ya teniendo todo estos actuadores, motores y sensores se necesita un componente que pueda controlarlos y que tenga un nivel no alto de complejidad de conexión entre ellos.

La Raspberry Pi 3B, detallada en la **tabla 2.8** junto con sus características, es un dispositivo sin las especificaciones más modernas de un microcontrolador, obligando a buscar programas compatibles que puedan ejecutarse en él y facilitar su uso.

**Tabla 2.9**

*Características de Raspberry Pi 3B*

Figura	Características
	Quad Core 1.2GHz Broadcom BCM2837
	32bit CPU
	1GB RAM
	GPIO extendido de 40 pines
	BCM43438 LAN inalámbrica y Bluetooth de baja energía (BLE)

*Nota. La tabla de información de la cámara de profundidad es de Amazon. [28]*

## 2.7 Selección de fuente de energía

La fuente de energía deberá alimentar a todo el sistema, dependiendo principalmente del microcontrolador que controlada todo, haciendo que las características que se buscan en esta fuente de poder serán:


- Voltaje requerido de alimentación: Como mínimo deberá proporcional 12 voltios.
- Tamaño y peso: Deberá ser ligero y no de gran dimensionalidad.

- Reusabilidad: La batería debería ser recargable para que el producto pueda ser utilizado cuantas veces el usuario desee.

La batería que cumple con los requisitos para alimentar el mecanismo y garantizar un funcionamiento adecuado sería una batería LiPo. Esta batería presenta las siguientes características

**Tabla 2.10**

*Características de batería LiPo*

Figura	Características
	Voltaje: 11.1V 3S 2
	Capacidad: 2200 mAh
	Descarga: 25C
	Dimensiones: 26*35*88mm

*Nota. La tabla de información de la batería LiPo sale de la página de compra roboticsec [29]*

## 2.8 Parámetros de diseño de control

### 2.8.1 Control del vehículo

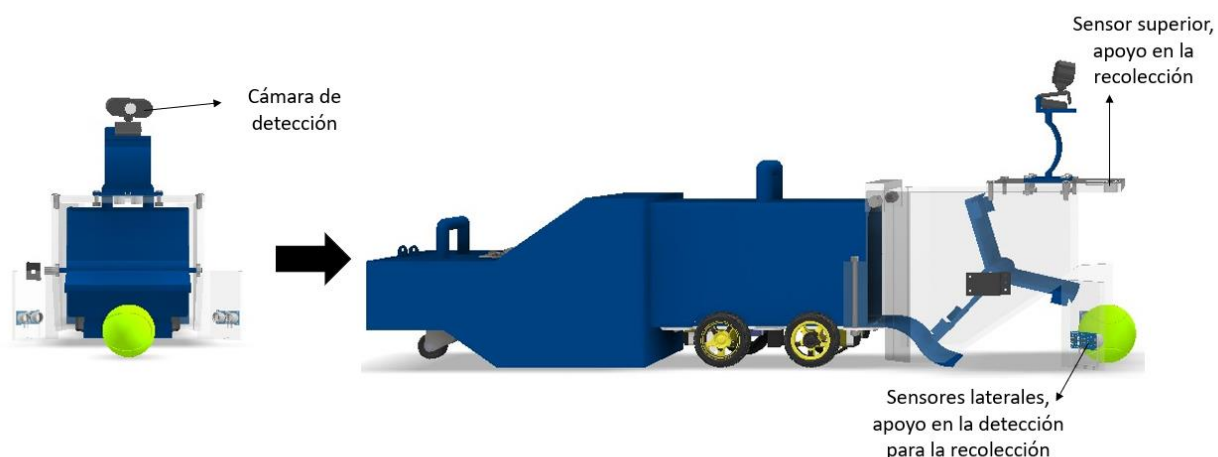
Para la inicialización del vehículo se deberá encender la Raspberry que es el cerebro del sistema, esta tendrá unas conexiones mediante puentes H a las 4 ruedas del vehículo permitiendo que se activen cuando el código elaborado lo requiera.

¿De qué manera se coordina el accionar de las ruedas para activarse, avanzar o retroceder, y cuál es la distancia precisa que deben recorrer? Todas estas operaciones se realizan mediante la interacción entre la cámara y los sensores.

La cámara integrada permite la detección del color de las pelotas de tenis, lo cual provocará una reducción del 50% en la velocidad de los motores del chasis, permitiendo así una aproximación lenta hacia el objetivo. A pesar de contar con una cámara, se optó por incorporar sensores ultrasónicos para detectar la proximidad de la pelota de tenis al mecanismo recolector. Estos sensores fueron ubicados en tres posiciones estratégicas: dos en las paredes laterales con una inclinación específica que requiere la detección simultánea para confirmar que la pelota está dentro del rango de recolección del vehículo autónomo. El tercer sensor se colocó en el soporte superior, este se encarga de verificar que la distancia de 21.30 cm respecto al suelo disminuya a 15 cm cuando detecte que la pelota de tenis está en una posición adecuada para activar el mecanismo de recolección.

**Figura 2.18**

*Dispositivos de detección del diseño*

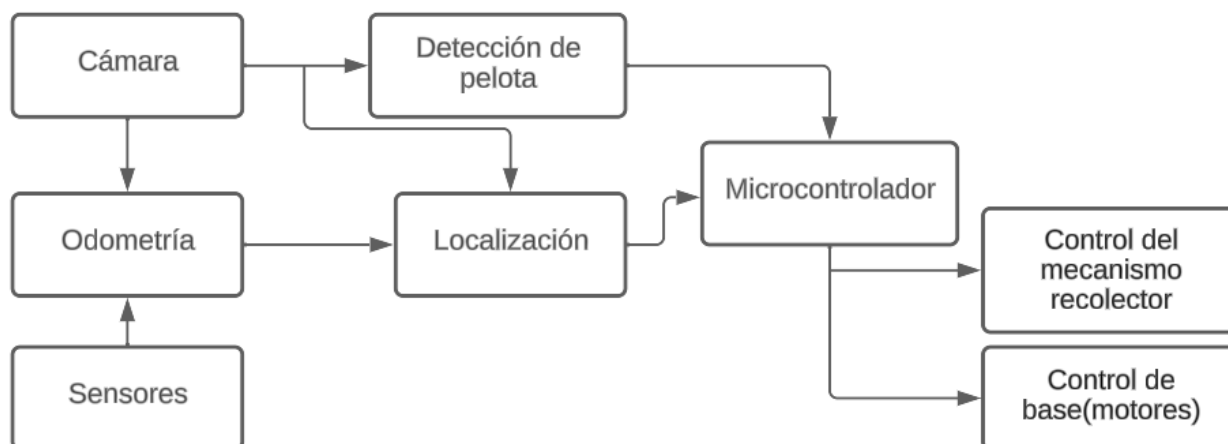


Todo este control del vehículo se puede leer en la programación en el **Apéndice B**.

El funcionamiento de control se puede apreciar en la siguiente ilustración:

**Figura 2.19**

*Funcionamiento del sistema de control*



Para el proceso de odometría se parte del diseño del robot tomando en consideración la cinemática de este. La configuración empleada fue la configuración diferencial la cual se puede apreciar en el **Apéndice D**.

### **2.8.2 Control del mecanismo de recolección**

Para este procedimiento se tomó en cuenta algunos factores como cuándo comienza el movimiento que debe efectuar para recoger la pelota y cuánto tiempo tardaría en apagarse.

Teniendo en cuenta lo previamente mencionado del control de motores de las ruedas, este mismo código tuviera que detener movimiento de estas cuando se active el mecanismo de recolección, debido a que la pelota ya se encontraría dentro del rango de las paletas de recolección, para luego ser puesta en el almacenamiento del vehículo.

Este mecanismo constaría de tres paletas dispuestas a un ángulo de 120 grados entre sí, permitiendo así el espacio necesario para alojar una o más pelotas. En el centro de estas paletas, se ubicaría un eje que transmitiría la potencia necesaria para el movimiento de las paletas al estar conectado a un motor.

Este motor se escogió pensando en el proceso de activarse y desactivarse continuamente siendo la mejor elección un motor paso a paso.

## Capítulo 3

### 3. Resultados y análisis

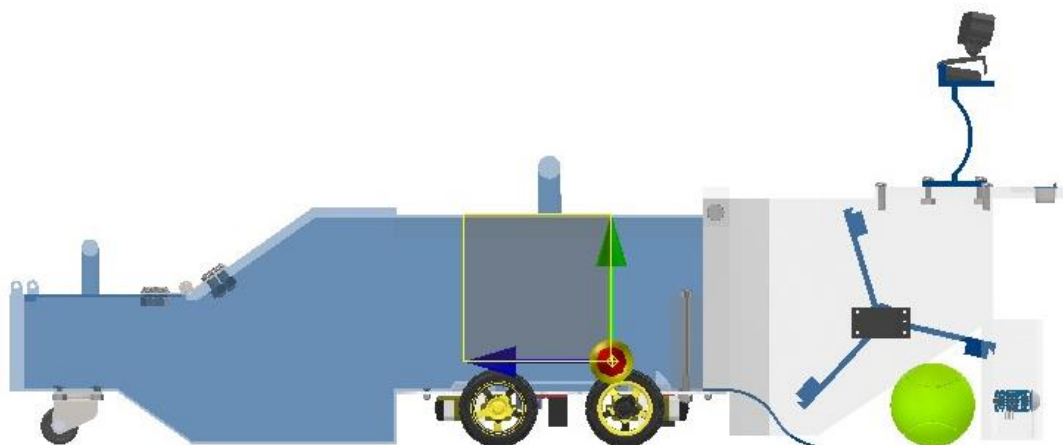
En esta sección se muestra los resultados del diseño de un robot autónomo recolector de pelotas de tenis asistido por visión por computador. Los resultados de cada punto, incluyendo funcionamiento y simulaciones, se exponen de forma individual. Se detallan las funciones del robot móvil y cómo trabajan en conjunto. Finalmente, se presenta el análisis de costos.

#### 3.1 Diseño final del dispositivo

Por los problemas de estabilidad que presentaba, hubo cambios respecto al modelo inicial. Para esto, se optó por modificar el depósito de pelotas de tenis haciéndolo más robusto y cambiando su geometría para agregarle una *caster wheel* en la parte posterior. De esta manera, se logró incrementar la estabilidad del robot al posicionar estratégicamente el centro de gravedad detrás de las ruedas delanteras. Este ajuste ha posibilitado el funcionamiento óptimo del dispositivo

**Figura 3.1**

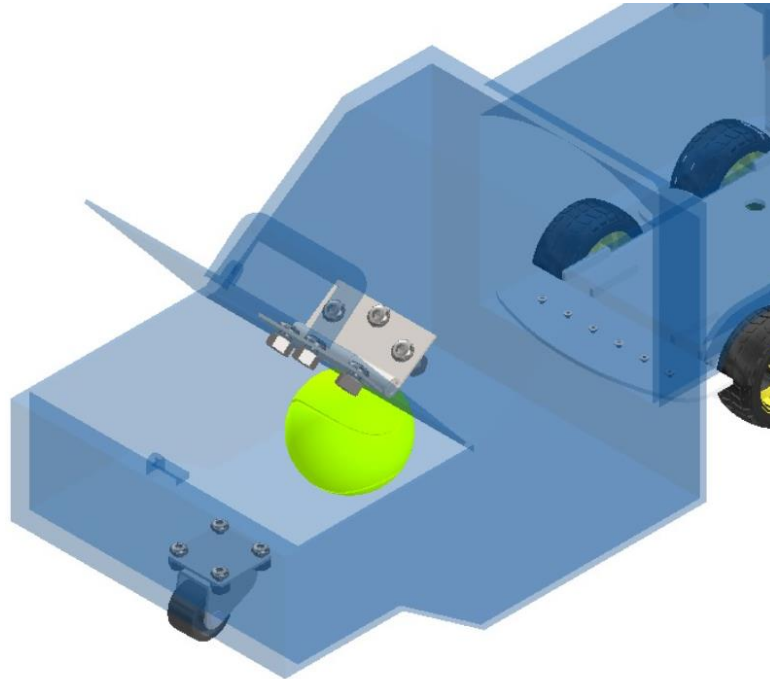
*Vista lateral del robot indicando el centro de gravedad del diseño*





**Figura 3.2**

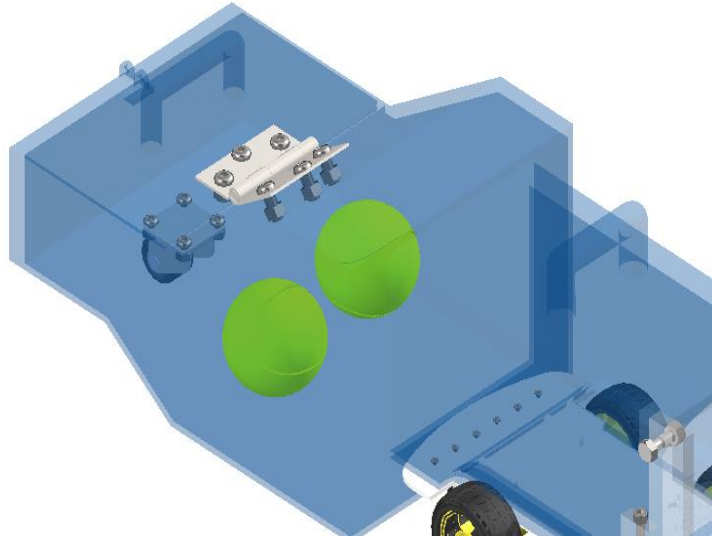
*Acople de caster wheel en parte posterior del depósito con pernos de sujeción*



Además, se le agregó una tapa en la parte posterior la cual permitirá la salida de las pelotas previamente recolectadas. Para incorporar dicha tapa se añadió una bisagra en la parte posterior del depósito sujeta a través de pernos M8 con sus respectivas tuercas. La tapa también cuenta con una aldaba para poder cerrar el depósito mientras el robot se encuentra en movimiento si así se requiere. Se agregaron también dos agarraderas, una en la tapa posterior y una en la parte superior del depósito, esto con el fin de facilitar tanto el transporte del robot, como el vaciado del depósito una vez se encuentre con su capacidad máxima.

**Figura 3.3**

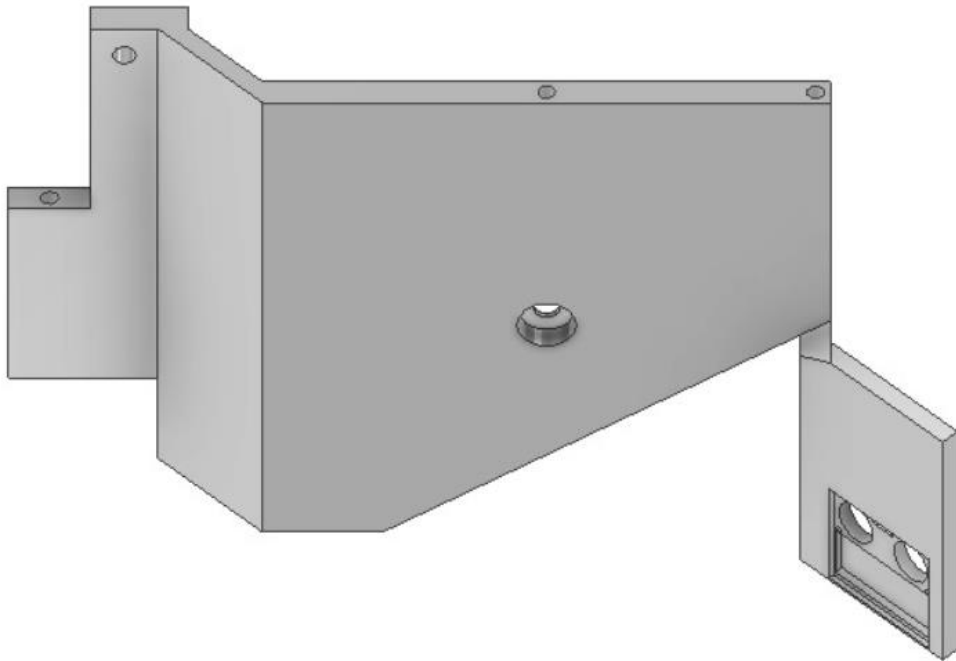
*Vista en perspectiva de la bisagra y las agarraderas.*



Finalmente se modificaron las paredes laterales, se le agregaron soportes en la parte frontal donde se ubicaron sensores ultrasónicos que permitirán tener una aproximación más precisa hacia la ubicación de la pelota una vez que el robot se encuentre a una distancia de aproximadamente 30 cm.

**Figura 3.4**

*Vista en perspectiva de pared lateral modificada*

**Figura 3.5**

*Vista en perspectiva del diseño final realizado.*

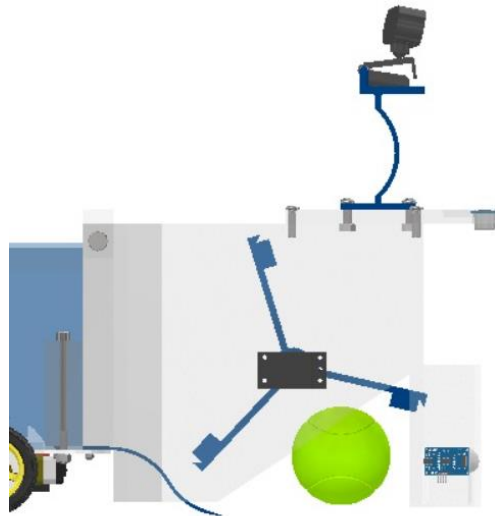


### 3.2 Funcionamiento del mecanismo de recolección

Se realizó la simulación del funcionamiento del mecanismo de paletas de recolección con la ayuda del software Inventor 2023 logrando así corroborar que cumple su función de forma óptima.

