



**UNIVERSIDAD POLITÉCNICA SALESIANA**  
**SEDE GUAYAQUIL**  
**CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN**

**DISEÑO E IMPLEMENTACIÓN DE UN ROBOT SOCCER AUTÓNOMO MEDIANTE  
VISIÓN ARTIFICIAL A TRAVÉS DEL SISTEMA EMBEBIDO RASPBERRY PI ZERO**

**Trabajo de titulación previo a la obtención del  
Título de Ingeniero en Electrónica**

**AUTORES:** AMY MILENA PARDO CORONEL  
JOSE EMMANUEL VERA CASTRO

**TUTOR:** ING. VICTOR DAVID LARCO TORRES, MGTR

Guayaquil – Ecuador  
2023

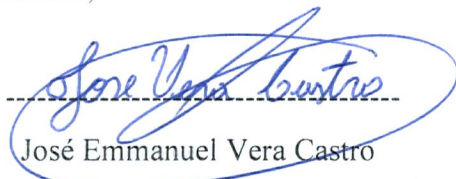
**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE  
TITULACIÓN A LA UNIVERSIDAD POLITECNICA SALESIANA**

Nosotros, José Emmanuel Vera Castro con documento de identificación N° 093018877-6 y Amy Milena Pardo Coronel con documento de identificación N° 070663259-3; expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación: “Diseño e implementación de un robot soccer autónomo mediante visión artificial a través del sistema embebido Raspberry Pi Zero”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Electrónica, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 25 de septiembre del 2023

Atentamente,

  
-----  
José Emmanuel Vera Castro  
093018877-6

  
-----  
Amy Milena Pardo coronel  
070663259-3


**CERTIFICADO DE RESPONSABILIDAD AUTORÍA DEL TRABAJO DE  
TITULACIÓN**

Nosotros, José Emmanuel Vera Castro con documento de identificación N° 093018877-6 y Amy Milena Pardo Coronel con documento de identificación N° 070663259-3, manifiesto que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 25 de septiembre del 2023

Atentamente,



José Emmanuel Vera Castro

093018877-6



Amy Milena Pardo Coronel

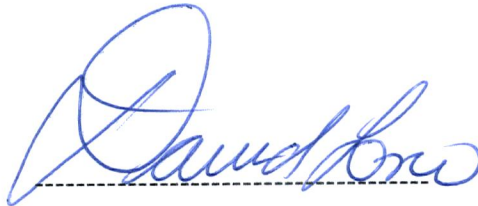
070663259-3

## CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Víctor David Larco Torres con documento de identificación N° 092327013-6, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO E IMPLEMENTACIÓN DE UN ROBOT SOCCER AUTÓNOMO MEDIANTE VISIÓN ARTIFICIAL A TRAVÉS DEL SISTEMA EMBEBIDO RASPBERRY PI ZERO, realizado por Vera Castro José Emmanuel con documento de identificación N° 093018877-6 y Pardo Coronel Amy Milena con documento de identificación N° 070663259-3, obteniendo como resultado final el trabajo de titulación bajo la opción de proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 25 de septiembre del 2023

Atentamente,



Ing. Víctor Larco Torres, Mgtr.

092327013-6

## DEDICATORIA I

Dedico esta tesis a mi Dios Padre celestial, quien ha sido mi guía y fortaleza a lo largo de este camino académico. También quiero dedicar este logro a mi amada familia, cuyo apoyo incondicional y amor han sido mi motor para alcanzar mis metas. En especial, agradezco a mi madre por haber confiado en mí desde el primer día que empecé a estudiar. Gracias por ese voto de confianza que me ha impulsado a seguir adelante, incluso en los momentos más difíciles. A lo largo de esta travesía, he enfrentado diversos desafíos, pero con la gracia de Dios, he podido superarlos y alcanzar este importante logro. Esta tesis es un testimonio de nuestro esfuerzo y dedicación, y con humildad, la dedico a todos aquellos que han sido parte de mi crecimiento personal y académico.

¡Gracias por estar siempre a mi lado y ser parte de este sueño hecho realidad!

*José Vera C.*

## DEDICATORIA I

Con profundo agradecimiento y emoción, presento este trabajo de titulación, el cual representa el esfuerzo de años de estudio y dedicación. Quiero expresar mi sincera gratitud a mis profesores y profesoras, cuyos conocimientos y enseñanzas han sido pilares en mi formación académica. A mi familia, les agradezco por su amor incondicional, paciencia y aliento constante durante todo este recorrido. A mis amigos, gracias por su apoyo incondicional y compañía en los momentos más desafiantes. Con humildad, dedico este trabajo a todos aquellos que me han inspirado y brindado su respaldo en este camino académico. Que este logro sea un reflejo de la perseverancia y el compromiso con el conocimiento y el crecimiento personal.

*Amy Milena Pardo Coronel*

## RESUMEN

En este proyecto de titulación se aborda el diseño y la implementación de un robot soccer autónomo que utiliza visión artificial como principal método de percepción para jugar fútbol de manera autónoma. El sistema se basa en el uso del sistema embebido Raspberry Pi Zero, que proporciona un entorno de desarrollo flexible y de bajo costo.

El objetivo principal de esta investigación es desarrollar un robot capaz de percibir su entorno, reconocer objetos relevantes como la pelota, y tomar decisiones estratégicas para moverse y patear la pelota con el fin de marcar goles. Para lograr esto, se utilizan técnicas de visión artificial para procesar las imágenes capturadas por una cámara y analizar el campo de juego en tiempo real.

La metodología utilizada incluye la configuración y programación de Raspberry Pi Zero con el sistema operativo Raspbian, la instalación y configuración del software de visión artificial y la integración de los componentes de hardware. Se realizan pruebas exhaustivas para evaluar la precisión y el rendimiento del sistema en un entorno de prueba simulado y en situaciones reales de juego.

Esta investigación contribuye al campo de la robótica autónoma y la visión artificial, demostrando la viabilidad de utilizar Raspberry Pi Zero como plataforma para desarrollar sistemas de robots soccer autónomos. Además, proporciona una base sólida para futuras investigaciones y mejoras en el diseño y la implementación de este tipo de sistemas.

Palabras clave: Robot soccer, Visión artificial, Raspberry Pi Zero, Robótica autónoma, Python.

## **Abstract**

This degree project deals with the design and implementation of an autonomous soccer robot that uses artificial vision as the main method of perception to play soccer autonomously. The system is based on the use of the Raspberry Pi Zero embedded system, which provides a flexible and low-cost development environment.

The main objective of this research is to develop a robot capable of perceiving its environment, recognizing relevant objects such as the ball, and making strategic decisions to move and kick the ball in order to score goals. To achieve this, artificial vision techniques are used to process images captured by a camera and analyze the playing field in real time.

The methodology used includes the configuration and programming of Raspberry Pi Zero with the Raspbian operating system, the installation and configuration of the machine vision software and the integration of the hardware components. Extensive testing is performed to evaluate the accuracy and performance of the system in a simulated test environment and in real game situations.

This research contributes to the field of autonomous robotics and machine vision, demonstrating the feasibility of using Raspberry Pi Zero as a platform for developing autonomous soccer robot systems. It also provides a solid foundation for future research and improvements in the design and implementation of such systems.

**Keywords:** Robot soccer, Artificial vision, Raspberry Pi Zero, Autonomous robotics, Python.

## ÍNDICE

1 INTRODUCCIÓN.....	1
2 PROBLEMA DE ESTUDIO.....	3
3 JUSTIFICACIÓN.....	3
4 OBJETIVOS: .....	4
4.1 OBJETIVO GENERAL .....	4
4.2 OBJETIVOS ESPECÍFICOS .....	4
5 MARCO TEÓRICO .....	4
5.1 PYTHON.....	4
5.2 ROBÓTICA AUTÓNOMA .....	5
5.3 SISTEMAS EMBEBIDOS.....	7
5.3.1 RASPBERRY PI ZERO .....	7
5.4 ALGORITMO DE PROCESAMIENTO DE IMÁGENES .....	9
5.5 OPENCV EN RASPBERRY PI .....	10
5.6 TRACCIÓN OMNIDIRECCIONAL.....	11
5.7 TORNEO DE ROBÓTICA: CATEGORÍA: SOCCER AUTÓNOMO ROBOCUP (2022).....	12
5.8 MODELO DE CÁMARA RASPBERRY PI .....	15
5.9 CONTROLADOR DE MOTOR .....	16
5.10 MOTOR DC SIN ESCOBILLAS .....	17
5.11 MÓDULO REGULADOR DE VOLTAJE 5V .....	18
5.12 ARDUINO NANO .....	20
6 METOLOGÍA .....	21
6.1 AUTOCAD AUTODESK.....	21
6.2 RASPBERRY PI ZERO.....	22
6.2.3 THONNY .....	22
6.3 ARDUINO NANO .....	23
7 DESARROLLO DEL PROYECTO.....	24
7.1 DISEÑO DEL ROBOT .....	24
7.1.1 SOPORTE PARA LA CAMARA DEL ROBOT .....	24
7.1.2 TAPA CON PIE DE AMIGO PARA LA CAMARA DEL ROBOT .....	25
7.1.3 PARTE SUPERIOR O CUERPO DEL ROBOT .....	25
7.1.4 PARTE INFERIOR O BASE DEL ROBOT .....	26



7.2 DIAGRAMA DE CONEXIONES PROPUESTO.....	27
7.3 PROGRAMACIÓN EN RASPBERRY PI Y ARDUINO NANO .....	28
7.3.1 CONTROL MEDIANTE ARDUINO NANO .....	28
7.3.2 CONTROL MEDIANTE RASPBERRY PI ZERO .....	31
7 RESULTADOS.....	33
8 CRONOGRAMA .....	38
9. PRESUPUESTO.....	39
10. CONCLUSIONES .....	40
11. RECOMENDACIONES.....	41
12. BIBLIOGRAFÍA .....	1
13 ANEXOS .....	3

### ÍNDICE DE FIGURA

Figura 1 .....	5
Figura 2 .....	8
Figura 3 .....	9
Figura 4 .....	11
Figura 5 .....	12
Figura 6 .....	14
Figura 7 .....	16
Figura 8 .....	17
Figura 9 .....	18
Figura 10 .....	19
Figura 11 .....	20
Figura 12 .....	24
Figura 13 .....	25
Figura 14 .....	26
Figura 15 .....	26
Figura 16 .....	28
Figura 17 .....	30
Figura 18 .....	32
Figura 19 .....	34
Figura 20 .....	35

Figura 21 ..... 36  
Figura 22 ..... 37

**ÍNDICE DE TABLA**

Tabla 1. .... 38  
Tabla 2. .... 39

# 1 INTRODUCCIÓN

El avance de la Robótica y la Inteligencia Artificial ha abierto nuevas posibilidades en diversos campos, incluido el deporte. En particular, el fútbol de robots ha ganado popularidad como un desafío técnico y científico emocionante que combina habilidades de percepción, toma de decisiones y control físico. El objetivo de este proyecto de titulación es diseñar e implementar un robot soccer autónomo mediante visión artificial utilizando el sistema embebido Raspberry Pi Zero.

En la Universidad Politécnica Salesiana, se han presentado diversos proyectos relacionados con robots soccer autónomos, cada uno utilizando diferentes dispositivos y lenguajes de programación. En este proyecto, se ha optado por emplear un Raspberry Pi Zero, que se integrará con una programación en Python de código abierto.

El fútbol de robots autónomos implica el desarrollo de sistemas robóticos capaces de jugar al fútbol sin intervención humana. Esta tarea desafiante requiere la integración de diversas tecnologías, como la visión artificial, el procesamiento de imágenes en tiempo real, la toma de decisiones estratégicas y el control de movimiento. El uso de sistemas embebidos, como Raspberry Pi Zero, ofrece una plataforma flexible y de bajo costo para abordar estos desafíos.

La visión artificial es un componente clave en la percepción del entorno por parte del robot. Permite que el robot capture imágenes del campo de juego, reconozca objetos relevantes, como la pelota y otros robots, y tome decisiones basadas en esta información visual. La implementación de algoritmos de visión artificial efectivos y eficientes es fundamental para el éxito de un robot soccer autónomo.

El Raspberry Pi Zero es una placa de computadora de bajo costo y tamaño reducido que proporciona capacidades de procesamiento y comunicación. Su versatilidad y facilidad de uso lo convierten en una opción ideal para implementar sistemas robóticos. Además, su capacidad de ejecutar el sistema operativo Raspbian, basado en Linux, brinda una amplia gama de herramientas y bibliotecas de software para el desarrollo de aplicaciones de visión artificial.

El propósito central de este proyecto se enfoca en trabajos de investigación para la Carrera de Electrónica y Automatización de la Universidad Politécnica Salesiana, en Guayaquil. Su objetivo primordial es desarrollar una rutina que permita familiarizarse con las tecnologías y procesos de control más avanzados, aplicados en el ámbito de la visión artificial, empleando un sistema de tracción omnidireccional para el movimiento del robot.

Para lograr esto, se abordarán aspectos como la selección y configuración de los componentes de hardware, el desarrollo de algoritmos de visión artificial y toma de decisiones, así como la integración y prueba del sistema completo.

Los resultados obtenidos en este proyecto contribuirán al avance de la robótica autónoma y la visión artificial, demostrando la viabilidad de utilizar Raspberry Pi Zero como plataforma para el desarrollo de robots de fútbol autónomos. Además, sentará las bases para futuras investigaciones y mejoras en el diseño y la implementación de sistemas similares.

## **2 PROBLEMA DE ESTUDIO**

En la actualidad, en competencias nacionales de robótica, la participación en la categoría Soccer Autónomo ha sido escasa. Adicionalmente, en la Carrera de Electrónica y Automatización se imparte la materia de Robótica Móvil, en la cual, no existen los suficientes prototipos para poder realizar prácticas; por lo que se considera imprescindible diseñar un prototipo que pueda participar en futuros concursos nacionales e internacionales, motivando la implementación de nuevas tecnologías y una mayor participación en la categoría.

Para ello, se busca diseñar un robot que cumpla con las especificaciones para participar en este tipo de categoría y al mismo tiempo incentivar al aprendizaje de los estudiantes que cursan la materia de Robótica para su respectivo estudio y proyección a futuras prácticas con el prototipo.

## **3 JUSTIFICACIÓN**

El desarrollo de robots autónomos capaces de jugar al fútbol utilizando visión artificial es un campo activo de investigación en la robótica. Este trabajo de titulación contribuirá al avance de esta área al diseñar e implementar un sistema utilizando Raspberry Pi Zero como plataforma principal. Los resultados y hallazgos obtenidos en esta investigación podrían ser útiles para futuros desarrollos en robótica autónoma y aplicaciones relacionadas. Adicional, la implementación de un sistema de robot soccer autónomo implica la integración de varias tecnologías, por lo tanto, proporcionará una oportunidad para explorar y comprender la integración efectiva de estas tecnologías en un contexto práctico, lo que puede tener aplicaciones más amplias en otros campos de la Robótica y la Inteligencia Artificial, beneficiando a estudiantes y docentes de la Carrera de Electrónica y Automatización.

## 4 OBJETIVOS:

### 4.1 OBJETIVO GENERAL

Diseñar e implementar un robot soccer autónomo mediante visión artificial a través del sistema embebido Raspberry Pi Zero.

### 4.2 OBJETIVOS ESPECÍFICOS

- Dimensionar los componentes apropiados para el control y mecanismo del robot soccer.
- Construir un robot soccer autónomo basado en diseños personalizados.
- Programar el robot para la búsqueda del balón y control del mecanismo de disparo en dirección al arco utilizando Python.
- Realizar pruebas de funcionamiento del robot.

## 5 MARCO TEÓRICO

### 5.1 PYTHON

Python es un lenguaje de programación de alto nivel conocido por su sencillez, legibilidad y versatilidad. Es popular debido a su uso en ciencia de datos y Machine Learning. Además, contiene una amplia librería con la que se puede ejecutar los códigos propuestos para este proyecto.