**Figura 3.6**

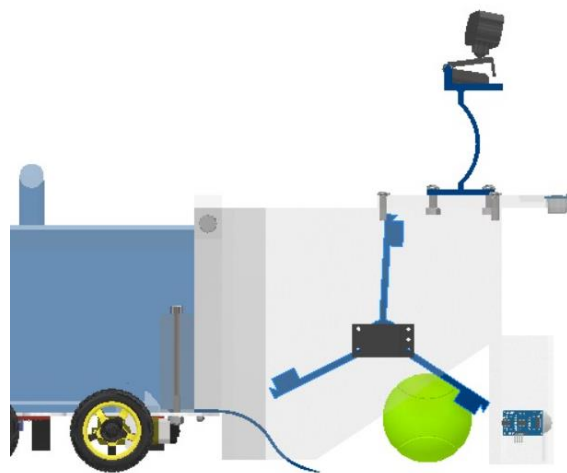
*Vista lateral del sistema de recolección*



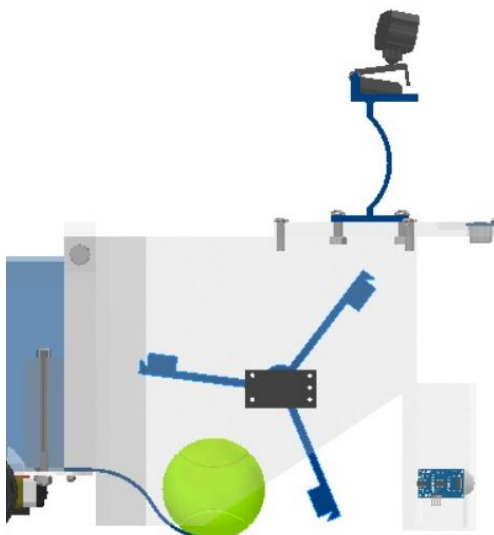
Se puede apreciar como las paletas de recolección golpean la pelota haciendo que la misma suba por la rampa y posteriormente se almacenen en el depósito posterior.

**Figura 3.7**

*Vista lateral, sistema de recolección golpeando la pelota de tenis*

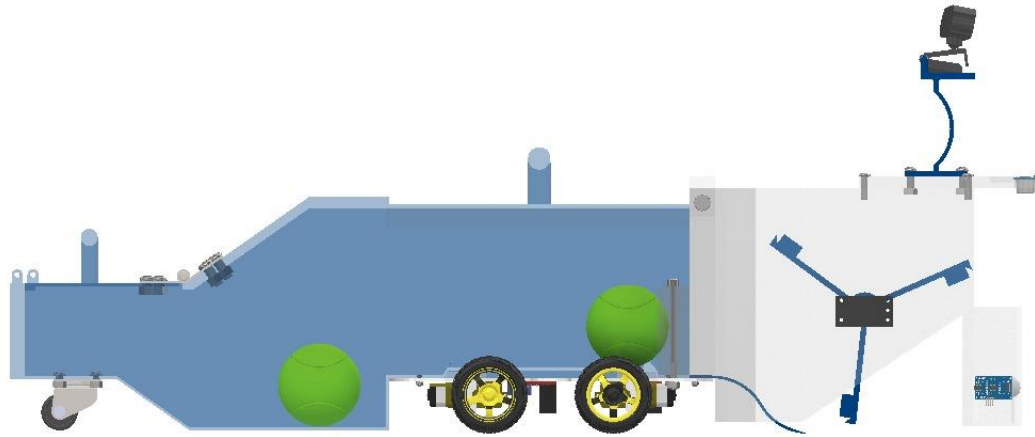
**Figura 3.8**

*Vista lateral pelota de tenis subiendo la rampa*



**Figura 3.9**

*Vista lateral pelota ingresando al depósito*



Enlace al video del funcionamiento del sistema de recolección:

<https://www.youtube.com/watch?v=c0pJnKKZdX4&feature=youtu.be>

### 3.3 Simulación de movimiento

Para evaluar la idoneidad del diseño mecánico para la movilización en un entorno del mundo real, se llevó a cabo una simulación mediante el uso de ROS (Robot Operating System) en su herramienta Gazebo.

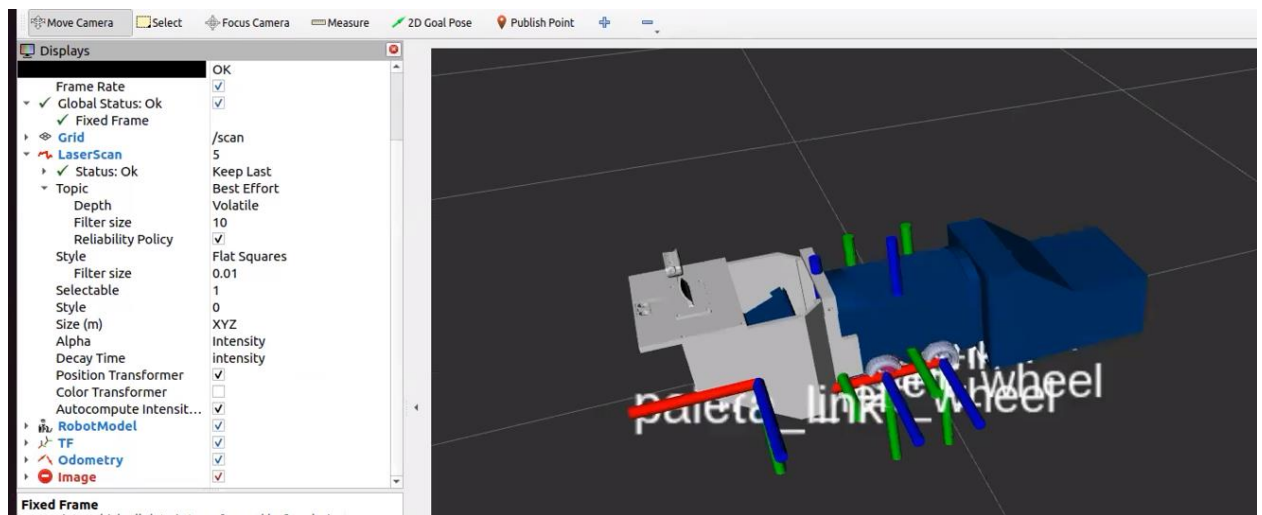
En la simulación de inventor se apreció el movimiento del mecanismo de recolección, mientras en esta, se ve el comportamiento del movimiento del diseño, se observa si el centro de masa está bien ubicado y permite el movimiento suave. También observar si el funcionamiento de un robot diferencial con cada lado del chasis se mueve simultáneamente es válido, realice correctamente el avance, retroceso y giros correspondientes.

Inicialmente, se procedió a realizar la conversión de las partes del diseño a formato URDF utilizando diversas herramientas, como SolidWorks 2023.

Una vez obtenida la carpeta con el diseño convertido, se configuró un espacio de trabajo en Gazebo para establecer el comportamiento de las cuatro ruedas. Estas ruedas operarían de manera diferencial, donde las dos ruedas izquierdas se moverían simultáneamente, al igual que las dos ruedas derechas. Este control se implementó en el archivo control.xacro, que actuó como el controlador en nuestra simulación.

**Figura 3.10**

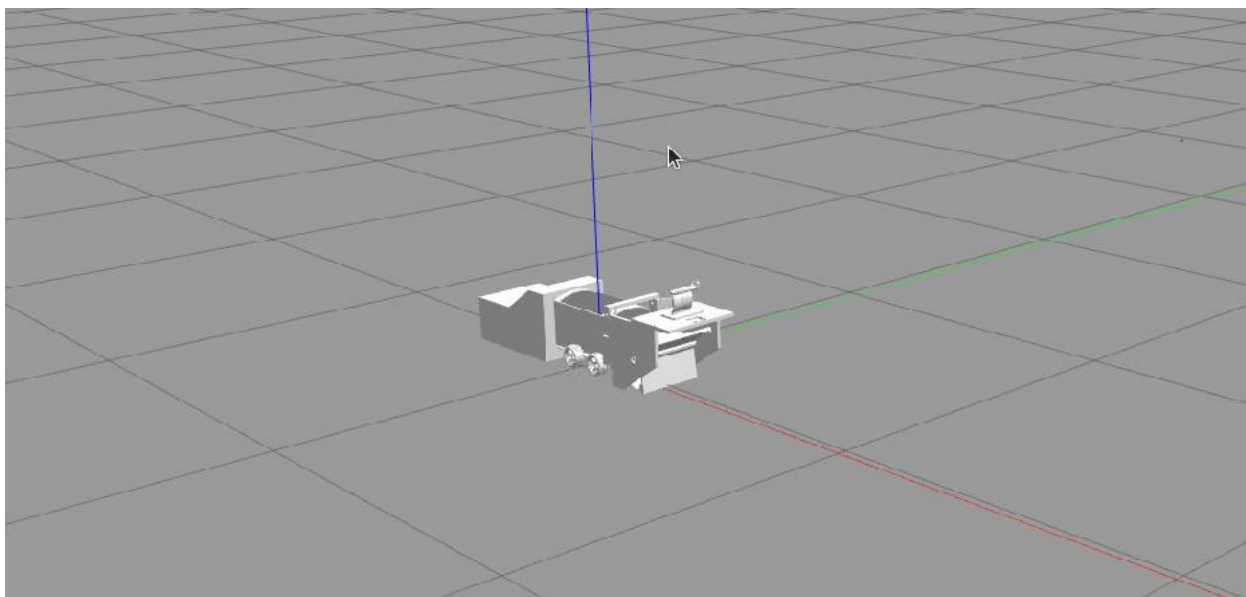
Vista de herramienta rviz



En la **figura 3.10**, presentada con antelación, se exhibe la herramienta Rviz, en la cual se pueden visualizar los enlaces de conexión entre los componentes, así como los marcos hacia los cuales están orientados. En consecuencia, se llevó a cabo la instalación de un sistema de dirección (*steering*) para facilitar el movimiento. Con todos estos elementos previamente mencionados, se accede al entorno de Gazebo, donde se procede a ejecutar movimientos en todas las direcciones, incluyendo los giros que el robot puede realizar.

**Figura 3.11**

*Vista de robot móvil simulado en Gazebo*



### **3.4 Simulación de detección de pelota de tenis desde robot**

Esta etapa de la simulación se encuentra relacionada con la visión por computadora. Al inicio, se creó un código para detectar el color verde, dado que nuestra Raspberry no puede procesar ciertas librerías, siendo esta la alternativa más viable.

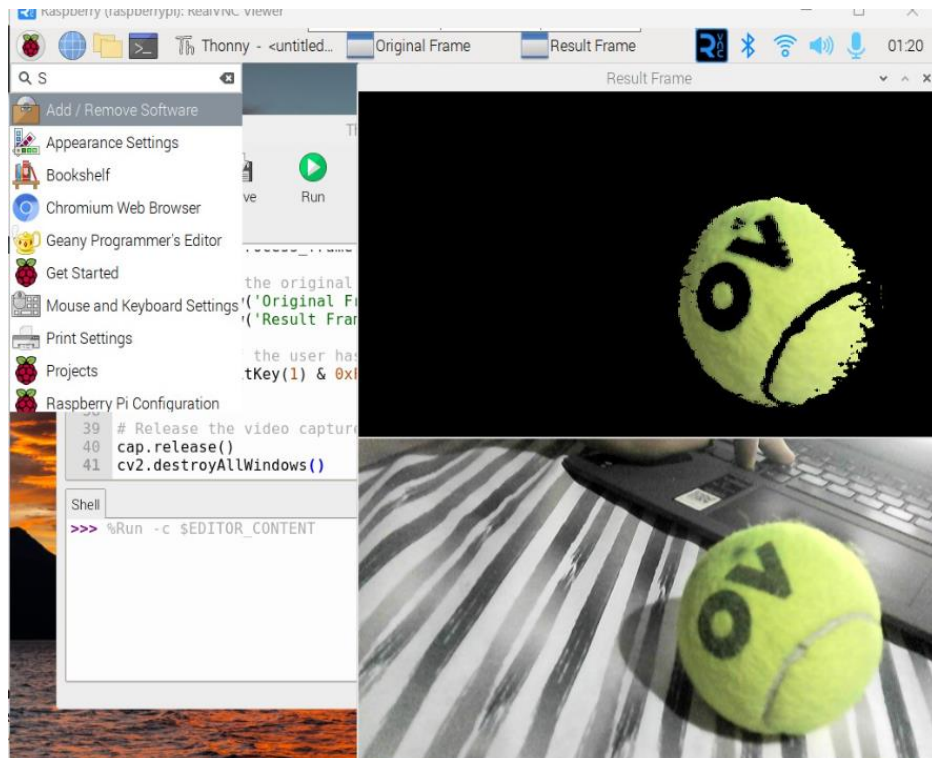
Una vez desarrollado el código, se procedió a realizar simulaciones para verificar su funcionamiento adecuado, confirmando su capacidad para detectar únicamente el color de la pelota de tenis. A partir de esta validación, se procederá a la visualización desde la Raspberry. Para ello, se deben instalar diversos softwares, siendo uno de ellos RealVNC Viewer, una herramienta que facilita la conexión de la Raspberry a una pantalla, permitiendo el control de los procesos internos, como el manejo del código, entre otras funciones.



Adicionalmente, se llevó a cabo la instalación de OpenCV con el propósito de posibilitar la observación de la señal capturada por la cámara al conectarse al dispositivo.

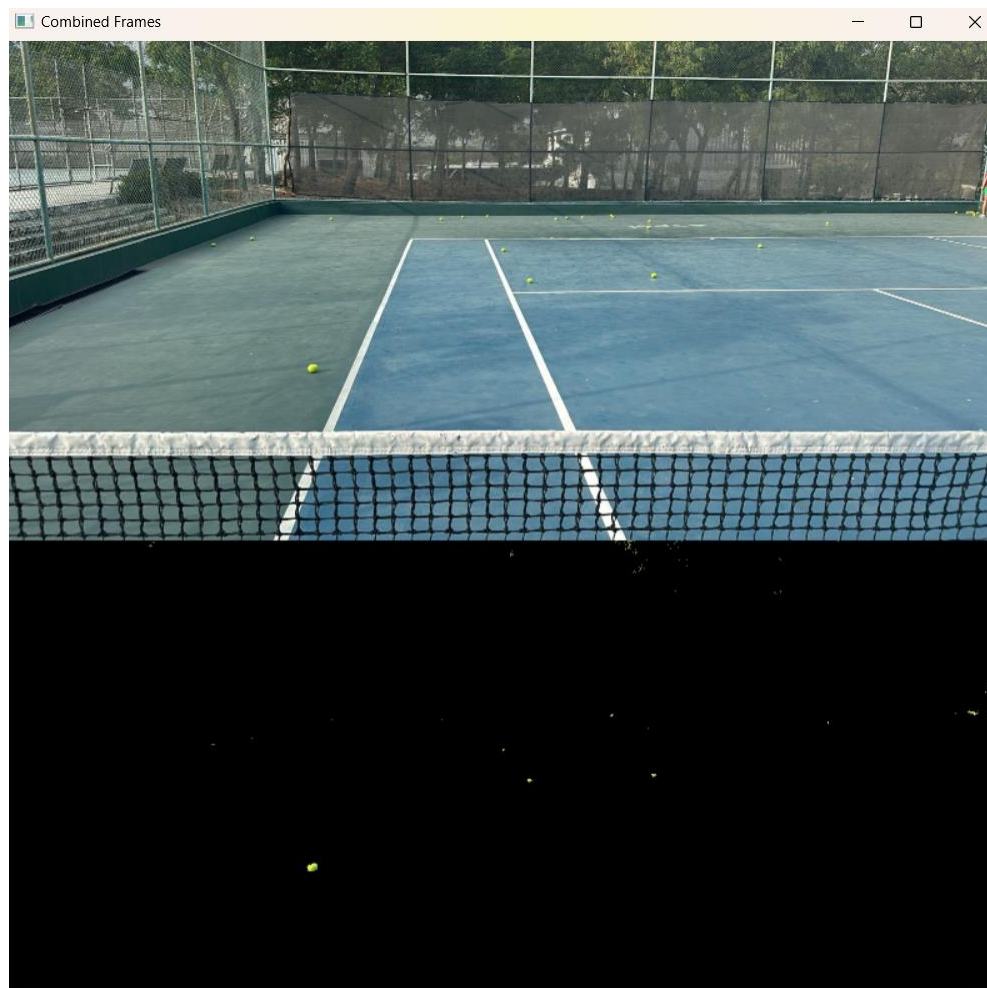
**Figura 3.12**

*Vista de funcionamiento de código de detección de color por visión por computador desde la Raspberry*



**Figura 3.13**

*Identificación de pelotas en una cancha de tenis utilizando visión por computadora*



En la sección superior de la pantalla se aprecia exclusivamente la detección de la pelota de tenis, mientras que en la parte inferior se muestra la totalidad del entorno capturado por la cámara. Esta distinción se debe al procesamiento que se lleva a cabo en la pantalla superior, enfocándose únicamente en la detección de colores, específicamente en el color de la pelota de tenis, como se puede observar de manera efectiva.

El código correspondiente a este procedimiento se detalla en el Apéndice B. Este proceso de visualización se realiza por computadora, aprovechando las capacidades ya mencionadas instaladas en la Raspberry.

### 3.5 Simulación de detección de objetos de una cámara panorámica.

En esta sección, se busca evaluar el rendimiento de una cámara ubicada en la parte superior y con visión panorámica de toda la cancha. Con este fin, se emplearon conjuntos de datos preexistentes para la detección de pelotas de tenis, haciendo uso de Google Colab. El objetivo final es obtener archivos en el formato YOLOv5, para lo que se realizó un proceso de entrenamiento para obtener el archivo best.py.

**Figura 3.14**

*Entrenamiento en Google Colab*

```

[ ] # train yolov5s on custom data for 100 epochs
# time its performance
%%time
!cd /content/yolov5/
!python train.py --img 416 --batch 16 --epochs 100 --data (dataset.location)/data.yaml --cfg ./models/custom_yolov5s.yaml --weights '' --name yolov5s_results --cache

/content/yolov5
2024-01-10 23:36:15.802141: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:920] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has already been registered
2024-01-10 23:36:15.802197: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has already been registered
2024-01-10 23:36:15.803536: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered
2024-01-10 23:36:16.812692: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
Train: weights=, cfg=./models/custom_yolov5s.yaml, data=/content/yolov5/Tennis-Ball-Detection-2/data.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=100, batch_size=16, imgsz=416, rect=False, resume=False, nosave=False, noval=False
github: # YOLOv5 is out of date by 635 commits. Use 'git pull' or 'git clone https://github.com/ultralytics/yolov5' to update.
YOLOv5 v6.1.306-gfba67e4 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)

hyperparameters: lr=0.01, lr_f=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=0.05, cls=0.5, cls_pw=1.0, obj=1.0, obj_pw=1.0, iou_t=0.2, anchor_t=4.0, fl_gamma=0.0, hsv_h
Weights & Biases: run 'pip install wandb' to automatically track and visualize YOLOv5 # runs (RECOMMENDED)
TensorBoard: Start with 'tensorboard --logdir runs/train', view at http://localhost:6006/
Download: https://ultralytics.com/assets/datal.ttf to /content/.conda/ultralitics/datal.ttf

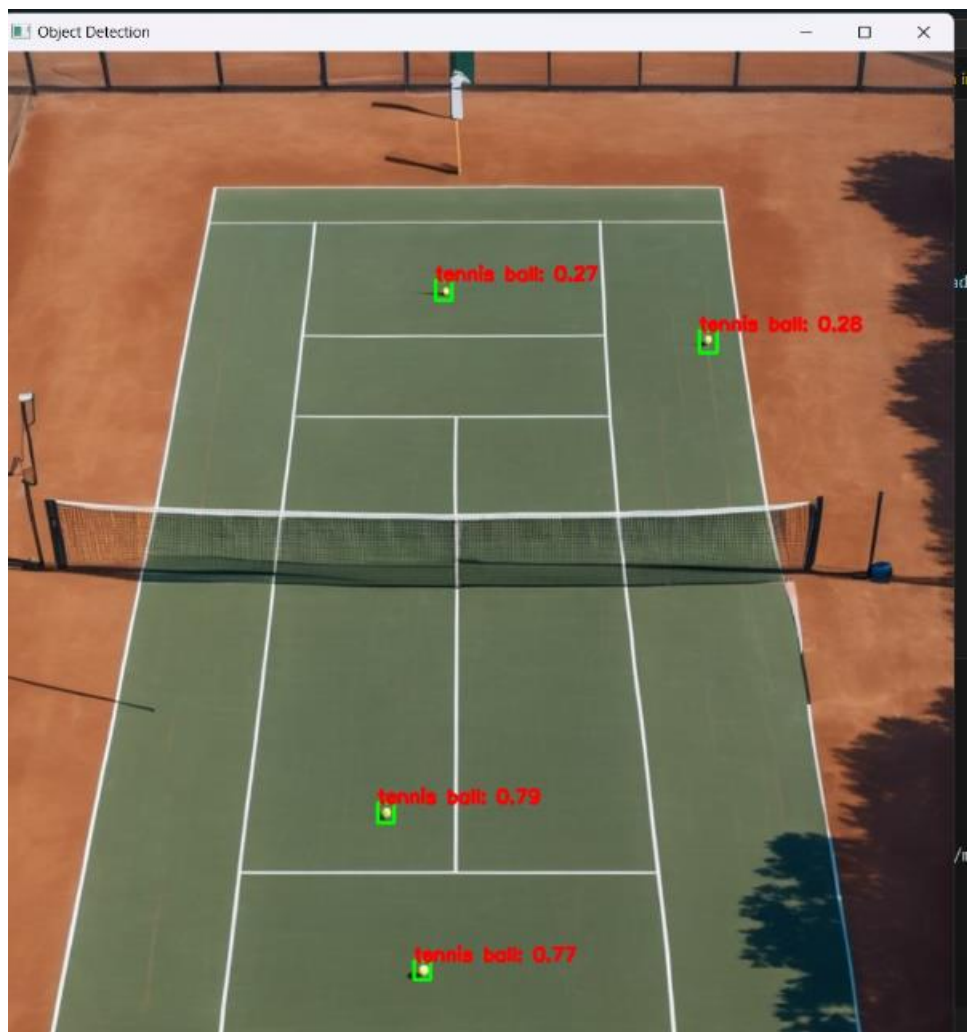
```

Posteriormente, con la obtención de dicho archivo y una fotografía capturada desde una perspectiva distante de la cancha, se procedió a desarrollar el código destinado a la detección de

objetos pertenecientes a la clase *tennis ball*. En esta primera fase, el enfoque se centra exclusivamente en la detección de estas pelotas de tenis en la cancha.

**Figura 3.15**

*Funcionamiento de primera fase del código*

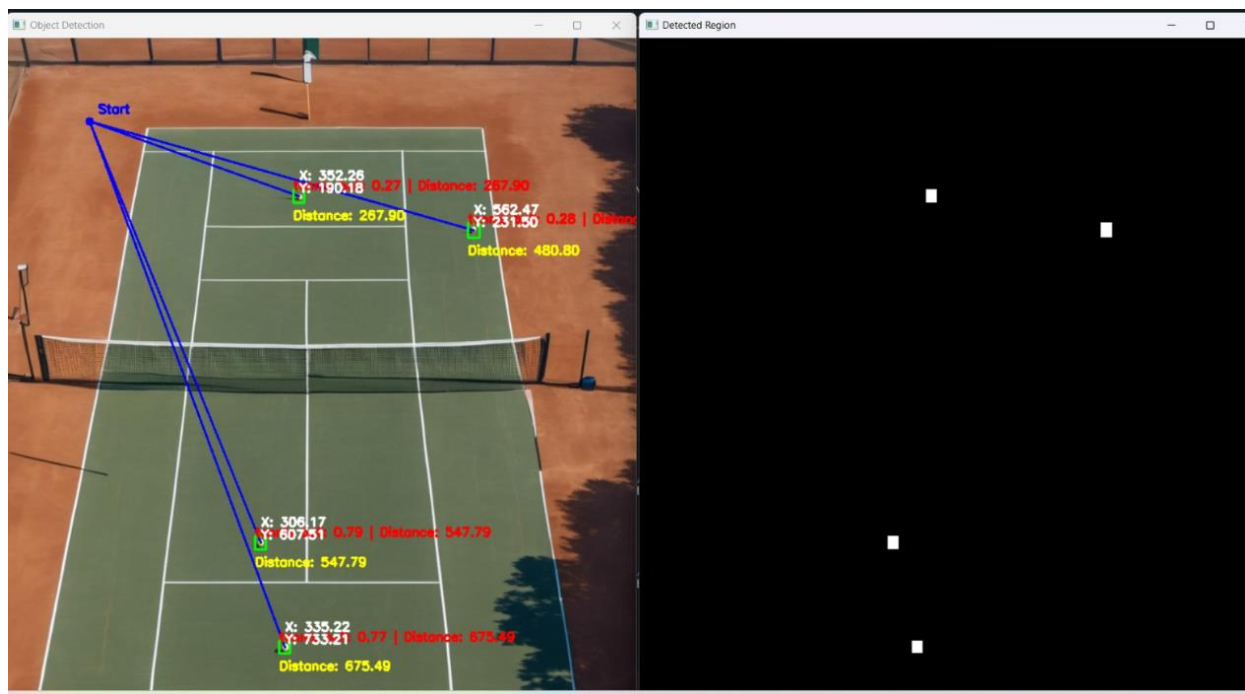


En la **figura 3.15** se presenta el resultado obtenido mediante la ejecución del primer código diseñado para la detección de pelotas. Después, se realizaron ajustes y modificaciones en los

parámetros para determinar la distancia a la que se encuentran los objetos detectados desde un punto específico.

**Figura 3.16**

*Funcionamiento del código con distancias*



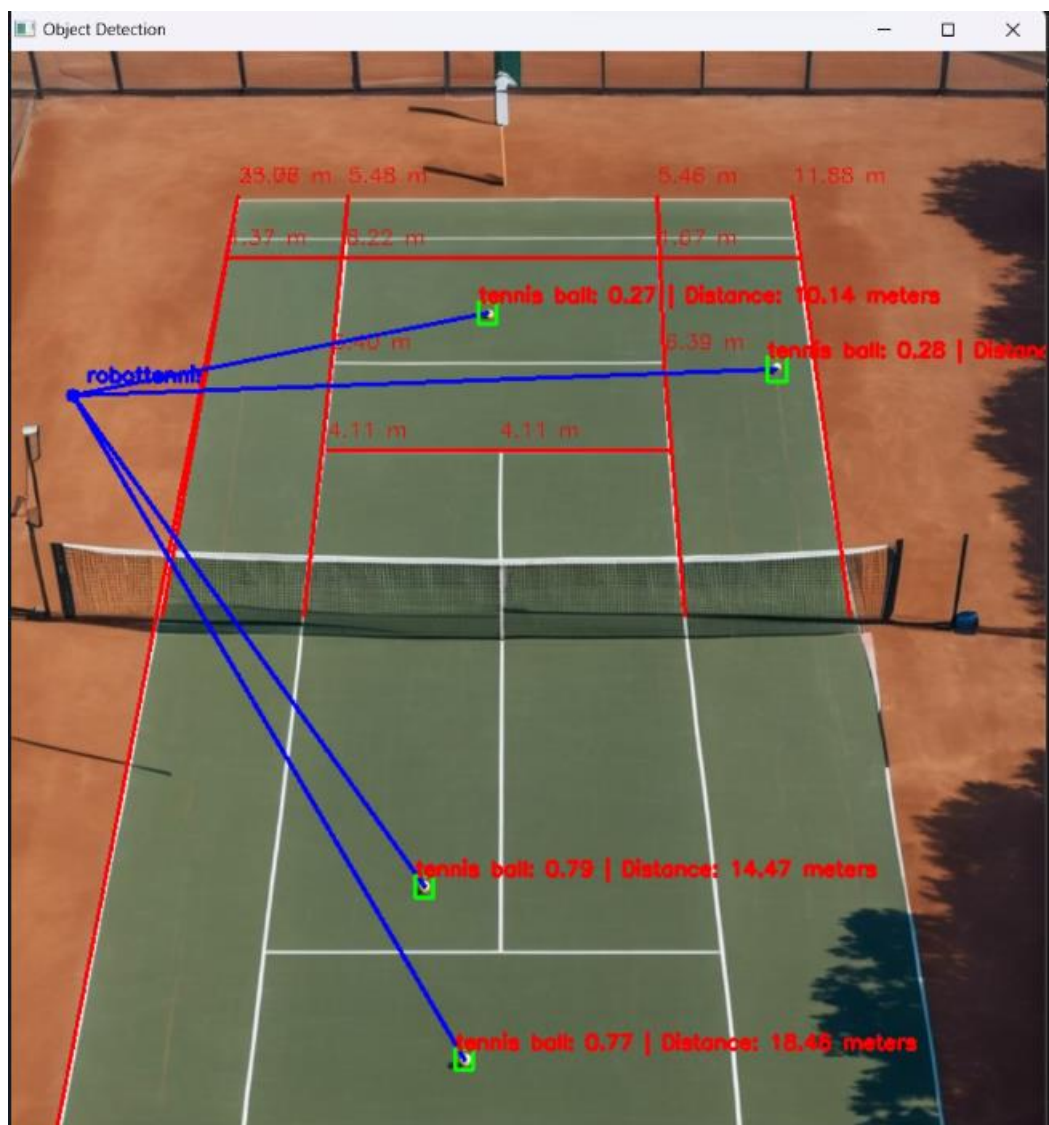
Se eligió un punto arbitrario como punto de inicio para evaluar el funcionamiento del código, proporcionando coordenadas tanto en el eje x como en el eje y. Esto permitió determinar la distancia en píxeles desde el punto de inicio hasta cada una de las pelotas detectadas.

Posteriormente, para obtener distancias en metros, se procedió a crear líneas en la cancha con mediciones de cada segmento como referencia al mundo real. Se llevaron a cabo todas las mediciones necesarias para realizar cálculos precisos, contando con estas referencias. Además, se estableció un punto táctico donde se ubicaría el robot para llevar a cabo su tarea de recolección, y

desde ese punto se midieron las distancias. Las líneas se trazaron en línea recta desde el punto de origen del robot hacia cada objeto detectado, como se ilustra en la **figura 3.17**.

**Figura 3.17**

*Funcionamiento del código con medidas de la cancha y distancias en metros*



Como se aprecia en la figura, se trazaron las líneas necesarias hacia los objetos, dando la distancia directa a cada uno de ellos. Los valores de 0.27, 0.26, 0.77 y 0.79 representan la



confianza que tiene nuestra detección de que los objetos identificados son pelotas de tenis. El objetivo futuro será plantear la formulación para comunicar esta cámara panorámica con el robot, y con estas distancias, diseñar una ruta óptima hacia cada punto, optimizando así el desplazamiento del robot. Los códigos correspondientes se encuentran detallados en el **Apéndice B**.

### **3.6 Análisis de costos**

Para implementar este proyecto, hay que adquirir los componentes tanto en la parte mecánica como en la electrónica, garantizando el funcionamiento efectivo del robot.

En la **tabla 3.1** se incluye todos los elementos del proyecto junto con sus respectivos costos, abarcando desde la cámara utilizada por el robot móvil hasta los diversos sensores y otros componentes necesarios.

Cabe destacar que, según las preferencias del cliente, sería factible tomar decisiones sobre la elección de una cámara u otros sensores alternativos para detectarlo, aunque podría conllevar modificaciones en segmentos específicos del diseño. Cabe señalar que se emplearon elementos de fácil adquisición; no obstante, se podría considerar la implementación de versiones más sofisticadas según las exigencias específicas del proyecto.

**Tabla 3.1***Costos de inversión para el proyecto*

<b>Modelo proyecto</b>	<b>Costo</b>
Chasis	\$25.00
Raspberry Pi 3 Model B	\$45.00
Sensores ultrasonido hc-sr04	\$10.00
Cámara	\$20.00
Impresiones en PLA	\$200.00
Parades laterales de acrílico	\$250.00
Sensores de velocidad	\$16.00
Puentes H	\$6.00
Batería Li-PO 11.1V- 1800mAh	\$59.60
Rodamientos	\$14.00
Pernos y tuercas	\$20.00
<b>TOTAL</b>	<b>\$665.60</b>

Es válido acotar que la tabla previamente vista considera únicamente el costo de los materiales, no obstante, el precio final del robot ya considerando el costo de ensamblaje de los componentes y un margen de ganancia del 30% por conceptos de ingeniería y diseño, sería de USD 980.0.



## Capítulo 4

## 4. Conclusiones y recomendaciones

### 4.1 Conclusiones

Debido a la expansión de la robótica en varios sectores, este diseño puede ayudar a abrir una rama más en el uso de la robótica en el deporte, específicamente en el tenis, haciendo que las prácticas de este deporte sean más fluidas, optimizando el tiempo.

- Se llevó a cabo el modelado del robot recolector usando el software Autodesk Inventor 2023, con un diseño de mecanismo de tres paletas que permiten un proceso de recolección óptimo. Cada paleta puede llevar de 1 a 2 pelotas hacia el depósito con un solo movimiento.
- El diseño propuesto se distingue por su accesibilidad económica, facilidad de transporte y una autonomía que abarca aproximadamente una hora de operación continua.
- El diseño está construido con materiales de bajo costo, con un precio total de alrededor de USD 980, siendo así 56% más económico en comparación con sus competencias en el mercado como el Tennibot cuyo precio ronda los USD 2200.
- Se logró la implementación efectiva del control del mecanismo de recolección mediante la incorporación de un microcontrolador Raspberry Pi 3B y motores. Este enfoque permitió una gestión óptima de recolección.
- Con respecto a la programación, se realizó una simulación centrada en la detección de colores y se implementó un sistema de reconocimiento de pelotas de tenis. Este aspecto del proyecto contribuye significativamente a la autonomía y capacidad de respuesta del robot, asegurando una identificación de las pelotas a recolectar.

- El sistema de visión panorámica para una cancha de tenis usando una cámara, permite identificar cuantas pelotas se encuentran dentro del campo de juego.
- El proyecto representa avances notables en la aplicación de la robótica en el tenis. Las soluciones propuestas tienen potencial para ser utilizadas en otros contextos, y sugieren mejoras y expansiones posibles en futuros desarrollos tecnológicos relacionados con la robótica en el ámbito deportivo.

## 4.2 Recomendaciones

En este proyecto se realizó el diseño de un robot móvil, pero algunos aspectos están abiertos a mejoras, entre ellos:

- Considerando las actividades realizadas durante un entrenamiento, considerando los impactos que puede recibir el sistema y si esto afecta al funcionamiento del diseño, se debería cambiar el material de construcción que proteja la cámara y los sensores.
- Realizar un sistema de comunicación para que la cámara superior envíe la información de un *spline* hacia el robot autónomo, de esta manera poder crear un sistema de ruta para que el robot se aproxime de forma rápida hacia la posición de la pelota.

- Enfocar la perspectiva hacia el futuro implica la consideración de la implementación de múltiples robots recolectores que operen de manera coordinada. Siguiendo el paradigma de las tecnologías inspiradas en los paneles de abejas, ampliamente usadas hoy, se puede expandir significativamente el alcance del proyecto. Esta expansión no solo se limitaría al ámbito del entrenamiento, sino que también contemplaría su aplicabilidad en contextos más amplios, como eventos o competiciones.
- Investigar y desarrollar estrategias para optimizar la duración de la batería, permitiendo una operación continua del robot por períodos más extensos, lo que puede ser crucial en situaciones de entrenamiento prolongado.
- Realizar un prototipo que permita validar y probar las capacidades de movimiento y recolección del robot diseñado. De esta manera, se puede realizar ajustes a la propuesta.