En Python, la organización de los datos de elevado desarrollo se desempeña de manera más eficaz con un planteamiento simple, al tener unos sintagmas dinámicos de expresión que se desempeña en un lenguaje de programación para scripting y con una fácil adaptación en diversos códigos. (Python Software Foundation, 2023)

## 5.2 ROBÓTICA AUTÓNOMA

La robótica autónoma es un campo de estudio que se centra en el diseño y desarrollo de robots capaces de operar de manera independiente, sin la necesidad de una supervisión o control humano constante. Estos robots están equipados con sistemas sensoriales y de percepción que les permiten interactuar con su entorno, tomar decisiones basadas en la información recibida y ejecutar acciones para cumplir con sus objetivos.

La autonomía en los robots implica la capacidad de percibir y comprender el entorno en el que se encuentran. Esto se logra mediante el uso de diversos sensores, como cámaras, sensores de proximidad, láseres o radares, que capturan información sobre el entorno físico y la transmiten al sistema de procesamiento del robot como se muestra en la figura 1.

(ROBOWORKS, 2023)

### Figura 1

*Robot Móvil Autónomo Mecabot*



*Nota:* En imagen se visualiza el robot móvil MECABOT, el cual contiene los siguientes elementos dentro de su estructura: cámara, sensores y llantas omnidireccionales, este robot está adaptado para comprender en el entorno que se encuentre para hacer sus movimientos

requeridos. Reproducido de Mecabot Plataforma robótica móvil autónoma para interior, por ROBOWORKS, 2023

<https://manuals.plus/es/robworks/mecabot-autonomous-mobile-robot-platform-for-indoor-manual>

Una vez que el robot ha adquirido datos sensoriales, estos son procesados y analizados para extraer información relevante. Esto puede incluir la detección y reconocimiento de objetos, la estimación de la posición y la orientación, la identificación de obstáculos y la comprensión del estado del entorno. Los algoritmos de procesamiento de datos y visión artificial juegan un papel fundamental en esta etapa para interpretar y comprender la información capturada.

Basado en la información percibida, el robot debe tomar decisiones sobre las acciones que debe emprender para lograr su objetivo. Esto implica la planificación y la toma de decisiones en función de los datos sensoriales y del conocimiento previo del robot. Los algoritmos de toma de decisiones pueden ser reglas predefinidas, algoritmos de aprendizaje automático o una combinación de ambos, dependiendo de la complejidad y los requisitos del problema.

Una vez que se ha tomado una decisión, el robot debe ejecutar las acciones necesarias para llevar a cabo la tarea. Esto implica el control de los actuadores, como motores y manipuladores, para realizar movimientos físicos en el entorno. Los algoritmos de control de movimiento permiten al robot coordinar sus actuadores para moverse de manera segura y precisa, evitando obstáculos y cumpliendo con la tarea establecida.



## 5.3 SISTEMAS EMBEBIDOS

Los sistemas embebidos desempeñan un papel fundamental en muchas aplicaciones tecnológicas y permiten la integración de hardware y software en un solo dispositivo para realizar tareas específicas. Estos sistemas se diseñan y desarrollan teniendo en cuenta los requisitos y restricciones particulares de cada aplicación, y ofrecen soluciones eficientes y especializadas para una amplia gama de aplicaciones en diversos sectores.

Adicional, son sistemas informáticos diseñados para realizar tareas específicas, generalmente con restricciones de recursos, tamaño y energía. Raspberry Pi Zero es un ejemplo de sistema embebido ampliamente utilizado que ofrece capacidades de procesamiento y comunicación en un factor de forma compacto y asequible. Su flexibilidad y facilidad de uso lo convierten en una opción atractiva para implementar sistemas robóticos.

### 5.3.1 RASPBERRY PI ZERO

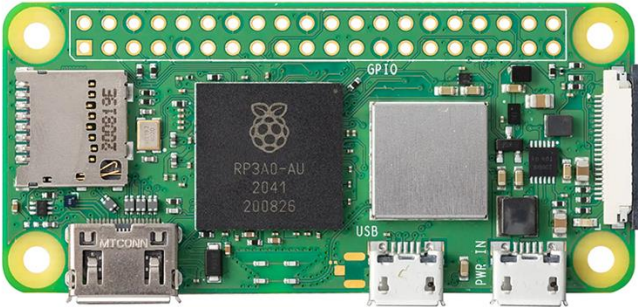
Raspberry Pi Zero es una placa de computadora de bajo costo y tamaño reducido. Está equipado con un procesador ARM, memoria RAM, puertos GPIO y capacidad de comunicación. Su versatilidad y capacidad para ejecutar el sistema operativo Raspbian lo convierten en una plataforma popular para proyectos de robótica y visión artificial. (Bisson, 2021)

En la figura 2, el Raspberry Pi Zero es parecido a la tarjeta madre que tienen los móviles inteligentes. Poseen controladores de una familia y suficiente del mismo periférico que son:

- HDMI
- Audio
- Micrófono
- Conector de tarjeta SD

## Figura 2

### Raspberry Pi Zero



*Nota:* En la imagen se puede visualizar la placa Raspberry Pi Zero con sus respectivos puertos: HDMI, conector de tarjeta SD, audio y micrófono, ya que trabaja como una minicomputadora y tiene una capacidad operativa que puede ayudar con visión artificial. Reproducido de Revisión de Raspberry Pi Zero 2 W: el dispositivo de placa única de bajo costo obtiene una actualización de cuatro núcleos. Bisson, 2021

<https://www.zdnet.com/article/raspberry-pi-zero-2-w-review-performance-in-a-small-form-factor/>

Nuevas versiones más actuales poseen Bluetooth y Wifi, sus interfaces la hacen interesante para controlar plataformas robóticas.

Este sistema operativo Raspbian permitirá explorar a las personas de diferentes edades, la programación en el área de computación de Python y Scratch en su lenguaje de programación.

## 5.4 ALGORITMO DE PROCESAMIENTO DE IMÁGENES

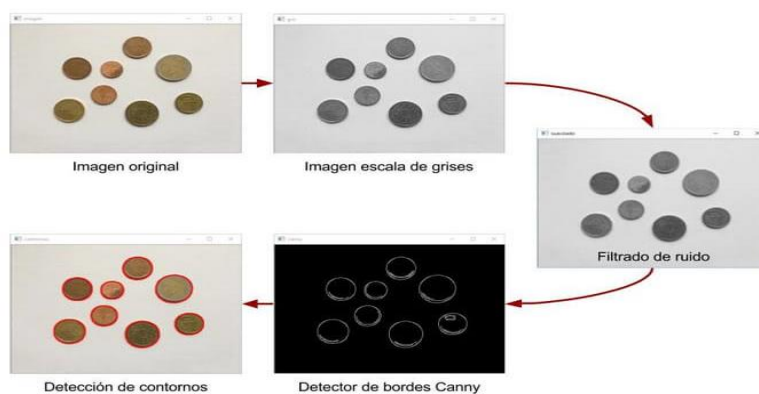
Los algoritmos de procesamiento de imágenes son técnicas utilizadas para analizar y modificar imágenes digitales. Estos algoritmos se aplican en la etapa de preprocesamiento de las imágenes capturadas por la cámara del robot soccer autónomo. Ejemplos de técnicas de procesamiento de imágenes son el filtrado, la segmentación, la detección de bordes y la extracción de características.

El método más eficaz para la compilación de datos es el modelo Raspberry Pi de cámara, es percibido de forma automáticamente una vez unido al puerto j3 del Raspberry Pi.