## Referencias

- [1] Ibañez, Á. (2021). Tecnología punta para que las personas con problemas de visión disfruten de las retransmisiones deportivas - Tecvolución. Tecvolución. <https://tecvolucion.com/tecnologia-problemas-vision-retransmisiones-deportivas/>
- [2] International Tennis Federation. (2017, p.37). Reglas del tenis,
- [3] Rorabaugh, Britt. (1999). Mechanical Devices for the Electronics Experimenter
- [4] Domínguez, R. G. (2018). Tennibot, el robot Roomba que recoge tus pelotas en tus entrenamientos de tenis. Meristation. [https://as.com/meristation/2018/05/28/betech/1527536528\\_971843.html](https://as.com/meristation/2018/05/28/betech/1527536528_971843.html)
- [5] Future Sports Academies. (2022). Tennis Lessons Singapore | Tennis Academy Singapore | Tennis Clubs. <https://futuresportsacademies.com/>
- [6] Autodesk. (2023). Robots Deportistas que mejoran la IA. Autodesk Journal. <https://www.autodeskjournal.com/robots-deportistas-mejoran-ia>
- [7] Banks, T. (2023). 7 robots que juegan deportes sorprendentemente bien. TecnoBreak | Ofertas y Reviews. <https://www.tecnobreak.com/7-robots-que-juegan-deportes-sorprendentemente-bien/>
- [8] Burt, M. (2019). CUE3: Toyota's new Record-Breaking Basketball robot - Toyota Media Site. Toyota Media Site. <https://media.toyota.co.uk/cue3-toyotas-new-record-breaking-basketball-robot/#lg=1&slide=0>
- [9] DONIC RoboPong 3050 XL.Sports Madison. <https://tenismesa.es/es/mesas-y-redes/robots-ping-pong/donic-robopong-3050-xl.html#datetime>

- [10] «Ollero Baturone, Aníbal. (2001). Robótica, manipuladores y robots móviles.pdf». Accedido: 21 de octubre de 2023. [En línea]. Disponible en: <http://el.uy/doc/robotica/Ollero%20Baturone,%20An%C3%ADbal.%202001.%20Rob%C3%B3tica,%20manipuladores%20y%20robots%20m%C3%B3viles.pdf>
- [11] The Carnegie Mellon University Autonomous Land Vehicle Project (NAVLAB). (n.d.). Carnegie Mellon University. <https://www.cs.cmu.edu/afs/cs/project/alv/www/>
- [12] T-MATIC. (2023). <https://www.linde-mh.es/es/Productos/Carretillas-automatizadas/T-Matic/>
- [13] Robot SR1. (n.d.). [http://www.superrobotica.com/sr1\\_robot.htm](http://www.superrobotica.com/sr1_robot.htm)
- [14] Figure 3. The Auriga- $\alpha$  mobile robot. (n.d.). ResearchGate. [https://www.researchgate.net/figure/The-Auriga-a-mobile-robot\\_fig2\\_4121845](https://www.researchgate.net/figure/The-Auriga-a-mobile-robot_fig2_4121845)
- [15] KUKA AG. (2019, April 10). KUKA omniMove desplaza en un espacio mínimo componentes pesados para la construcción del Airbus A380. <https://www.kuka.com/es-es/sectores/base-de-datos-de-soluciones/2016/07/solution-robotics-airbus>
- [16] Borenstein, H. R. Everett, L. Feng, & University of Michigan. (1996). Where am I? Sensors and Methods for Mobile Robot Positioning. <https://wwwpersonal.umich.edu/~johannb/Papers/pos96rep.pdf>
- [17] Figura 2 -Modelo do sistema de odometria do robô. (n.d.). ResearchGate. [https://www.researchgate.net/figure/Figura-2-Modelo-do-sistema-de-odometria-do-robo\\_fig2\\_343646784](https://www.researchgate.net/figure/Figura-2-Modelo-do-sistema-de-odometria-do-robo_fig2_343646784)

- [18] <https://aviacionparatodos1.blogspot.com/2016/06/el-sistema-de-navegacion-inercial.html>
- [19] Servo-User. (2023). Cuál es la diferencia entre un encoder incremental y absoluto? Servomotors. <https://www.servomotorsadjust.com/cual-es-la-diferencia-entre-un-encoder-incremental-y-absoluto/>
- [20]Universidad Nacional de San Juan. (s. f.). Procesamiento Digital de Imágenes. <http://dea.unsj.edu.ar/imagenes/recursos/Capitulo1.pdf>
- [21] Cornieles, P. (2021). Cómo funciona la visión artificial o Computer Vision - IA Latam. IA Latam. <https://ia-latam.com/2021/08/18/como-funciona-la-vision-artificial-o-computer-vision/>
- [22]Pereira, C. a. P., León, G. M. P., Hernández, A. I. M., & González, R. O. (2016). Procesamiento digital de imágenes: Determinación del color en muestras de alimentos y durante la. ResearchGate. [https://www.researchgate.net/publication/312605106\\_Procesamiento\\_digital\\_de\\_imagenes\\_Determinacion\\_del\\_color\\_en\\_muestras\\_de\\_alimentos\\_y\\_durante\\_la\\_maduracion\\_de\\_frutos](https://www.researchgate.net/publication/312605106_Procesamiento_digital_de_imagenes_Determinacion_del_color_en_muestras_de_alimentos_y_durante_la_maduracion_de_frutos)
- [23] Muñoz, J. (2020, July 29). La Física y la percepción de la velocidad de las pistas. Puntodebreak. <https://www.puntodebreak.com/2014/04/18/la-fisica-y-la-percepcion-de-la-velocidad-de-las-pistas>
- [24] Robot Car Kit 4WD | Joy-IT. (n.d.). <https://joy-it.net/en/products/Robot03>
- [25] Ultra-lab. (2023, November 9). Sensor Ultrasonido HC-SR04 - Sensor de distancia - Ultra-lab. <https://ultra-lab.net/producto/sensor-ultrasonido-hc-sr04-sensor-de-distancia/>
- [26] Speed sensor | Joy-IT. (n.d.). <https://joy-it.net/en/products/SEN-Speed>

[27] Dewanxin Webcam 1080P Full HD CMOS Cámara Web de Alta Micrófono Reductor de Ruido y Corrección de Automática,USB Plug and Play,Base Giratoria de 360°, para PC Computadora Portátil, Videollamadas Juegos: Amazon.es: Informática. (n.d.). <https://www.amazon.es/Micr%C3%B3fono-Correcci%C3%B3n-Autom%C3%A1tica-Computadora-Videollamadas/dp/B0891YNFQT>

[28] Melopero Raspberry Pi 3 Model B, CPU Quad Core 1,2GHz Broadcom BCM2837 64bit, 1GB RAM, WiFi, Bluetooth BLE, plata : Amazon.es: Informática. (n.d.). <https://www.amazon.es/Raspberry-Pi-Modelo-Quad-Core-Cortex-A53/dp/B01CD5VC92>

[29] BATERIA LIPO 11.1V 2200mAh 3S 35C - Robotics. (2023, September 1). Robotics. <https://roboticsec.com/producto/bateria-lipo-11-1v-2200mah-3s-35c/>

[30] Administrador. (2020, February 11). Detección de colores en OpenCV – Python (En 4 pasos). OMES. <https://omes-va.com/deteccion-de-colores/>

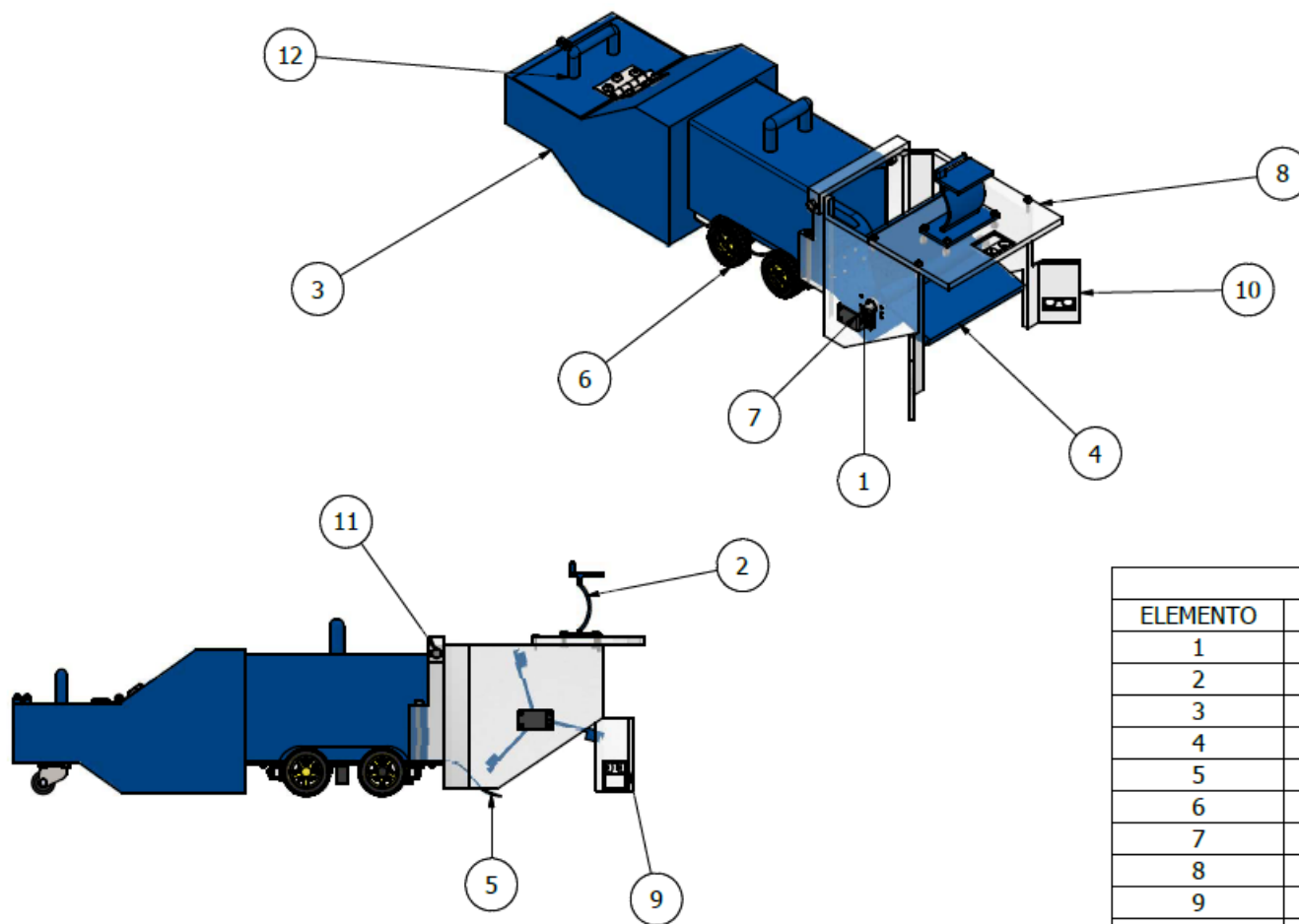
[31] Admin. (2022, June 7). How to control a DC motor using Raspberry Pi with L298N Driver. DIY Projects Lab. <https://diyprojectslab.com/raspberry-pi-with-l298n-driver/>



## **Apéndice**

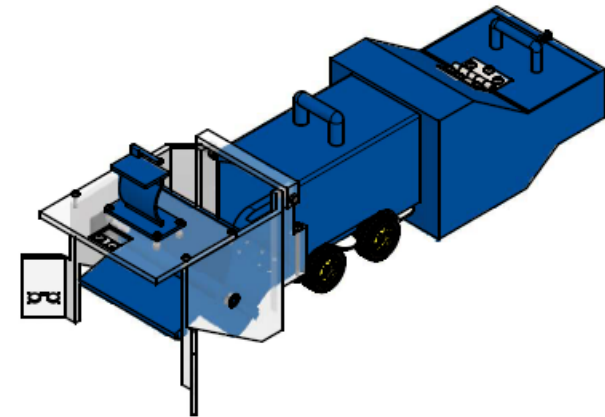
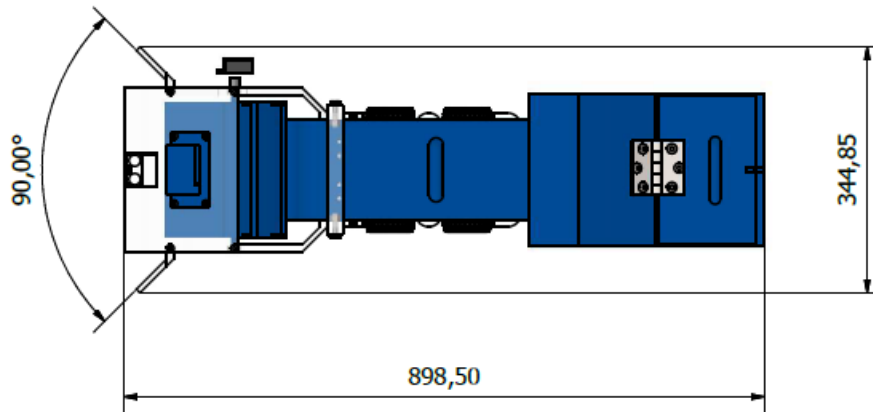
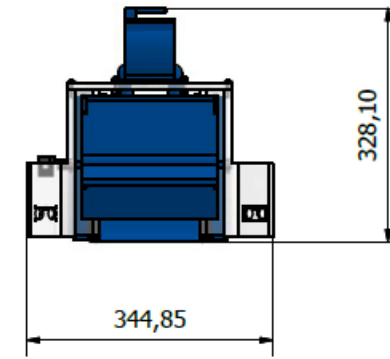
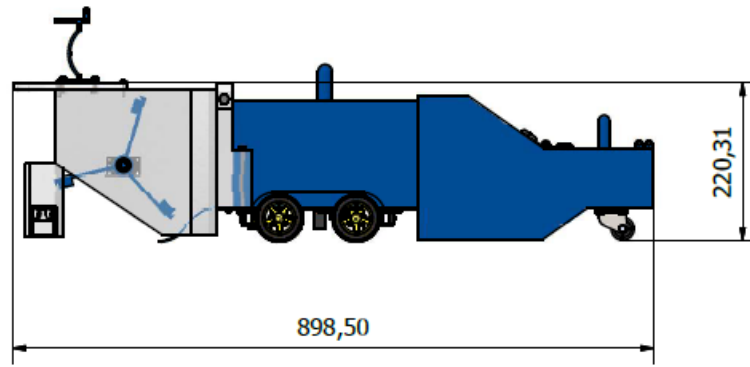
## **Apéndice A**

Planos mecánicos

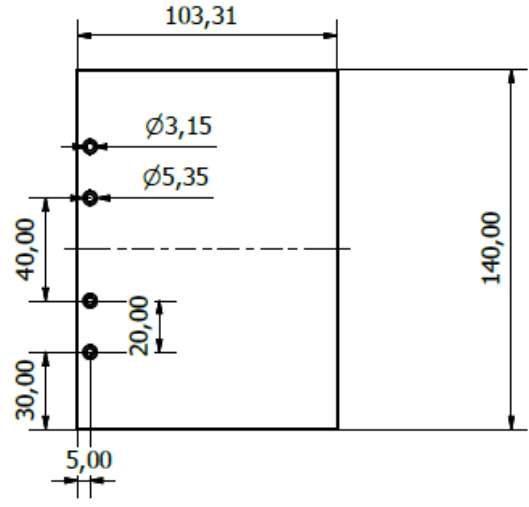
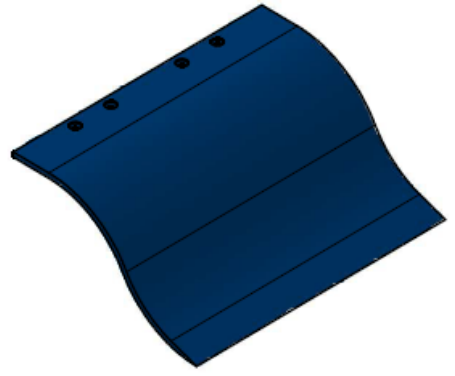
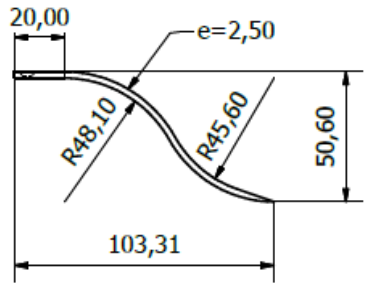
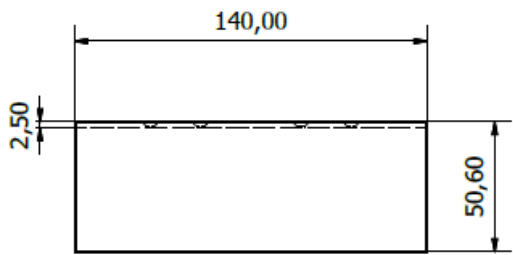


LISTA DE PIEZAS		
ELEMENTO	CTDAD	Nº DE PIEZA
1	1	Acople de Matrimonio
2	1	Base de Webcam
3	1	Depósito de pelotas
4	1	Paletas de Recolección
5	1	Rampa de recolección
6	1	Robot Kit Arduino
7	1	Soporte de Motor
8	1	Base superior
9	1	Soporte lateral derecho
10	1	Soporte lateral izquierdo
11	1	Barra de soporte superior
12	1	Tapa Trasera

Escuela Superior Politécnica del Litoral		FACULTAD DE INGENIERÍA MECÁNICA Y CIENCIAS DE LA PRODUCCIÓN			
Elaborado por:	A. Mata	Revisado:	Ing. Efraín Terán	Escala:	1 : 7
	D. Vega	Fecha:	26/01/2024	Medidas:	mm.
Fecha:	23/01/2024	Nombre de la parte:			Material:
		RARP			N/A
				Hoja:	1/13




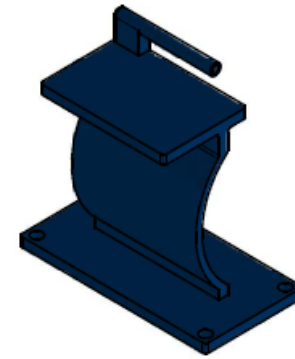
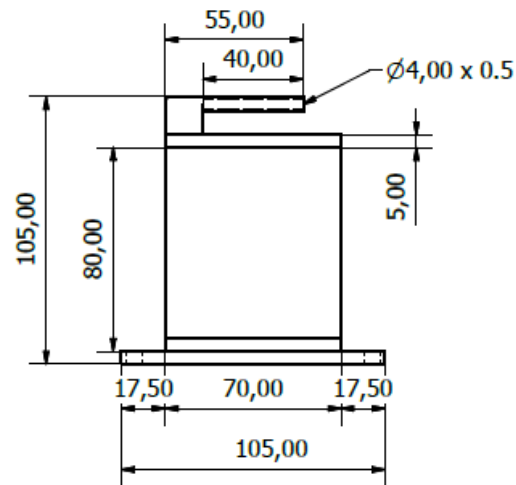
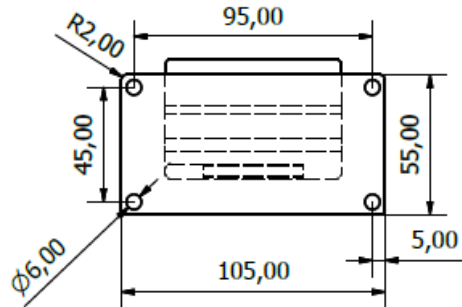
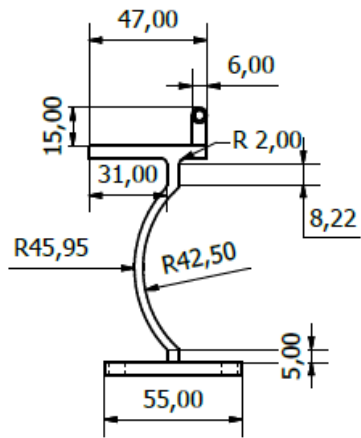
		<b>FACULTAD DE INGENIERÍA MECÁNICA Y CIENCIAS DE LA PRODUCCIÓN</b>			
Elaborado por:	A. Mata	Revisado:	Ing. Efraín Terán	Escala:	1 : 7
	D. Vega	Fecha:	26/01/2024	Medidas:	mm
Fecha:	23/01/2024	Nombre de la parte:			Material:
		RARP			N/A
					Hoja: 2/13