Para la captura de imágenes como se puede visualizar en la figura 3. Se puede realizar mediante un software utilizando una cámara que ayudará a transferir información mediante paquetes de almacenamientos temporales en la memoria mediante esto se utilizará el Raspberry Pi. (Arduino, Programar Fácil, 2017)

**Figura 3**

*Algoritmo De Procesamiento De Imágenes*



*Nota:* Se explica el procesamiento de la imagen en donde empieza con la gráfica original, continúa con la fase de conversión a escala de grises, filtrado de ruidos, detector de bordes,

detección de contornos y aplicando el algoritmo de Canny se puede detectar los bordes de la imagen inicial. Reproducido de Detector de bordes Canny, cómo contar objetos con OpenCV y Python, por Arduino, Programar Fácil, 2017

<https://programarfácil.com/blog/vision-artificial/detector-de-bordes-canny-opencv/>

## **5.5 OPENCV EN RASPBERRY PI**

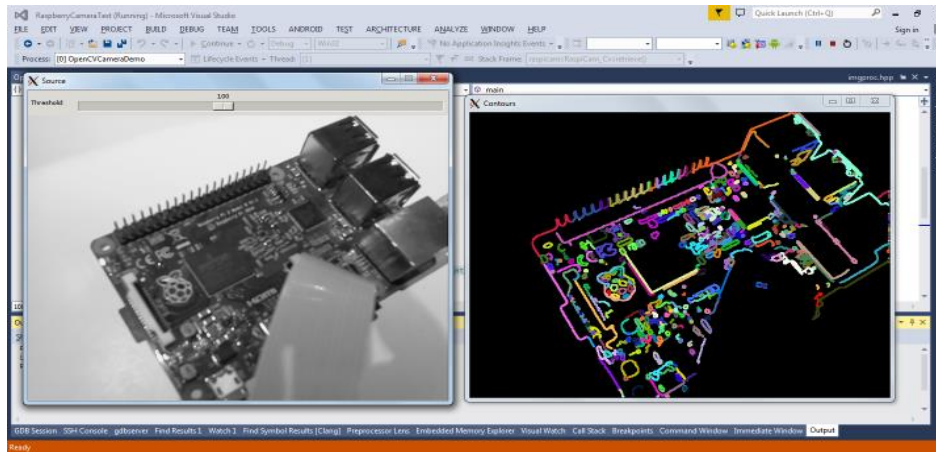
OpenCV permite a los desarrolladores implementar algoritmos y técnicas de visión por computadora en aplicaciones y proyectos.

En el caso de Raspberry Pi, OpenCV se puede utilizar en diversas aplicaciones para realizar tareas relacionadas con la visión por computadora. Raspberry Pi es un ordenador de placa única (SBC) de bajo costo y tamaño compacto, que ofrece capacidades de procesamiento y E/S adecuadas para proyectos de visión por computadora más simples.

Al utilizar OpenCV en Raspberry Pi, puedes realizar una amplia gama de tareas, como reconocimiento de objetos, detección y seguimiento de rostros, detección de colores, análisis de movimiento, calibración de cámaras, procesamiento de imágenes y videos, entre otros. De esta manera, permite a los usuarios manipular y analizar datos visuales capturados o procesados por máquinas como se muestra en la figura 4. (visualgdb, 2015)

## Figura 4

### Uso De La Librería Opencv En Raspberry Pi



*Nota:* En la imagen se divide en dos partes; a la derecha se puede observar la imagen procesada mediante la librería OpenCV donde se detecta los bordes del objeto, muy similar al algoritmo de Canny, y al lado izquierdo se puede visualizar la misma gráfica, pero en escala de grises para identificar el objeto al cual se quiere procesar. Reproducido de Using OpenCV with Raspberry Pi 2 Camera, por visualgdb, 2015

<https://visualgdb.com/tutorials/raspberry/opencv/camera/>

Mediante el uso de OpenCV con una Raspberry Pi, los desarrolladores pueden crear aplicaciones que utilizan cámaras para analizar el entorno en tiempo real. Esto puede resultar especialmente útil para tareas como la supervisión de grabaciones de seguridad, el análisis del flujo de tráfico o incluso la creación de sistemas domóticos inteligentes.

## 5.6 TRACCIÓN OMNIDIRECCIONAL

La tracción omnidireccional es un tipo de mecanismo de tracción que permite a un vehículo o robot moverse en cualquier dirección sin tener que cambiar su orientación. Esto se consigue utilizando varias ruedas u orugas situadas alrededor del perímetro del vehículo o robot,

lo que le permite avanzar, retroceder, desplazarse lateralmente o en diagonal en cualquier ángulo.

En la tracción omnidireccional, cada rueda u oruga suele accionarse de forma independiente, lo que permite controlar con precisión el movimiento del vehículo en cualquier dirección. Este tipo de sistema de tracción se utiliza habitualmente en vehículos autónomos terrestres, como robots de reparto o de limpieza, para desplazarse por entornos y obstáculos complejos como se muestra en la figura 5. (fablab.uchile, 2018)

### **Figura 5**

#### *Robot Con Tracción Omnidireccional*



*Nota:* En la imagen se puede visualizar a la base de un robot incorporado por ruedas omnidireccionales, en la que le permite moverse libremente hacia cualquier dirección.

Reproducido de Robot Omnidireccional, por fablab.uchile, 2018

<http://www.fablab.uchile.cl/proyectos/268/robot-omnidireccional/>

### **5.7 TORNEO DE ROBÓTICA: CATEGORÍA: SOCCER AUTÓNOMO ROBOCUP (2022)**

La RoboCup es una competencia internacional que tiene como objetivo desarrollar y promover la investigación en robótica y sistemas autónomos. Una de las categorías más

populares de la Robo Cup es la Liga de Fútbol Robótico, donde los equipos compiten utilizando robots autónomos para jugar partidos de fútbol.

En la categoría de fútbol autónomo de la Robo Cup, los equipos están formados por robots humanoides o robots móviles que están programados para jugar al fútbol de manera autónoma, sin intervención humana durante el partido. Estos robots están equipados con sensores, actuadores y sistemas de visión por computadora que les permiten percibir el entorno, tomar decisiones y ejecutar acciones en tiempo real.

Los partidos de fútbol autónomo de la RoboCup se llevan a cabo en un campo de juego específico, con reglas similares a las del fútbol convencional. Los equipos compiten para marcar goles y evitar que el equipo contrario anote. Los robots deben ser capaces de moverse de manera autónoma, comunicarse entre sí y trabajar en equipo para lograr sus objetivos.

Para lograr el fútbol autónomo, los equipos deben desarrollar algoritmos y estrategias sofisticadas que permitan a los robots tomar decisiones en tiempo real. Estos algoritmos pueden incluir técnicas de inteligencia artificial, aprendizaje automático, visión por computadora y planificación de movimiento, entre otros.

La competencia en la categoría de fútbol autónomo de la Robo Cup impulsa la investigación en robótica y sistemas autónomos, fomentando el desarrollo de nuevas tecnologías y enfoques para la percepción, el razonamiento y la toma de decisiones en entornos dinámicos y complejos. Además, la Robo Cup promueve la colaboración y el intercambio de conocimientos entre investigadores, académicos y entusiastas de la robótica de todo el mundo. (Universidad técnica Federico Santa María, 2018)

En la figura 6, Robot Soccer Word Cup consiste en la aplicación de:

- Un sistema de fútbol robotizado.
- Está formado por jugadores robots.
- transmisores de RF.
- Cámara de visión artificial.

### Figura 6

*Equipo Participante En El Torneo Robocup Soccer League 2018*



*Nota:* En la imagen se puede visualizar los dos robots soccer que participaron en el Torneo Robocup Soccer League 2018, equipo que represento a la Universidad Técnica Federico Santa María en la categoría Small Size de la competencia. Reproducido de Sansanos clasifican a mundial de fútbol de robots autónomos RoboCup Soccer League 2018, por Universidad tecnica Federico Santa María, 2018

<https://noticias.usm.cl/2018/03/20/sansanos-clasifican-a-mundial-de-futbol-de-robots-autonomos-robocup-soccer-league-2018/>



## 5.8 MODELO DE CÁMARA RASPBERRY PI

El módulo de cámara de Raspberry Pi es un accesorio que se conecta a los puertos CSI (Camera Serial Interface) de una Raspberry Pi. Permite capturar imágenes y videos de alta calidad utilizando la potencia de procesamiento de la placa Raspberry Pi. Estas cámaras tienen un sensor de imagen de alta calidad y una lente fija incorporada.