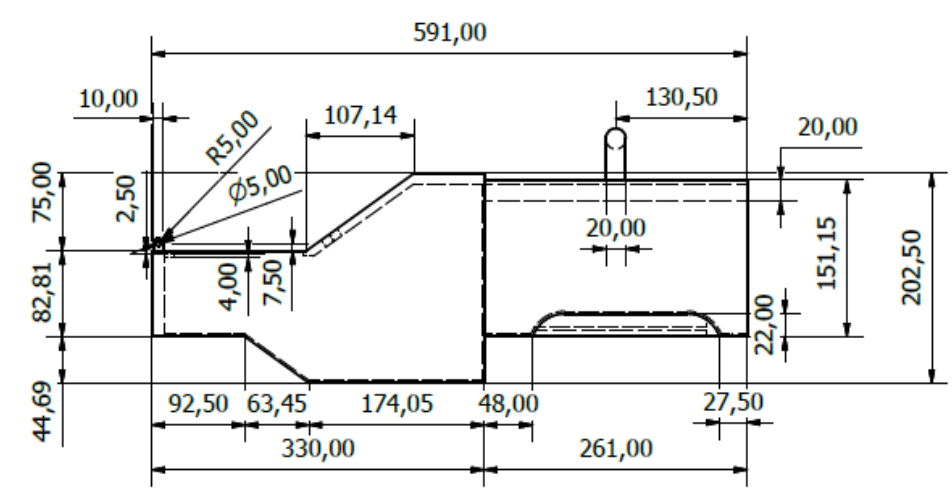
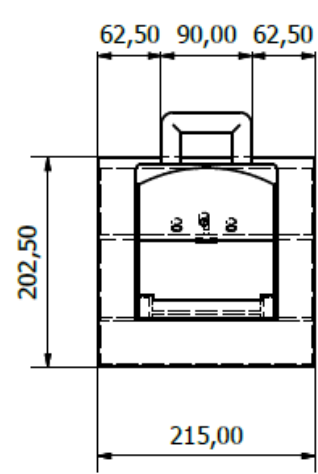
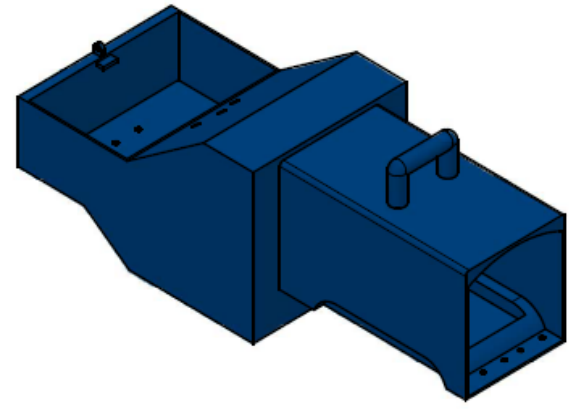
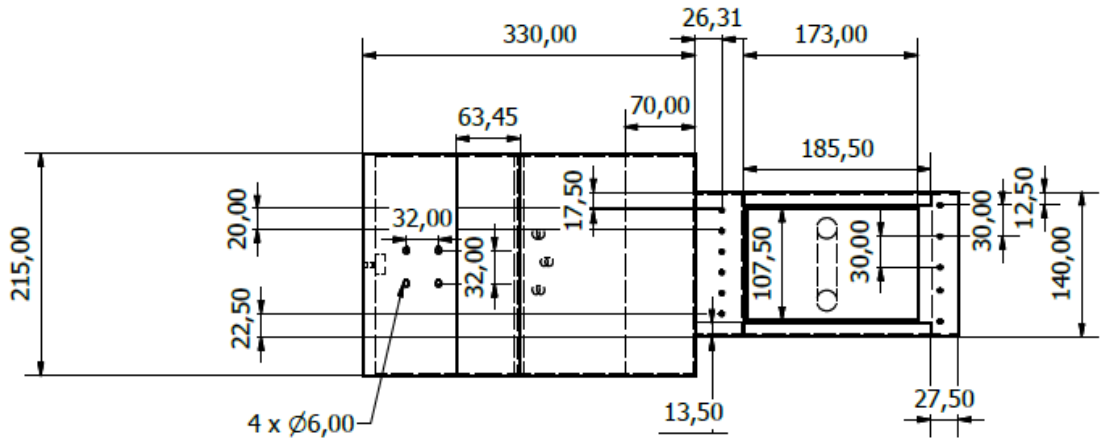
espol Escuela Superior Politécnica del Litoral

FACULTAD DE INGENIERÍA MECÁNICA Y CIENCIAS DE LA PRODUCCIÓN

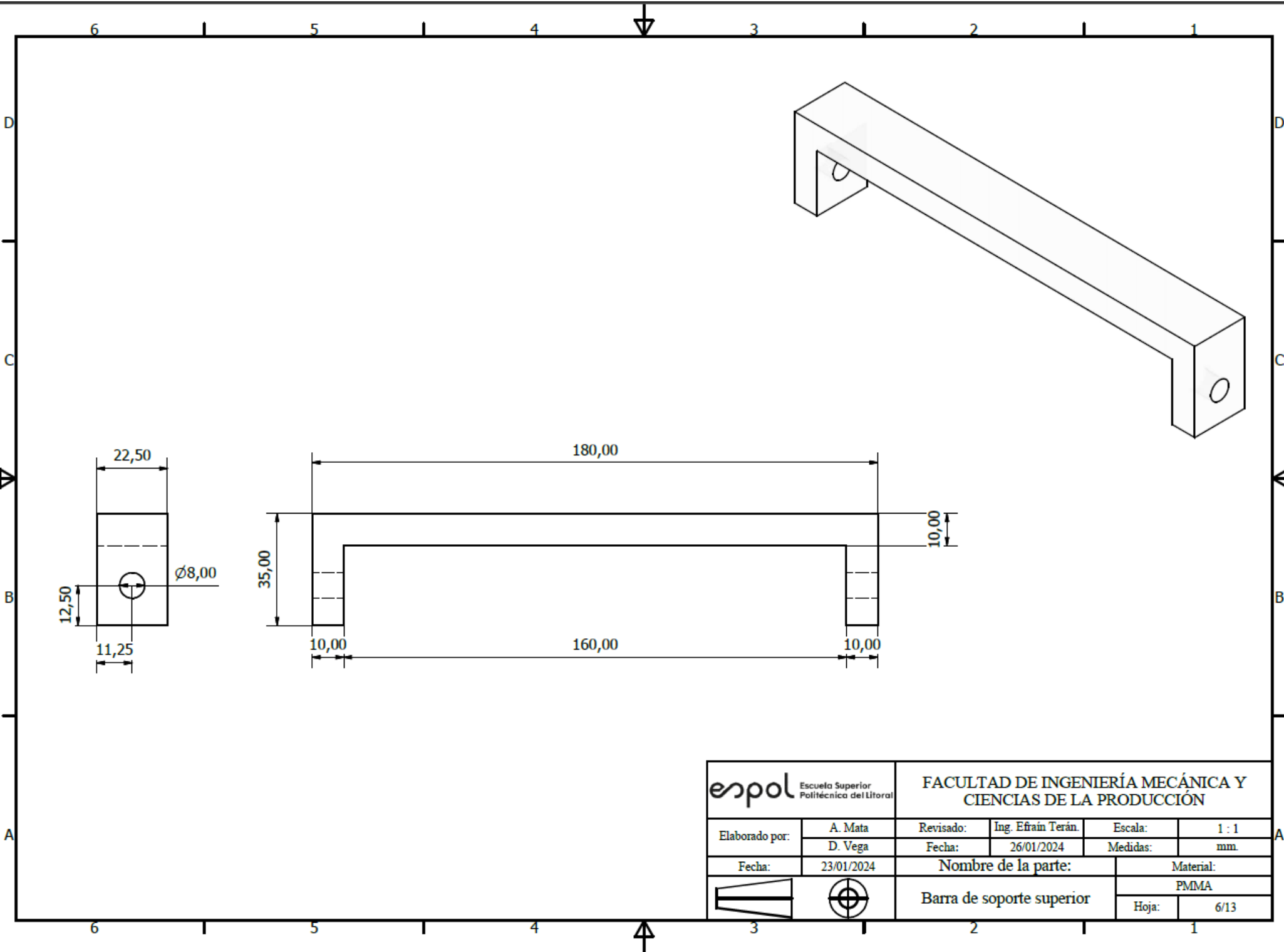
Elaborado por:	A. Mata	Revisado:	Ing. Efraín Terán	Escala:	1 : 2
	D. Vega	Fecha:	26/01/2024	Medidas:	mm.
Fecha:	23/01/2024	Nombre de la parte:		Material:	
		Rampa de recolección		PLA	
				Hoja:	3/13



Escuela Superior Politécnica del Litoral		FACULTAD DE INGENIERÍA MECÁNICA Y CIENCIAS DE LA PRODUCCIÓN			
Elaborado por:	A. Mata	Revisado:	Ing. Efraim Terán	Escala:	1 : 2
	D. Vega	Fecha:	26/01/2024	Medidas:	mm.
Fecha:	23/01/2024	Nombre de la parte:			Material:
		Base de Webcam			PLA
				Hoja:	4/13




Escuela Superior Politécnica del Litoral		FACULTAD DE INGENIERÍA MECÁNICA Y CIENCIAS DE LA PRODUCCIÓN			
Elaborado por:	A. Mata	Revisado:	Ing. Efraín Terán	Escala:	1 : 5
	D. Vega	Fecha:	26/01/2024	Medidas:	mm.
Fecha:	23/01/2024	Nombre de la parte:			Material:
		Depósito de pelotas			PLA
				Hoja:	5/13

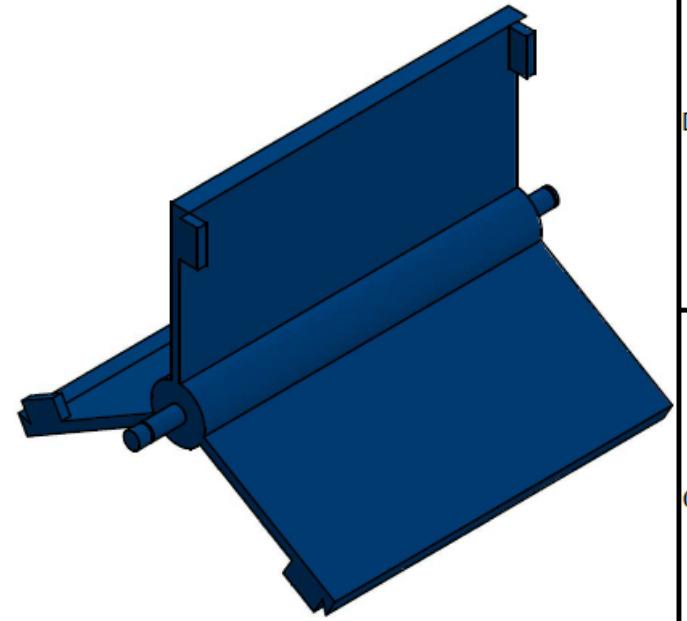
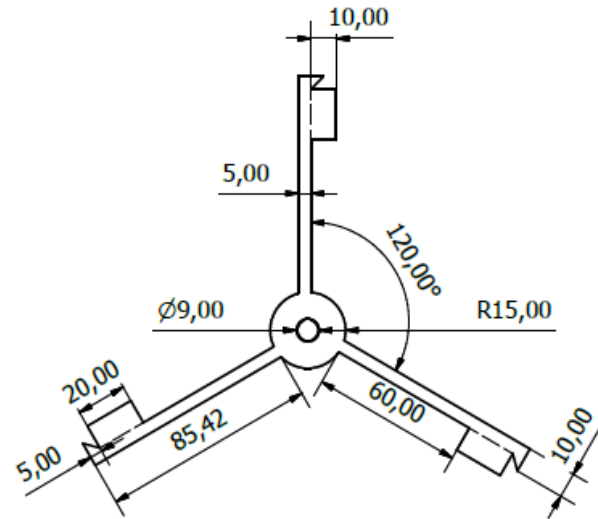
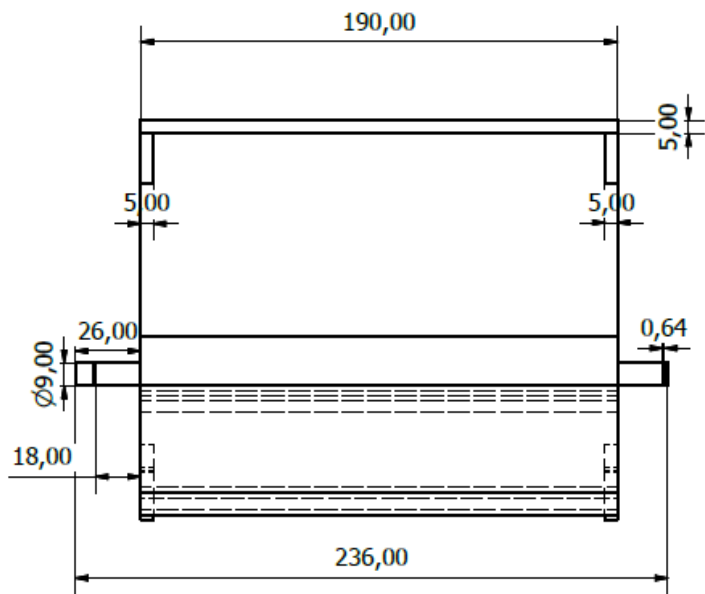


espol Escuela Superior Politécnica del Litoral

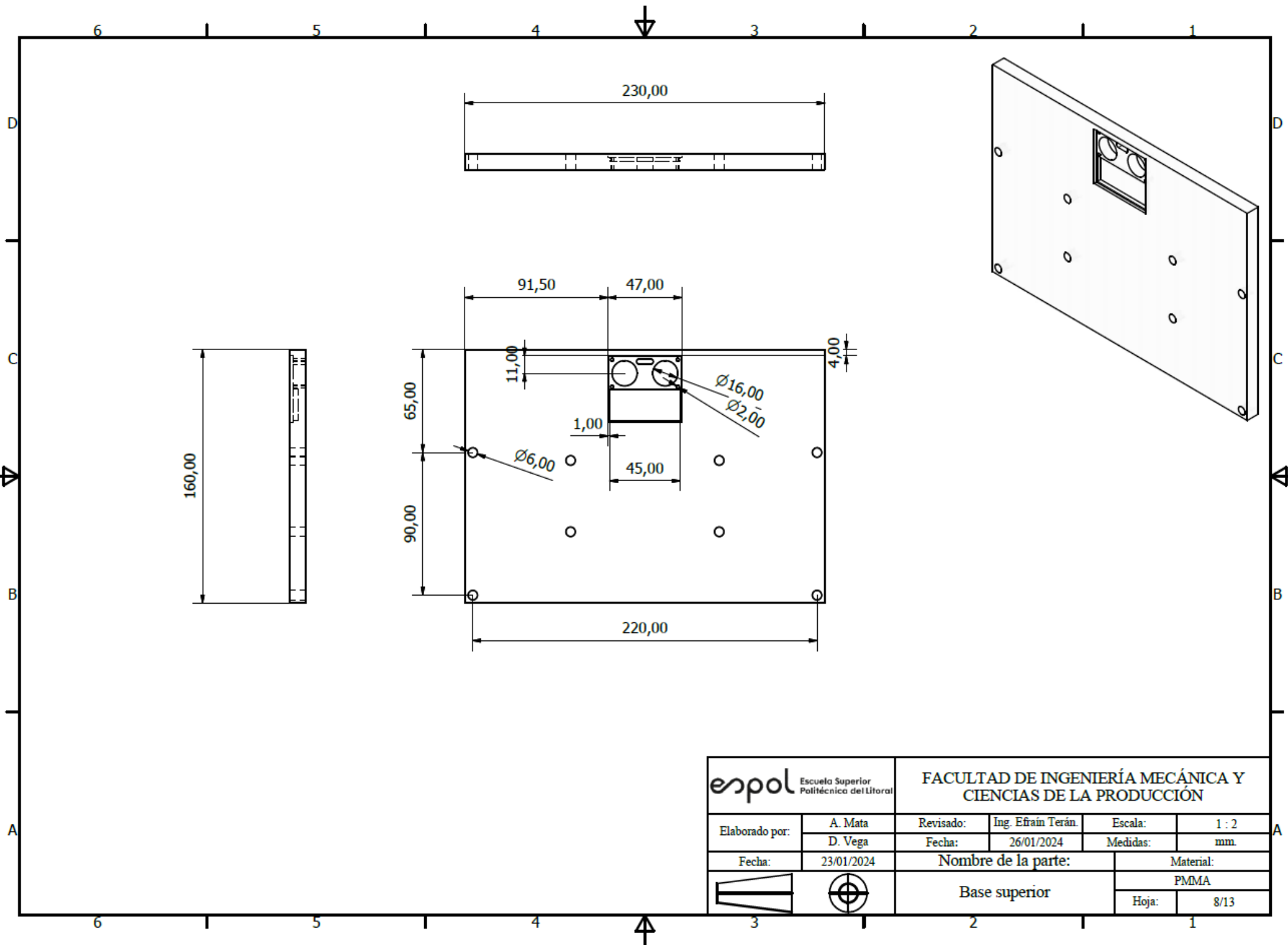
FACULTAD DE INGENIERÍA MECÁNICA Y CIENCIAS DE LA PRODUCCIÓN

Elaborado por:	A. Mata	Revisado:	Ing. Efraín Terán	Escala:	1 : 1
	D. Vega	Fecha:	26/01/2024	Medidas:	mm
Fecha:	23/01/2024	Nombre de la parte:			Material:
		Barra de soporte superior			PMMA
				Hoja:	6/13