El módulo de cámara se conecta a la Raspberry Pi a través de un cable flexible que transfiere datos y alimentación. En la placa Raspberry Pi, hay un conector específico para el cable de la cámara, generalmente etiquetado como "CSI". Al conectar el cable correctamente, el módulo de cámara se integra con la Raspberry Pi y está listo para su uso.

Una vez conectado, puedes utilizar el software de la Raspberry Pi para acceder al módulo de cámara y capturar imágenes o grabar videos. Raspberry Pi proporciona una biblioteca llamada "piconera" que simplifica la captura de imágenes y videos desde el módulo de cámara. Esta biblioteca se puede utilizar con varios lenguajes de programación, como Python.

La cámara de Raspberry Pi es bastante versátil y puede utilizarse en una amplia variedad de proyectos. Puedes utilizarla para crear un sistema de videovigilancia, un dispositivo de detección de movimiento, un time-lapse, un escáner 3D o incluso como una cámara de seguridad para un robot. (ecuarobot, 2020)

El módulo de cámara IMX219 (figura 7) tiene un sensor de 8 megapíxeles que puede

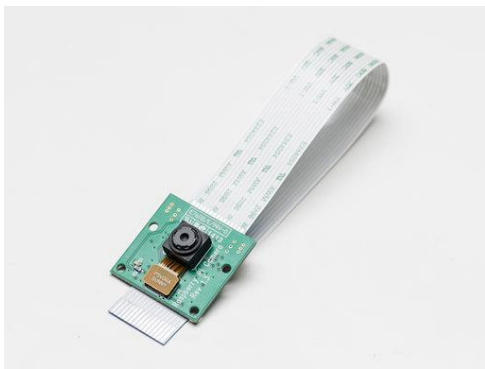
capturar imágenes a una resolución de hasta 3280 x 2464 píxeles, y también puede grabar video a 1080p a 30 cuadros por segundo.

El módulo de cámara se conecta al puerto CSI (Camera Serial Interface) de la Raspberry Pi, que está diseñado específicamente para soportar cámaras de alta velocidad y alta resolución.

Este módulo de cámara es muy útil para una variedad de proyectos de Raspberry Pi que requieren capacidades de visión por computadora o fotografía.

### **Figura 7**

*Módulo De Cámara Arducam IMX219*



*Nota:* El módulo de cámara para Raspberry Pi proporciona una forma conveniente y asequible de agregar capacidades de captura de imágenes y video a tus proyectos de Raspberry Pi. Reproducido de Obtenido de Comenzando con el módulo de cámara Raspberry Pi, por

<https://ecuarobot.com/2020/02/19/comenzando-con-el-modulo-de-camara-raspberry-pi/>

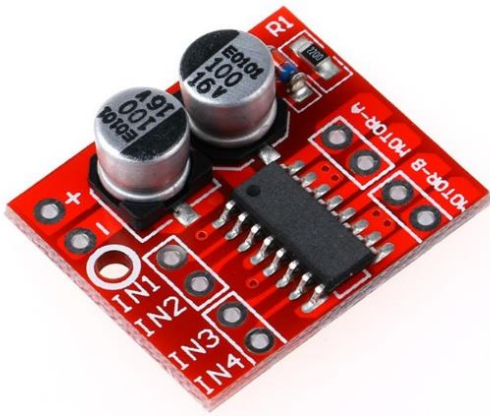
## **5.9 CONTROLADOR DE MOTOR**

Como se observa en la figura 8, el controlador de motores 222aak facilita el uso del controlador de motor de corriente continua, que puede funcionar en modo de doble canal para el control bidireccional independiente de dos motores o el modo de un solo canal para

conducir un motor con mayor corriente. Tiene un amplio rango de voltaje de funcionamiento de 2V-12V y puede entregar un continuo 1.5A, a cada canal de motor. (puntoflotante, 2023)

## Figura 8

### MÓDULO 222aak



*Nota:* El módulo 222aak driver es una opción popular para controlar motores de corriente continua y motores paso a paso en proyectos de robótica y electrónica debido a su capacidad de manejar corrientes y voltajes altos, su facilidad de uso y su asequibilidad. Reproducido de DRIVER-DUAL-PWM-L298N-PUENTE-H-MOTORES-CORRIENTE-DIRECTA, por puntoflotante, 2023

<https://www.puntoflotante.net/DRIVER-PWM-PUENTE-H-MOTORES-CD.htm>

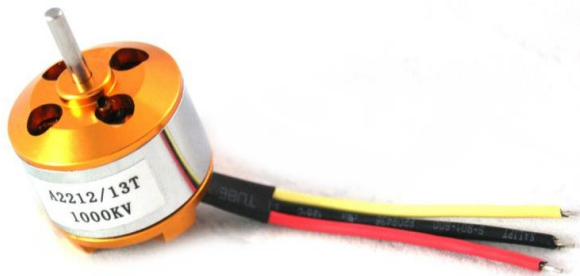
## 5.10 MOTOR DC SIN ESCOBILLAS

Los motores eléctricos sin escobillas, como se muestra en la figura 9, solían poseer un copilador de delgadas o un par de anillos rozantes. Este proceso, que ocasiona rozamiento, minimiza el rendimiento, emana calor y ruido, y esto demanda demasiado

mantenimiento y puede ejercer partículas de carbón que mancha el motor de un polvo, que además puede ser conductor. (tytorobotics, 2021)

## Figura 9

### *Motor DC Sin Escobillas*



*Nota:* El control de un motor DC sin escobillas generalmente se realiza mediante un controlador electrónico que secuencía las bobinas del estator para generar el campo magnético necesario para el movimiento. Reproducido de ¿Qué es un motor sin escobillas?, por tytorobotics, 2021

. <https://www.tytorobotics.com/blogs/articles/what-is-a-brushless-motor>

Este controlador puede utilizar diferentes técnicas, como la conmutación trapezoidal o la conmutación sinusoidal, para lograr un funcionamiento suave y eficiente del motor.

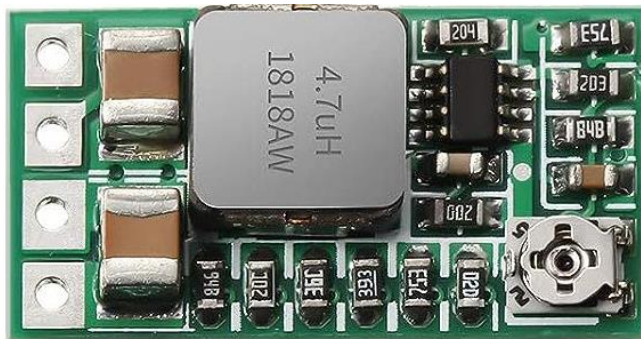
## 5.11 MÓDULO REGULADOR DE VOLTAJE 5V

El funcionamiento de un módulo regulador de voltaje de 5 V (figura 10) se basa en un circuito integrado regulador de voltaje lineal, como el popular LM7805. Este tipo de regulador de voltaje toma una entrada de voltaje más alto, que puede variar dependiendo del diseño, y a través de su circuito interno, ajusta la tensión para mantenerla en un nivel constante de 5 V en la salida. Esto se logra mediante la comparación del voltaje de entrada con una referencia interna. Si el voltaje de entrada es mayor que el voltaje de referencia, el regulador reduce el voltaje para

mantenerlo en 5 V. Si el voltaje de entrada es menor, el regulador no realiza ningún ajuste y simplemente pasa la tensión sin cambios. (Tardigrade Outdoors, 2021)

### Figura 10

*Módulo Regulador De Voltaje 5V*



*Nota:* Cuando se conecta una fuente de alimentación variable al módulo regulador de voltaje 5V, el circuito integrado se encarga de ajustar y mantener el voltaje de salida en 5 voltios, independientemente de las fluctuaciones en el voltaje de entrada. Reproducido de Overhauling a Monoprice MP Select Mini V2 MPSM V2 with SKR 1.4 board, octoprint, and morepor, por Tardigrade Outdoors, 2021

. <https://www.tardigradeoutdoors.com/blog/overhauling-a-monoprice-mp-select-mini-v2-mpsm-v2-with-skr-14-board-octoprint-and-more>

Esto es importante para garantizar un suministro estable de energía a los dispositivos electrónicos, ya que muchos componentes y microcontroladores, como el mencionado Arduino Nano, requieren una tensión de 5V para funcionar correctamente.