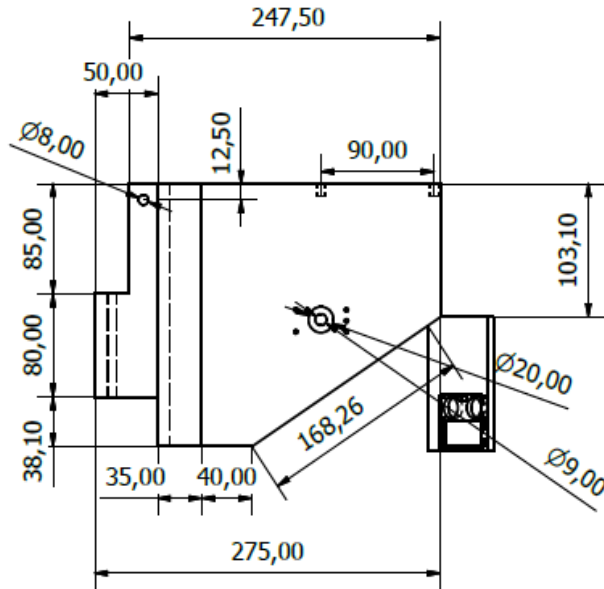
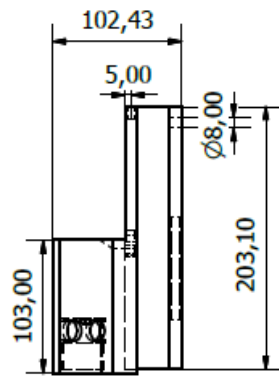
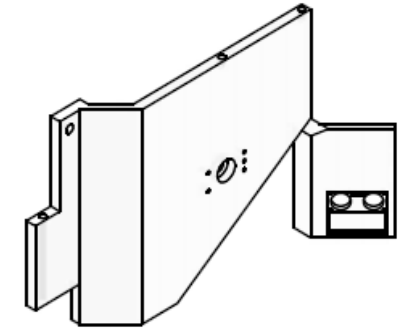
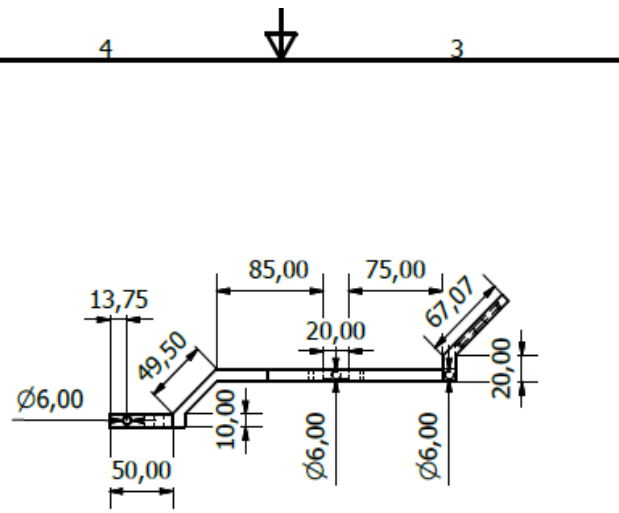






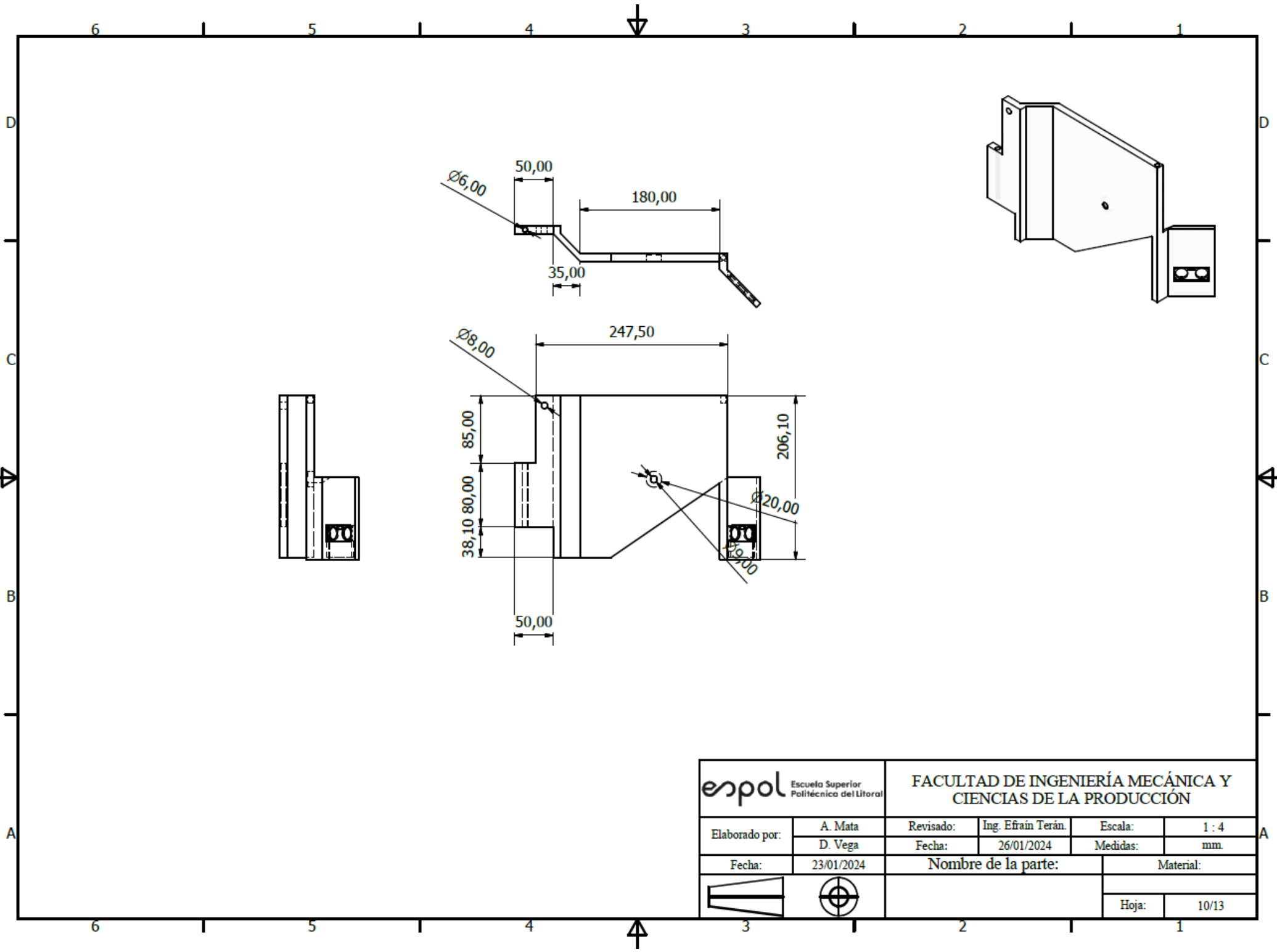
		<b>FACULTAD DE INGENIERÍA MECÁNICA Y CIENCIAS DE LA PRODUCCIÓN</b>			
Elaborado por:	A. Mata	Revisado:	Ing. Efraín Terán	Escala:	1 : 2
	D. Vega	Fecha:	26/01/2024	Medidas:	mm
Fecha:	23/01/2024	Nombre de la parte:			Material:
		Paletas de recolección			PLA
		Hoja:			7/13





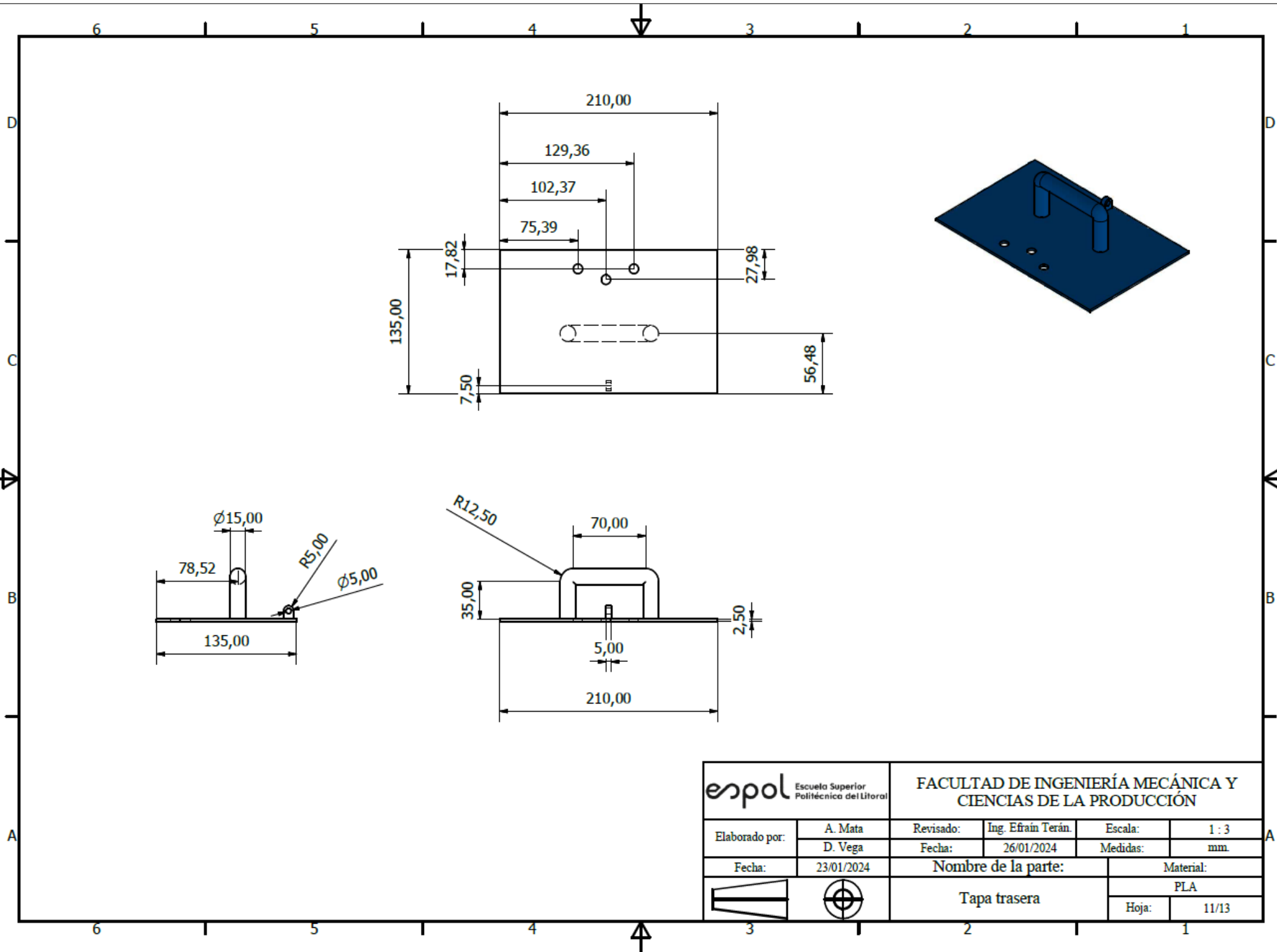
		<b>FACULTAD DE INGENIERÍA MECÁNICA Y CIENCIAS DE LA PRODUCCIÓN</b>			
Elaborado por:	A. Mata	Revisado:	Ing. Efraín Terán	Escala:	1 : 2
	D. Vega	Fecha:	26/01/2024	Medidas:	mm.
Fecha:	23/01/2024	Nombre de la parte:		Material:	
		Base superior		PMMA	
				Hoja:	8/13





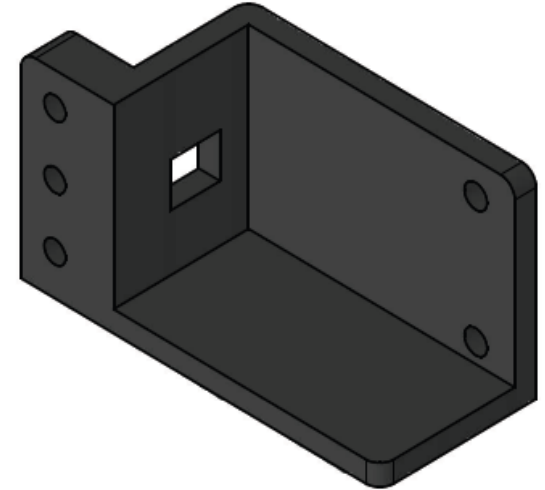
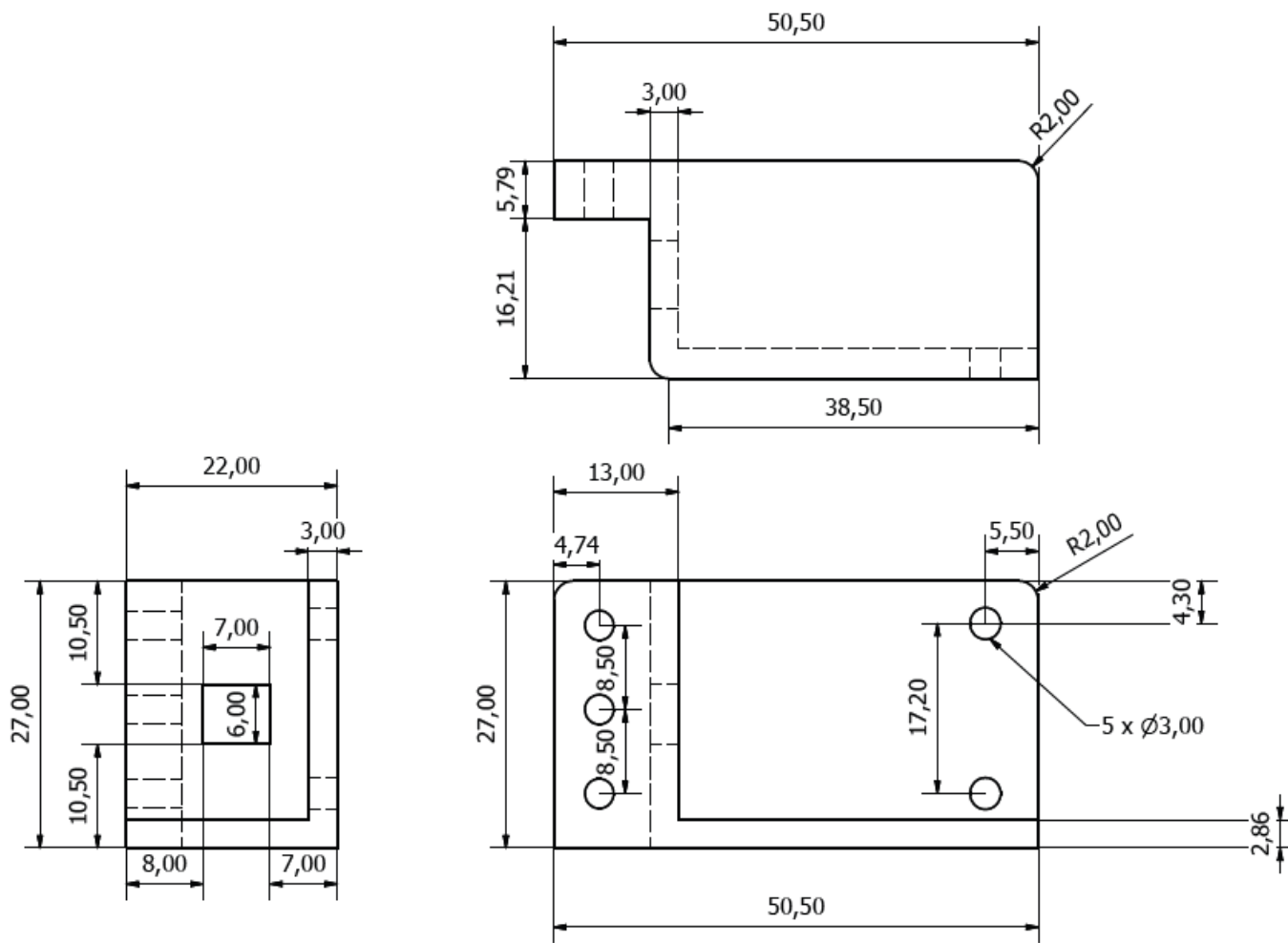
 Escuela Superior Politécnica del Litoral		FACULTAD DE INGENIERÍA MECÁNICA Y CIENCIAS DE LA PRODUCCIÓN			
Elaborado por:	A. Mata	Revisado:	Ing. Efraim Terán	Escala:	1 : 4
	D. Vega	Fecha:	26/01/2024	Medidas:	mm.
Fecha:	23/01/2024	Nombre de la parte:			Material:
		Soporte lateral derecho			PMMA
				Hoja:	9/13






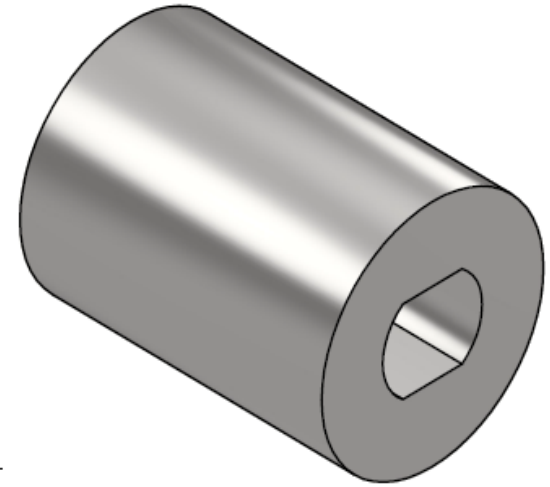
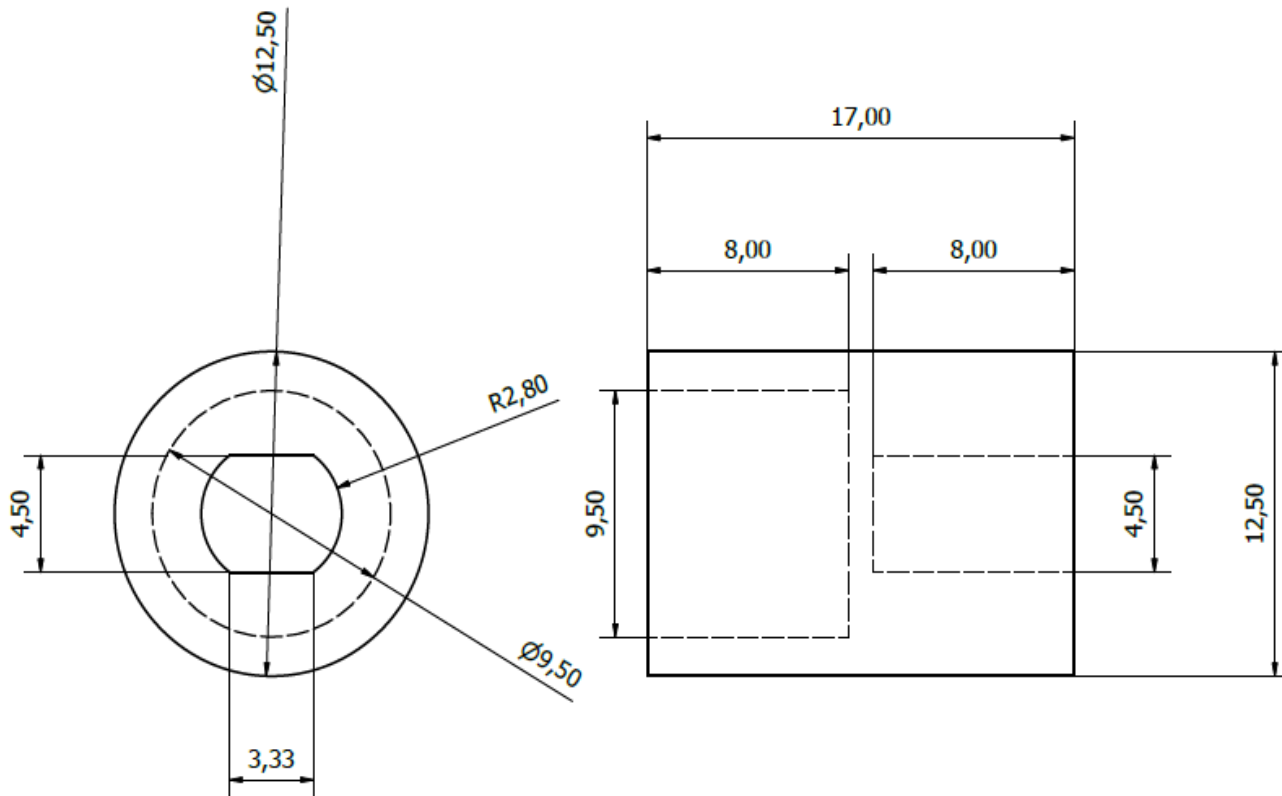
 Escuela Superior Politécnica del Litoral		FACULTAD DE INGENIERÍA MECÁNICA Y CIENCIAS DE LA PRODUCCIÓN			
Elaborado por:	A. Mata	Revisado:	Ing. Efraín Terán	Escala:	1 : 4
	D. Vega	Fecha:	26/01/2024	Medidas:	mm
Fecha:	23/01/2024	Nombre de la parte:			Material:
				Hoja:	10/13



 Escuela Superior Politécnica del Litoral		FACULTAD DE INGENIERÍA MECÁNICA Y CIENCIAS DE LA PRODUCCIÓN			
Elaborado por:	A. Mata	Revisado:	Ing. Efraín Terán	Escala:	1 : 3
	D. Vega	Fecha:	26/01/2024	Medidas:	mm
Fecha:	23/01/2024	Nombre de la parte:			Material:
		Tapa trasera			PLA
					Hoja: 11/13



 Escuela Superior Politécnica del Litoral		FACULTAD DE INGENIERÍA MECÁNICA Y CIENCIAS DE LA PRODUCCIÓN			
Elaborado por:	A. Mata	Revisado:	Ing. Efraín Terán.	Escala:	2 : 1
	D. Vega	Fecha:	26/01/2024	Medidas:	mm.
Fecha:	23/01/2024	Nombre de la parte:			Material:
 		Soporte de motor			PLA
		Hoja:			12/13



		<b>FACULTAD DE INGENIERÍA MECÁNICA Y CIENCIAS DE LA PRODUCCIÓN</b>			
Elaborado por:	A. Mata	Revisado:	Ing. Efraim Terán	Escala:	5 : 1
	D. Vega	Fecha:	26/01/2024	Medidas:	mm.
Fecha:	23/01/2024	Nombre de la parte:			Material:
		Acople de matrimonio			PLA
				Hoja:	13/13

## **Apéndice B**

### Programación

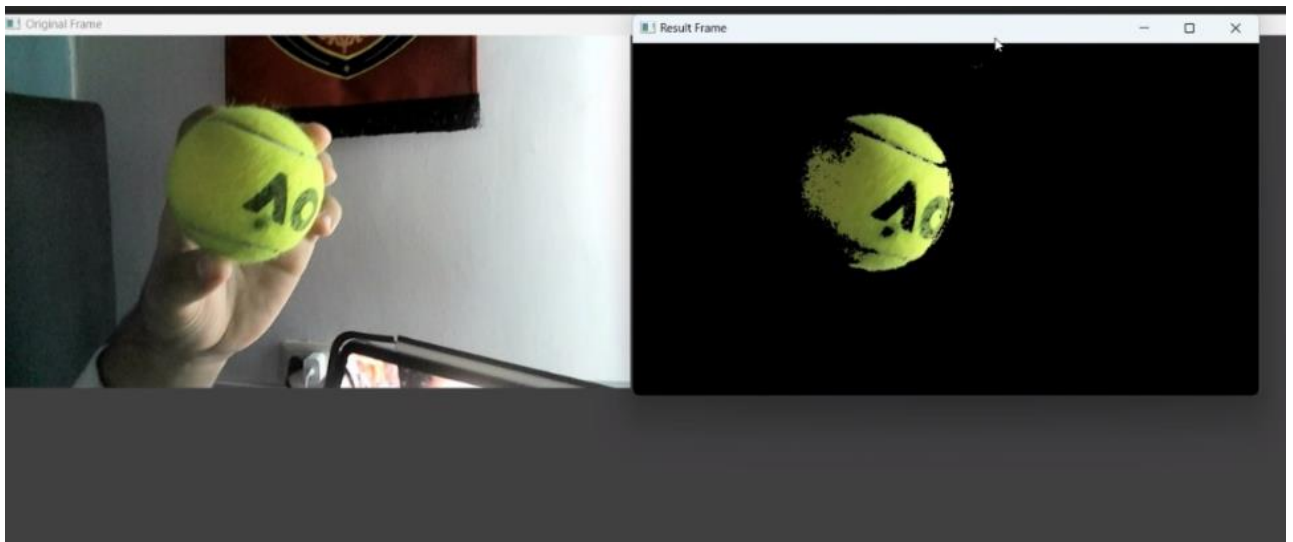


## Código de detección de color

El presente código de control está diseñado para ser implementado en Raspberry Pi 3 model B+, el lenguaje de programación es Python. El funcionamiento del código es para detectar objetos verdes como el de las pelotas de tenis, cuando se simulan dos pantallas en una se apreciará todo el ambiente de la cámara, mientras que en el otro solo estará todo negro y se mostrará el objeto que tenga los colores que detectar.

**Figura B.1**

*Ejemplo de funcionamiento del código de detección de color en un entorno.*



```
#Define una función para realizar el filtrado de color en el espacio de color HSV
def procesar_frame(frame):
    # Convierte el fotograma al espacio de color HSV
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    # Define los límites inferior y superior para el filtro de color verde
    verde_inferior = np.array([30,100,100])
    verde_superior = np.array([70,255,255])

    # Aplica el filtro de color al fotograma
    mascara = cv2.inRange(hsv, verde_inferior, verde_superior)
    resultado = cv2.bitwise_and (frame, frame, mask=mascara)

    return resultado

# Crea un objeto de captura de video para capturar el flujo de video desde la
cámara
cap = cv2.VideoCapture(0)

while True:

    # Captura el fotograma de la cámara
    ret, fotograma = cap.read()

    # Llama a la función procesar_frame para procesar el fotograma
    resultado = procesar_frame(fotograma)

    # Muestra el fotograma original y el fotograma resultante uno al lado del otro
    cv2.imshow('Fotograma Original', fotograma)
    cv2.imshow('Fotograma Resultante', resultado)

    # Verifica si el usuario ha presionado la tecla 'q'. Si es así, rompe el bucle
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Libera el objeto de captura de video y cierra todas las ventanas
cap.release()
cv2.destroyAllWindows()
```

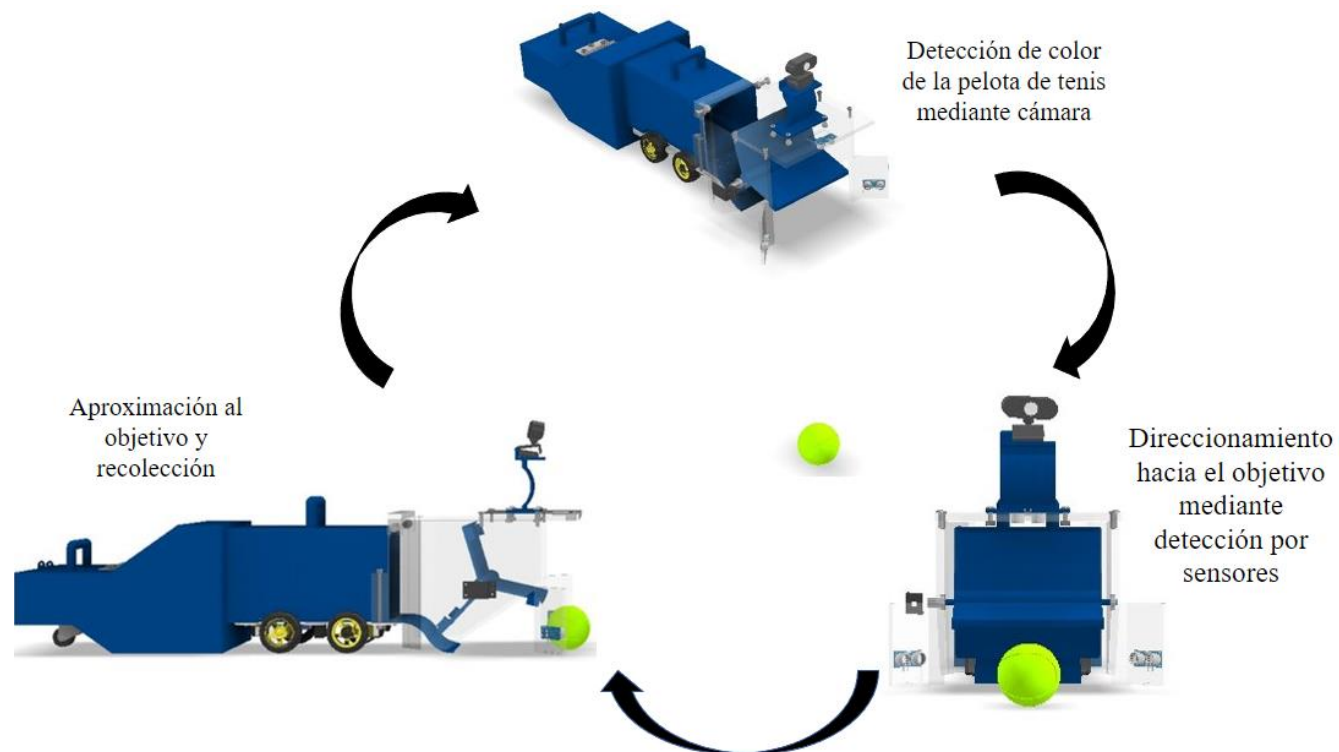
## Código de control del vehículo

El presente código fue realizado en lenguaje de programación de Python, está diseñado para operar un robot con motores y sensores ultrasónicos en una Raspberry Pi 3B+, integrando también la detección de objetos de color verde mediante una cámara, una parte clave para realizar esta parte del código, es conocer que valores pueden tomar cada uno de los canales de HSV en OpenCV: H: 0 a 179 ,S: 0 a 255 y V: 0 a 255 [30]. Los motores tendrán un funcionamiento a un robot diferencial debido que sus los motores izquierdos actuaran simultáneamente y los derechos de la misma manera para realizar estas conexiones a un puente H L298N se usó [31]. El funcionamiento se lo puede observar en la **Figura 2.19**.

El código inicia con los motores del chasis al 100% de su velocidad, inicializa los pines del motor y los sensores ultrasónicos. Luego, establece un bucle principal que constantemente captura los fotogramas de la cámara y ejecuta diversas funciones. El procesamiento de imágenes entra en acción, identificando un color predefinido y ajustando la velocidad de los motores en consecuencia. Para garantizar un movimiento más suave, la velocidad de los motores se reduce al 50%. Los sensores ultrasónicos miden distancias y, mediante un control proporcional, ajustan la posición del robot según las variaciones en la distancia. Contando con el respaldo de los sensores laterales, el robot determina el movimiento necesario para posicionar la pelota en el centro de su diseño. Además, el programa responde a la detección de una pelota mediante el sensor superior. Cuando este sensor percibe un cambio en la distancia al suelo, pasando de 21.30 cm a 15 cm, el programa mantiene los motores del chasis activos durante 2 segundos adicionales, luego los detiene brevemente. Activa el motor de las paletas durante un periodo específico de 3 segundos y posteriormente retoma el control de los motores originales. Este bucle de funciones se ejecuta de manera continua hasta que el usuario decide finalizar el programa al presionar la tecla 'q'. Este proceso asegura la liberación de recursos y la terminación ordenada del programa.

**Figura B.2**

*Representación de funcionamiento de código*



```

Import cv2.
import numpy as np
import RPi.GPIO as gpio
import time

# Definir pines del motor
motorderecho_pwm = 17
motorderecho_dir = 22
motorizq_pwm = 23
motorizq_dir = 24
motor_paletas = 25

# Definir pines del sensor ultrasónico
sensor_izq_trig = 18
sensor_izq_echo = 27
sensor_derecho_trig = 5
sensor_derecho_echo = 6
sensor_sup_trig = 13
sensor_sup_echo = 19

# Programación de detección de color
def process_frame(frame):
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    verde_inferior = np.array([30, 100, 100])
    verde_superior = np.array([70, 255, 255])
    mascara = cv2.inRange(hsv, verde_inferior, verde_superior)
    resultado = cv2.bitwise_and(frame, frame, mask=mascara)
    return resultado, mascara

# Inicializar pines GPIO del motor
def init_motors():
    gpio.setmode(gpio.BCM)
    gpio.setup(motorderecho_pwm, gpio.OUT)
    gpio.setup(motorderecho_dir, gpio.OUT)
    gpio.setup(motorizq_pwm, gpio.OUT)
    gpio.setup(motorizq_dir, gpio.OUT)
    gpio.setup(motor_paletas, gpio.OUT)

    motor_pwm1 = gpio.PWM(motorderecho_pwm, 100)
    motor_pwm2 = gpio.PWM(motorizq_pwm, 100)
    new_motor_pwm_obj = gpio.PWM(motor_paletas, 100)

    motor_pwm1.start(0)
    motor_pwm2.start(0)
    new_motor_pwm_obj.start(0)

```

```

# Función para establecer la velocidad del motor
def set_motor_speed(speed, motor_pwm):
    motor_pwm.ChangeDutyCycle(speed)

# Función para establecer la dirección del motor
def set_motor_direction(direction1, direction2):
    gpio.output(motorderecho_dir, direction1)
    gpio.output(motorizq_dir, direction2)

# Inicializar sensores ultrasónicos
def setup_ultrasonic(trig_pin, echo_pin):
    gpio.setup(trig_pin, gpio.OUT)
    gpio.setup(echo_pin, gpio.IN)

# Medir la distancia utilizando el sensor ultrasónico
def measure_distance(trig_pin, echo_pin):
    gpio.output(trig_pin, True)
    time.sleep(0.00001)
    gpio.output(trig_pin, False)

    pulse_start = time.time()
    pulse_end = time.time()

    while gpio.input(echo_pin) == 0:
        pulse_start = time.time()

    while gpio.input(echo_pin) == 1:
        pulse_end = time.time()

    pulse_duration = pulse_end - pulse_start
    distance = pulse_duration * 17150
    distance = round(distance, 2)

    return distance

# Control proporcional para ajustar la posición basada en la diferencia de
distancias
def adjust_position(left_distance, right_distance, target_distance, kp=1):
    error = right_distance - left_distance
    correction = kp * error

    return correction

# Crear un objeto de captura de video para capturar el flujo de video de la
cámara
cap = cv2.VideoCapture(0)

```

```

# Inicializar motores y sensores ultrasónicos
init_motors()
setup_ultrasonic(sensor_izq_trig, sensor_izq_echo)
setup_ultrasonic(sensor_derecho_trig, sensor_derecho_echo)
setup_ultrasonic(sensor_sup_trig, sensor_sup_echo)

half_speed = 50 # Puedes ajustar este valor según sea necesario

while True:
    set_motor_speed(100, motorderecho_pwm)
    set_motor_speed(100, motorizq_pwm)
    # Capturar un frame del video
    ret, frame = cap.read()

    # Procesar el frame para detectar el color
    resultado, mascara = process_frame(frame)

    # Medir la distancia con los sensores ultrasónicos
    dist_izq = measure_distance(sensor_izq_trig, sensor_izq_echo)
    dist_der = measure_distance(sensor_derecho_trig, sensor_derecho_echo)

    # Si detecta el color deseado, reducir la velocidad a la mitad
    if np.any(mascara):
        set_motor_speed(half_speed, motorderecho_pwm)
        set_motor_speed(half_speed, motorizq_pwm)
    else:
        # Si no detecta el color, seguir moviéndose normalmente
        set_motor_speed(100, motorderecho_pwm)
        set_motor_speed(100, motorizq_pwm)

    # Medir distancias utilizando sensores ultrasónicos
    distancia_izquierda = measure_distance(sensor_izq_trig, sensor_izq_echo)
    distancia_derecha = measure_distance(sensor_derecho_trig,
sensor_derecho_echo)
    distancia_superior = measure_distance(sensor_sup_trig, sensor_sup_echo)

    # Calcular la corrección para ajustar la posición del robot
    correccion = adjust_position(distancia_izquierda, distancia_derecha,
target_distance=30) # Ajustar target_distance según sea necesario

    # Calcular las velocidades de los motores basadas en la corrección
    velocidad_motor_izquierdo = 100 - correccion
    velocidad_motor_derecho = 100 + correccion

    # Ajustar las velocidades de los motores basadas en la diferencia de distancias