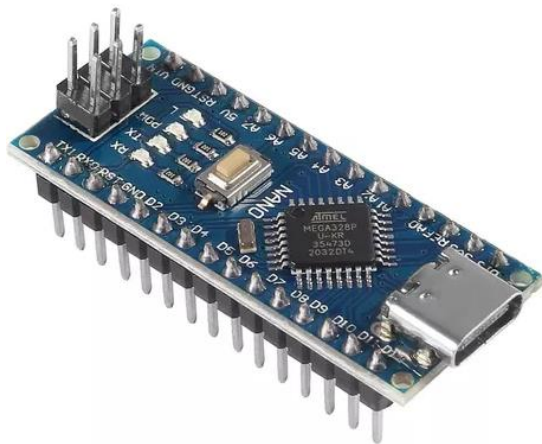
## 5.12 ARDUINO NANO

El Arduino Nano (figura 11) es una placa de desarrollo electrónica basada en el microcontrolador ATmega328P. Es una versión compacta y de bajo costo del popular Arduino Uno. El Nano se destaca por su tamaño reducido y su capacidad de ser utilizado en proyectos donde el espacio es limitado.

El Arduino Nano es compatible con una amplia variedad de sensores, actuadores y módulos, lo que lo hace ideal para proyectos de electrónica y robótica. Puede ser utilizado en aplicaciones como domótica, sistemas de control, monitoreo ambiental, robótica educativa, entre otros. La aplicación del Nano dentro de este proyecto será para mecanismo de control de los motores. (studiopieters, 2022)

**Figura 11**

*ARDUINO NANO*



*Nota:* El Arduino Nano se utiliza en proyectos de electrónica DIY, robótica, domótica, automatización y muchas otras aplicaciones. Reproducido de Arduino® Nano – Pinout, por studiopieters, 2022

<https://www.studiopieters.nl/arduino-nano-pinout/>

## 6 METODOLOGÍA

La metodología de este proyecto pretende resolver la problemática mencionada en los puntos anteriores, a través de varias etapas y varios lineamientos a seguir en la robótica, como primer paso ensamblar la estructura del robot, de acuerdo a sus especificaciones técnicas, como segundo paso se tiene, empezar a colocar los componentes y todas las piezas del robot para realizar todas las conexiones y pruebas a través de las rutinas previamente programadas y los procedimientos a través del lenguaje seleccionado.

### 6.1 AUTOCAD AUTODESK

Para el proyecto de titulación del robot de fútbol autónomo, se utilizó AutoCAD de las siguientes formas:

1. ***Diseño del cuerpo del robot:*** AutoCAD se usó para crear un modelo en 2D o 3D del cuerpo del robot y sus componentes. Se diseñó la estructura, la disposición de los motores, los soportes, las ruedas u otros elementos mecánicos necesarios. Utilizando las herramientas de dibujo y modelado de AutoCAD, se puede obtener una representación visual precisa del diseño.

2. ***Diseño de piezas personalizadas:*** se pudo crear dibujos técnicos y modelos 3D de las piezas, especificando dimensiones, tolerancias y otros detalles necesarios para la fabricación. Esto facilita la comunicación con los fabricantes o talleres de mecanizado e impresión.

3. ***Documentación técnica:*** AutoCAD es útil para crear planos, diagramas y documentación técnica relacionada con el proyecto. Se utilizó para crear esquemas de montaje y diseño de las piezas robot, que formarán parte de la documentación del proyecto.

En conclusión, el software permitió diseñar la estructura del robot y piezas que forman

parte de este. Adicionalmente, se pudo crear los planos de las piezas que conformar al soccer.

## 6.2 RASPBERRY PI ZERO

La Raspberry Pi es un sistema embebido muy popular y versátil que puede desempeñar un papel importante en tu proyecto de robot de fútbol autónomo. A continuación, se detallarán es algunas formas en la que se beneficia utilizar la Raspberry Pi para este proyecto:

1. **Unidad de procesamiento:** La Raspberry Pi es la unidad de procesamiento principal del robot. Se pudo programar para realizar tareas de procesamiento de imágenes utilizando bibliotecas como OpenCV. Esto te permitió implementar algoritmos de visión artificial para reconocer objetos, detectar líneas de campo y realizar seguimiento de la pelota.

2. **Comunicación:** La Raspberry Pi actúa como el centro de comunicación del robot. Se establece una conexión inalámbrica, vía Wi-Fi. Esto permite enviar comandos al robot y recibir datos desde el mismo en tiempo real.

### 6.2.3 THONNY

Thonny es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para Python que se utilizó en el proyecto de la siguiente forma:

1. **Programación del robot:** Thonny permite escribir y ejecutar código Python en la Raspberry Pi, que es la unidad de procesamiento principal de tu robot. Se utilizó Thonny para escribir el código que controla el comportamiento del robot, incluyendo la lógica de movimiento, la comunicación con el Arduino, y la implementación de algoritmos de visión artificial utilizando bibliotecas como OpenCV.



Thonny es una herramienta popular y fácil de usar para programar en Python, y ha sido una opción conveniente para el desarrollo del proyecto de robot de fútbol autónomo en la Raspberry Pi.

### 6.3 ARDUINO NANO

El Arduino Nano desempeña un papel importante en el proyecto del robot soccer autónomo. Aquí se explicará algunas formas en las que se utilizó el Arduino Nano:

1. **Control de motores:** El Arduino Nano es ideal para controlar los motores del robot. Se usó módulos y controladores de motor compatibles con Arduino para controlar la velocidad y dirección de los motores que permiten el movimiento del robot.

2. **Comunicación con la Raspberry Pi:** se establece una comunicación entre la Raspberry Pi y el Arduino Nano para intercambiar datos y comandos. La Raspberry Pi puede enviar instrucciones al Arduino Nano sobre el movimiento del robot.

3. **Control de actuadores:** Además de controlar motores, el Arduino Nano también puede controlar otros actuadores, como el motor brushless. Se utilizó el Arduino Nano para activar y controlar este actuador.

## 7 DESARROLLO DEL PROYECTO

En este apartado se explica sobre los diseños de las piezas personalizadas que forman parte del robot soccer que fueron realizadas en el software de diseño gráfico AutoCAD. Además, se explicará sobre el código para el control del movimiento de los motores por medio de Arduino y procesamiento de imágenes mediante Raspberry Pi Zero.

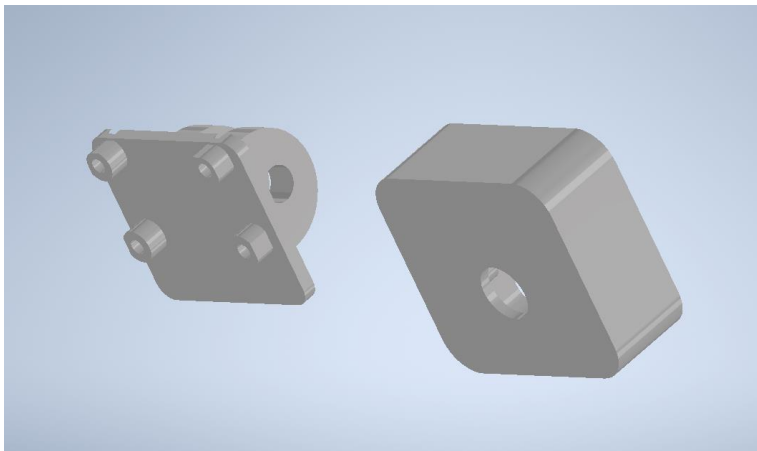
### 7.1 DISEÑO DEL ROBOT

#### 7.1.1 SOPORTE PARA LA CAMARA DEL ROBOT

Para iniciar la planificación del diseño del robot, se diseñó en el programa AutoCAD, la pieza de protección de la cámara que permitirá detectar la pelota y de esa manera enviar los datos a la Raspberry Pi. En la figura 12 se muestra la pieza del soporte de la cámara en 3D.