```

```

if distancia_derecha > distancia_izquierda:
    set_motor_speed(velocidad_motor_izquierdo, motorderecho_pwm)
    set_motor_speed(60, motorizq_pwm) # ¡Mantener las ruedas izquierdas a
velocidad mayor de la mitad

else:
    set_motor_speed(60, motorderecho_pwm) # ¡Mantener las ruedas derechas
a una velocidad mayor de la mitad
    set_motor_speed(velocidad_motor_derecho, motorizq_pwm)

# Establecer la dirección del motor (ajustar según sea necesario para tu
configuración de motores)
set_motor_direction(gpio.HIGH, gpio.HIGH)

# Imprimir valores de distancia de los sensores ultrasónicos y corrección
print(f"Distancia Sensor Izquierdo: {distancia_izquierda} cm")
print(f"Distancia Sensor Derecho: {distancia_derecha} cm")
print(f"Distancia Sensor Superior: {distancia_superior} cm")
print(f"Corrección: {correccion}")

# Verificar la condición de detección de la pelota
if 15 < distancia_superior < 21.30:
    print("¡Pelota Detectada!")

    # Mantener la actividad actual de los motores durante 2 segundos
    time.sleep(2) #luego de esos dos segundo se apagan los motores para que se
pare el robot y pueda realizar la recoleccion
    set_motor_speed(0, motorderecho_pwm)
    set_motor_speed(0, motorizq_pwm)
    # Activar el nuevo motor durante 3 segundos
    set_motor_speed(100, motor_paletas) # Ajustar la velocidad según sea
necesario.
    time.sleep(3)
    set_motor_speed(0, motor_paletas) # Detener el nuevo motor.

# Capturar el fotograma de la cámara
ret, frame = cap.read()
resultado, mascara = process_frame(frame)

# Mostrar el fotograma original y el resultado lado a lado
cv2.imshow('Fotograma Original', frame)
cv2.imshow('Fotograma de Resultado', resultado)

# Verificar si el usuario ha presionado la tecla 'q'. Si es así, salir del bucle.
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

```



```
# Liberar el objeto de captura de video, limpiar GPIO y cerrar todas las ventanas
cap.release()
gpio.cleanup()
cv2.destroyAllWindows()
```

## **Código de cámara panorámica**

El presente código es para una cámara panorámica la base de datos que se uso fue de la página roboflow y fue el siguiente: <https://universe.roboflow.com/ramiro-chuva-1geds/tennis-ball-8ixxh>. Esta base de datos se utilizó para ser entrenada en Google Colab y obtener el archivo best.py. Este código funciona de manera que detecte las pelotas de tenis de una imagen dada con ayuda de las clases de la base de datos. Luego se dibujaron las líneas con las mediciones reales de una cancha de tenis para de esta manera usando trigonometría pueda obtener la ubicación desde un punto establecido hacia cada objeto detectado.

```

import cv2.
import torch
import numpy as np
from PIL import Image
import math

def calcular_distancia(pt1, pt2, píxeles_por_metro):
    distancia_píxeles = np.sqrt((pt2[0] - pt1[0]) ** 2 + (pt2[1] - pt1[1]) ** 2)
    distancia_metros = distancia_píxeles / píxeles_por_metro
    return distancia_metros

# Cargar el modelo preentrenado para la detección de objetos
modelo = torch.hub.load('ultralytics/yolov5', 'custom',
path='C:\\Users\\diegu\\Downloads\\best (2).pt', force_reload=True)

# Cargar la imagen que se desea detectar
ruta_imagen = 'C:\\Users\\diegu\\Downloads\\tenisccomoc.jpg'
imagen = cv2.imread(ruta_imagen)
imagen_rgb = cv2.cvtColor(imagen, cv2.COLOR_BGR2RGB) # Convertir
el color de la imagen a RGB para PIL
imagen_pil = Image.fromarray(imagen_rgb) # Convertir la imagen a objeto
PIL Image

# Realizar inferencia en la imagen
resultados = modelo(imagen_pil)

# Definir la clase del objeto a detectar
clase_objetivo = 'pelota de tenis'

# Agregar un punto de inicio a la imagen
punto_inicio = (50, 250) # Reemplazar con las coordenadas deseadas del
punto de inicio
cv2.circle(imagen, punto_inicio, 5, (255, 0, 0), -1) # Dibujar un círculo azul
en el punto de inicio.
cv2.putText(imagen, 'roboTennis', (punto_inicio[0] + 10, punto_inicio[1] -
10),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 0, 0), 2) # Agregar
texto 'Inicio' cerca del punto de inicio.

# Crear una copia de la imagen original para dibujar el cuadro delimitador,
la etiqueta y la máscara
imagen_salida = imagen.copy()

# Inicializar una máscara vacía
mascara = np.zeros_like(imagen)

```

```

# Inicializar la posición del robot
posicion_robot = np.array(punto_inicio)

# Dibujar cuadros delimitadores y etiquetas para objetos detectados en la
imagen
for resultado in resultados.xyxy[0]:
    x1, y1, x2, y2, conf, cls = resultado

    # Obtener la clase predicha
    clase_predicha = 'pelota de tenis'

    # Verificar si el objeto detectado es de la clase especificada
    if clase_predicha == clase_objetivo:
        # Calcular las coordenadas del centro del objeto detectado
        centro_objeto = np.array([(x1 + x2) / 2, (y1 + y2) / 2])

        # Calcular la distancia entre el punto de inicio y el objeto
        distancia = np.sqrt(np.sum((centro_objeto - posicion_robot) ** 2))

        # Calcular el ángulo entre el punto de inicio y el objeto
        ángulo = math.atan2(centro_objeto[1] - posicion_robot[1],
centro_objeto[0] - posicion_robot[0])

        # Dibujar una línea desde el punto de inicio al objeto
        cv2.line(imagen_salida, punto_inicio, (int(centro_objeto[0]),
int(centro_objeto[1])), (255, 0, 0), 2)

        # Dibujar el cuadro delimitador y la etiqueta en la imagen de salida
        cv2.rectangle(imagen_salida, (int(x1), int(y1)), (int(x2), int(y2)), (0,
255, 0), 2)
        etiqueta = f'{clase_predicha}: {conf:.2f} | Distancia: {distancia:.2f}'
        cv2.putText(imagen_salida, etiqueta, (int(x1), int(y1)),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)

        # Agregar coordenadas X e Y a la imagen
        cv2.putText(imagen_salida, f'X: {centro_objeto[0]:.2f}',
(int(centro_objeto[0]), int(centro_objeto[1]) - 20),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
        cv2.putText(imagen_salida, f'Y: {centro_objeto[1]:.2f}',
(int(centro_objeto[0]), int(centro_objeto[1]) - 5),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)

        # Agregar información de distancia a la imagen
        cv2.putText(imagen_salida, f'Distancia: {distancia:.2f}', (int(x1),
int(y2) + 20), cv2.FONT_HERSHEY_SIMPLEX, 0.5,
(0, 255, 255), 2)

```

```

# Crear una máscara binaria para el objeto detectado
máscara_roi = np.zeros_like(imagen)
máscara_roi[int(y1):int(y2), int(x1):int(x2)] = 1
mascara += máscara_roi

# Implementar una planificación de ruta básica - Mover el robot hacia
la pelota de tenis
velocidad_robot = 5 # Ajustar la velocidad según sea necesario
posicion_robot[0] += int(velocidad_robot * math.cos(ángulo))
posicion_robot[1] += int(velocidad_robot * math.sin(ángulo))

# Líneas y distancias del segundo código
líneas_y_distancias = [
    ((162, 150), (248, 150), 63),
    ((248, 150), (475, 150), 27.6),
    ((475, 150), (580, 150), 63),
    ((170, 105), (115, 410), 8.84),
    ((170, 105), (35, 800), 29.79),
    ((250, 105), (238, 226), 22.2),
    ((475, 105), (480, 226), 22.2),
    ((238, 226), (218, 410), 28.9),
    ((480, 226), (495, 410), 28.9),
    ((235, 290), (360, 290), 30.4),
    ((360, 290), (485, 290), 30.4),
    ((573, 105), (615, 410), 25.91),
]

for información_linea in líneas_y_distancias:
    (pt1, pt2, píxeles_por_metro) = información_linea

    # Dibujar una línea en la imagen
    cv2.line(imagen_salida, pt1, pt2, (0, 0, 255), 2)

    # Calcular la distancia en metros
    distancia_metros = calcular_distancia(pt1, pt2, píxeles_por_metro)

    # Mostrar el texto de distancia
    texto = f'{distancia_metros:.2f} m'
    cv2.putText(imagen_salida, texto, (pt1[0], pt1[1] - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1)

# Mostrar la imagen de salida con cuadros delimitadores, etiquetas y región
detectada
cv2.imshow('Detección de objetos', imagen_salida)

```

```
cv2.imshow('Región detectada', mascara * 255) # Escalar los valores de la
máscara para visualizar.
cv2.waitKey(0)
cv2.destroyAllWindows()
```

## **Apéndice C**

Cálculos

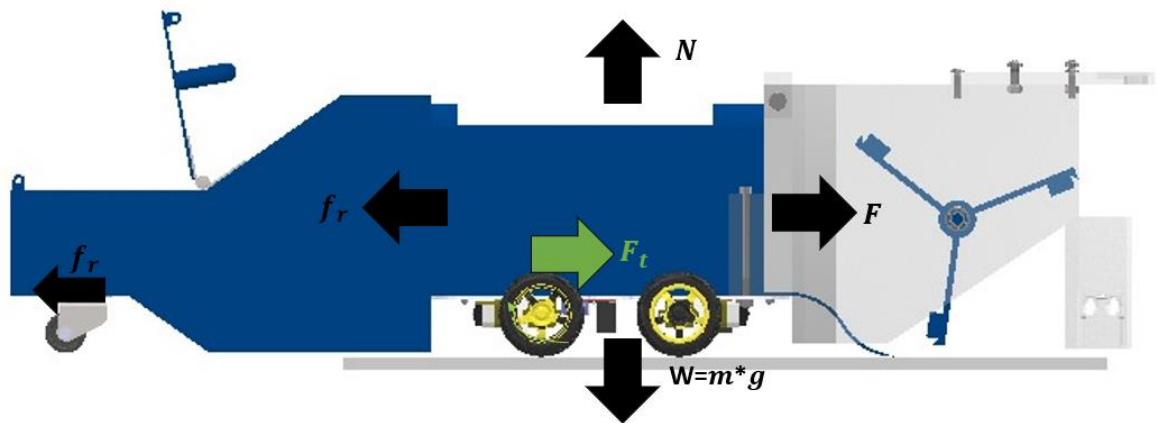
## Análisis del comportamiento de todos los componentes del robot móvil

Para conocer el torque necesario para los motores del robot se determinó el peso de cada uno de los componentes, tomando en cuenta el material del cual están compuestos y el espesor de estos, teniendo así los datos de la **tabla 2.4**.

Además, se realizó un diagrama de cuerpo libre del robot para conocer las fuerzas que actúan sobre él.

**Figura C.1**

*D.C.L de diseño de robot móvil*



Donde:

$f_r$ : es la fuerza de fricción entre las llantas y el suelo.

$g$ : es la fuerza de la gravedad.

$N$ : es la fuerza normal en la superficie.

$F$ : Fuerza ficticia

$F_t$ : es la fuerza de tracción de las llantas.

Procediendo con el cálculo de estas se obtuvo:

$$f_r = \mu_r * N \quad (C.1)$$



Donde:

$\mu_r$  es la constante de fricción del suelo de la cancha de tenis

$$\sum F_y = 0$$
$$N = m * g \quad (C.2)$$

Donde:

$m$ : es la masa del robot

$g$ : es la aceleración gravitacional

Reemplazando la ecuación 2.2 en la ecuación 2.1 y sustituyendo los valores numéricos de las variables se tiene, sabiendo que el coeficiente de fricción en una cancha de tenis de cemento es en promedio 0.63 [23].

$$f_r = 0.63 * 5.47 \text{ Kg} * 9.8 \frac{m}{s^2}$$
$$f_r = 33.77 [N]$$

Además, realizando la fuerza ficticia la cual solamente depende de la masa del robot cuando es sometida a su aceleración:

$$F = m * a \quad (C.3)$$

Donde:

$a$ : es la aceleración máxima esperada del robot móvil  $2 \left[ \frac{m}{s^2} \right]$

Sustituyendo los valores de las variables de obtiene:

$$F = m * a$$
$$F = 5.47 \text{ Kg} * 2 \frac{m}{s^2}$$
$$F = 10.94 [N]$$

De este modo se calculó la fuerza de tracción necesaria para que el robot se mueva a través de la expresión:

$$\sum F_x = 0$$
$$F_t = F_i + f_r \quad (C.4)$$

Sustituyendo los valores en la expresión se tiene:

$$F_t > 10.94 [N] + 33.77 [N]$$
$$F_t > 44.71 [N]$$

Sabiendo que el radio de las ruedas es de 0.065 m y la fuerza debe ser mayor a 44.71 N:

$$F_t = \frac{T}{r} = 44.71 [N]$$
$$T = (F_t * r) / 4 = 0.726 [Nm]$$

La velocidad angular es determinada usando la formula a continuación, poniendo como límite que la máxima velocidad lineal del robot será 0.9[m/s].

$$w = v/r = 13.84 [rad/s] \quad (C.5)$$

De esta relación, se deduce el valor de la velocidad del eje,  $n = 132$ [rpm]. Teniendo el torque y la velocidad del eje, se puede encontrar la potencia que debería tener cada uno de los motores para mover el robot móvil:

$$P = T * w = 10.04 [W] \quad (C.6)$$

De este modo se consideraron motores que cuenten con una potencia mayor a la encontrada para garantizar el óptimo funcionamiento del robot móvil.

Además, para la parte del diseño del eje de las paletas se tuvo que realizar el cálculo del diámetro mínimo que debería tener dicho eje para poder soportar la carga que se le va a aplicar, teniendo en cuenta el torque de motor de 0.14 Nm y las propiedades técnicas del material PLA (Apéndice C), las cuales indican la fuerza máxima de resistencia a la tracción, siendo esta 70 Mpa.

Usar adecuadamente la ecuación de torsión, que describe el comportamiento de los ejes sometidos a un par de torsión externo. La ecuación se aplica a ejes sólidos o tubulares, y se expresa de la siguiente manera:

$$\tau = \frac{T\rho}{J} \quad (C.7)$$

donde:

$\tau$ : es el esfuerzo cortante en el eje (en Pa)

$T$ : es el par de torsión (en Nm)

$J$ : es el momento de inercia de la sección transversal del eje (en m<sup>4</sup>)

El sistema de inercia para ejes sólidos está definido de la siguiente manera:

$$J = \frac{\pi}{4} c^4 \quad (C.8)$$

Siendo  $c$  el valor del radio que es lo que se desea encontrar se procede a sustituir la ecuación 2.5 en la ecuación 2.4 y se despeja el nuestra incógnita  $c$ , obteniendo:

$$c = \sqrt[3]{\frac{2T}{\pi * \tau}}$$

Con esto se procede a sustituir los valores reales:

$$c = \sqrt[3]{\frac{2(0.14 \text{ Nm})}{\pi * 70 \text{ Mpa}}}$$

$$c = 0.001m$$

$$d = 2 * c = 0.002m = 2mm$$

Obteniendo el valor necesario para modelar el eje de diámetro mínimo del eje de las paletas de recolección del diseño, que es de 2 mm, y aplicando un factor de seguridad de aproximadamente

4.5, el diámetro resultante es de 9 mm. Esto garantiza un margen de seguridad suficiente para evitar posibles fallos en su funcionamiento.

**Figura C.2**

*Vista lateral solamente del eje*



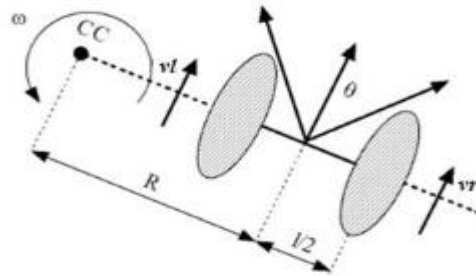
## **Apéndice D**

## Configuración del robot móvil

Se implemento para el movimiento del robot móvil, una cinemática diferencial, la cual se puede observar a continuación.

**Figura D.1**

*Cinemática de un robot diferencial*



Donde:

$v_r$ : velocidad de la rueda derecha

$v_l$ : velocidad de la rueda izquierda

$R$ : trayectoria circular que sigue el centro del robot

$\omega$ : velocidad angular del robot

$l$ : distancia entre las ruedas

$\theta$ : orientación angular del robot

A través de la geometría antes mostrada, se conoce que la velocidad de cada una de las ruedas se puede representar a través de las siguientes ecuaciones:

$$v_r = \omega * \left(R + \frac{l}{2}\right)$$

$$v_l = \omega * (R - \frac{l}{2})$$

Del mismo modo, resolviendo del sistema de ecuaciones se obtiene que:

$$R = \frac{[l * (v_r + v_l)]}{[2 * (v_r - v_l)]}$$

$$\omega = \frac{(v_r - v_l)}{l}$$

Luego, al aplicar las ecuaciones de cinemática directa se puede conocer la posición exacta del robot:

$$x(t) = \frac{1}{2} * \int_{0-t}^t [v_r(t) + v_l(t)] * \cos[\theta(t)] * dt$$

$$y(t) = \frac{1}{2} * \int_{0-t}^t [v_r(t) + v_l(t)] * \sin[\theta(t)] * dt$$

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} \cos(w\delta t) & -\sin(w\delta t) & 0 \\ \sin(w\delta t) & \cos(w\delta t) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x - x_{cc} \\ y - y_{cc} \\ \theta \end{pmatrix} + \begin{pmatrix} x_{cc} \\ y_{cc} \\ \theta\delta t \end{pmatrix}$$

Las ecuaciones previas, indican el movimiento de rotación de un robot a una distancia R entre el centro del eje de las dos ruedas y el centro de curvatura CC con velocidad angular  $\omega$ . Mediante la integración de la ecuación anterior, se puede determinar la posición del robot en un tiempo t, a partir de los parámetros de control  $v_1(t)$  y  $v_2(t)$ .

$$\theta(t) = \frac{1}{2} * \int_0^t [v_r(t) + v_l(t)] * dt$$

Para determinar los valores de  $v_r$  y  $v_l$ , hay que utilizar las ecuaciones cinemáticas inversas del robot. Dado que resolver el sistema de ecuaciones anterior es difícil, se propone si  $v_l$  es igual a  $v_r = v$ , la ecuación anterior se convierte en:

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x + v \cos(\theta)\delta t \\ y + v \sin(\theta)\delta t \\ \theta \end{pmatrix}$$