#### Figura 12

*Soporte del Módulo de la Cámara Arducam IMX219*



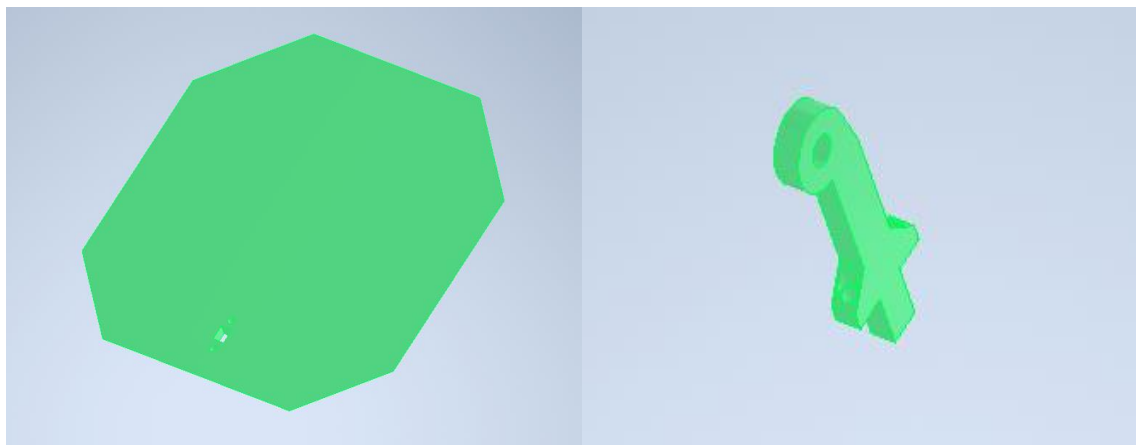
*Nota:* Estructura de la base que funciona como soporte de la cámara que conectará al Raspberry Pi.

### 7.1.2 TAPA CON PIE DE AMIGO PARA LA CAMARA DEL ROBOT

En la figura 13, se visualiza la tapa superior del robot en donde se colocarán el soporte para la base de la cámara (pie de amigo) y un interruptor de activación del robot.

**Figura 13**

*Tapa del Robot Soccer y Pie de Amigo para la Cámara del Robot*



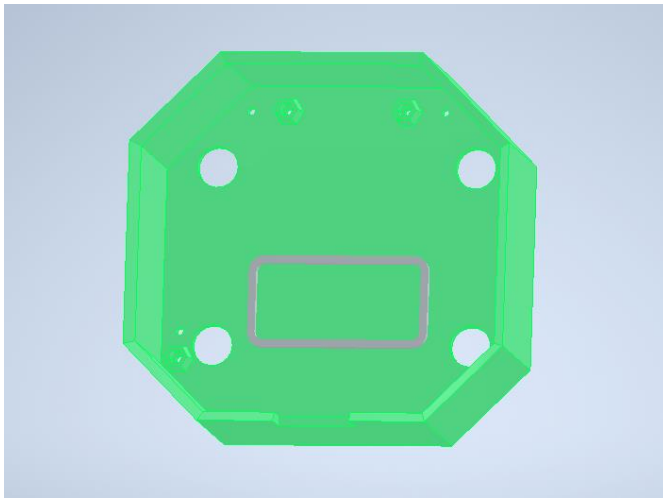
*Nota:* A la izquierda, se muestra la figura de la tapa del robot soccer en donde se colocará el pie de amigo y un interruptor. A la derecha, se observa el pie de amigo que funcionará como base para colocar el case de la cámara.

### 7.1.3 PARTE SUPERIOR O CUERPO DEL ROBOT

En la figura 14, se observa el cuerpo del robot o donde irá el control del robot. En esta pieza se colocarán componentes como: Raspberry Pi Zero y Arduino Nano. Además, irá colocada la batería de 3 celdas o 12 V.

**Figura 14**

*Parte Superior del Robot*



*Nota:* Estructura de segundo piso del robot donde se colocarán solo los componentes de control.

#### **7.1.4 PARTE INFERIOR O BASE DEL ROBOT**

En la figura 15, se observa la base del robot o donde irá la parte de fuerza del robot. En esta pieza se colocarán componentes como: motores Pololu, motor sin escobilla, puente H 222aak y módulos de regulador de voltaje 5v.

**Figura 15**

*Parte Inferior del Robot*



*Nota:* Estructura del primer piso del robot donde se colocarán solo los componentes de fuerza.

## 7.2 DIAGRAMA DE CONEXIONES PROPUESTO

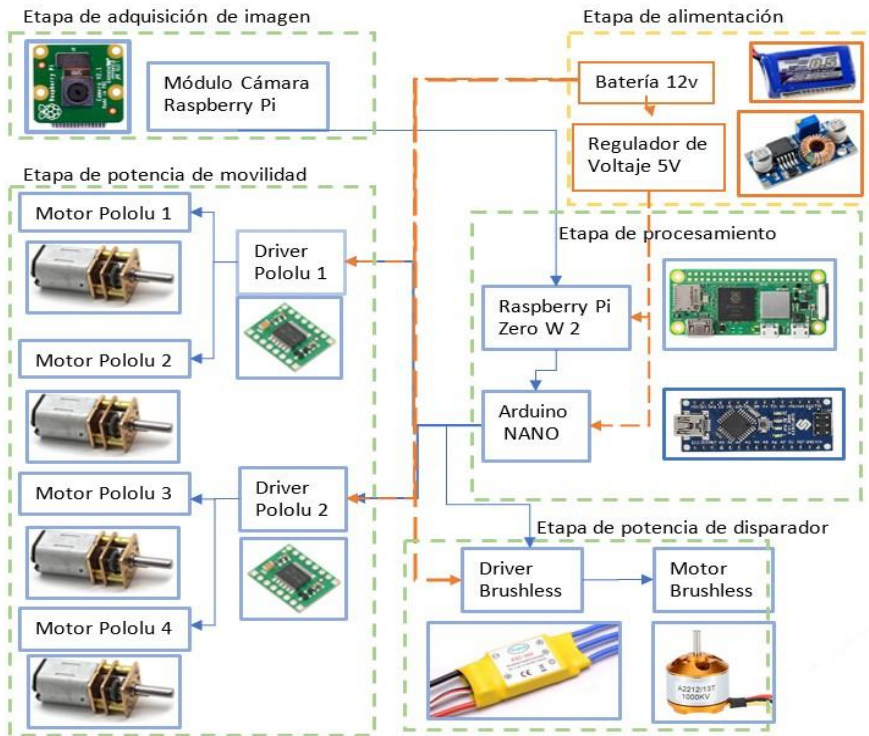
En la figura 16, el diagrama de conexiones del robot soccer consta de 5 etapas para un correcto funcionamiento del soccer para el cual son: etapa de alimentación, etapa de adquisición de imagen, etapa de potencia de movilidad, etapa de procesamiento y etapa de potencia de disparador.

A continuación, se detalla los componentes que forman parte en cada una de las etapas para el funcionamiento del robot:

1. ***Etapa de adquisición de imagen:*** se utiliza el módulo cámara Raspberry Pi para el procesamiento de imágenes y envío de datos.
2. ***Etapa de alimentación:*** se usa una batería de 12 v (3 celdas) para la alimentación del circuito y reguladores de voltaje que van conectados a los puentes H y Arduino Nano.
3. ***Etapa de potencia de movilidad:*** lo conforman los motores Pololu y puentes H 222aak.
4. ***Etapa de procesamiento:*** son los componentes que forman parte del proceso de control del robot. Estos son: Raspberry Pi Zero y Arduino Nano.
5. ***Etapa de potencia de disparador:*** forman parte el motor DC sin escobilla y el controlador driver para este.

**Figura 16**

*Diagrama de conexiones propuesto*



*Nota:* Se presenta las 5 etapas para la conexión y funcionamiento del robot soccer autónomo.

## 7.3 PROGRAMACIÓN EN RASPBERRY PI Y ARDUINO NANO

### 7.3.1 CONTROL MEDIANTE ARDUINO NANO

En la figura 17, el programa comienza inicializando el Arduino Nano y configurando los pines de entrada/salida necesarios para controlar los motores y actuadores. Luego, el programa entra en un bucle donde lee entradas, ya sea desde la Raspberry Pi o desde sensores conectados al Arduino Nano.

Si se recibe un comando de movimiento válido, el programa activa los motores para

mover el robot según el comando recibido. Esto implica enviar señales a los controladores de motor adecuados para controlar la velocidad y dirección de los motores.

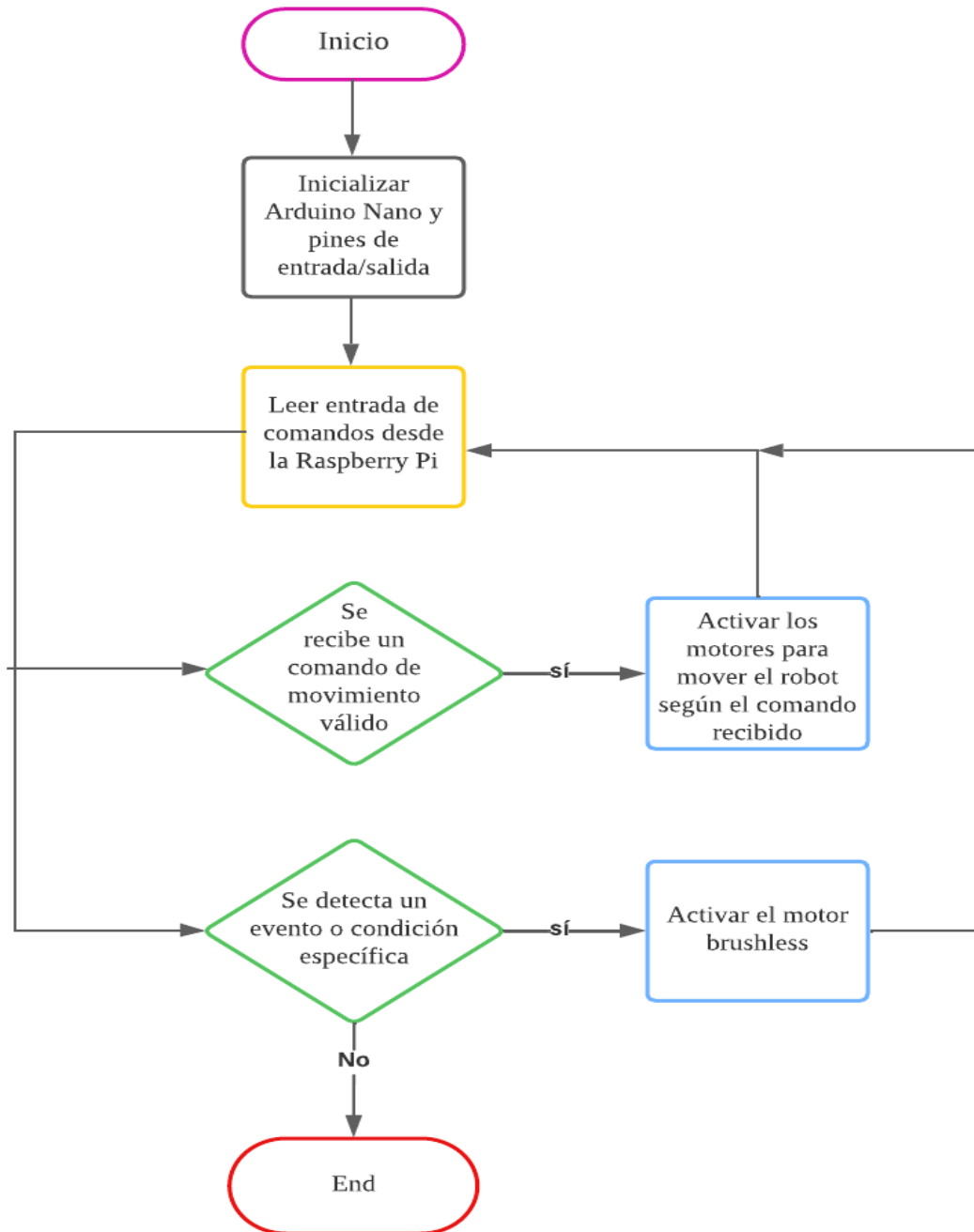
Además, si se detecta un evento o condición específica, el programa ejecuta acciones relacionadas con los actuadores, en este caso sería el motor brushless o motor DC sin escobilla.

Después de realizar las acciones necesarias, el programa vuelve al paso de "Leer entrada de comandos" para continuar leyendo y respondiendo a nuevas entradas.

A continuación, se presenta el diagrama de flujo en donde explica el funcionamiento del código en Arduino:

**Figura 17**

*Diagrama de flujo de Arduino Nano*



*Nota:* En este flujograma se detalla las condiciones que se involucran en el código de programación para el control de los motores Pololu y motor DC sin escobillas.



### 7.3.2 CONTROL MEDIANTE RASPBERRY PI ZERO

Este diagrama de flujo muestra el flujo general de procesamiento de la información en la Raspberry Pi Zero (figura 18). Comienza capturando una imagen desde la cámara conectada a la Raspberry Pi Zero. Luego, se puede realizar un preprocesamiento de la imagen si es necesario para mejorar la calidad o reducir el ruido.

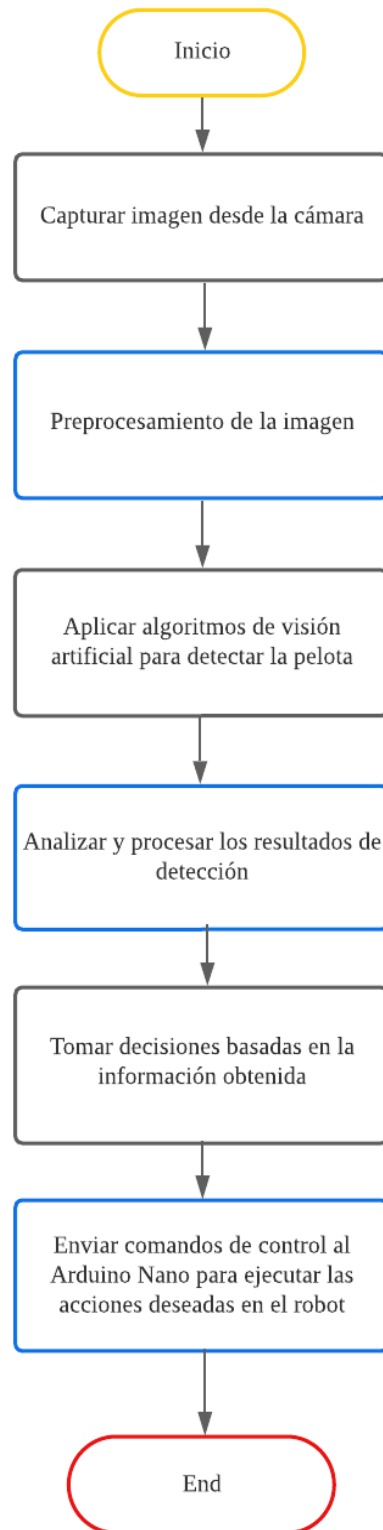
A continuación, se aplican algoritmos de visión artificial para detectar objetos de interés en la imagen, como la pelota. Una vez que se han detectado este objeto, se analizan y procesan los resultados de detección para obtener información relevante, como la posición y orientación de la pelota y los jugadores.

Utilizando esta información, se pueden tomar decisiones sobre las acciones que el robot debe realizar, como moverse hacia la pelota o evitar a los jugadores. Estas decisiones se traducen en comandos de control que se envían al Arduino Nano, que se encargará de controlar los motores y otros actuadores del robot para ejecutar las acciones deseadas.

Finalmente, el proceso se repite continuamente para que el robot pueda percibir su entorno, tomar decisiones y ejecutar acciones en tiempo real mientras juega al fútbol.

**Figura 18**

*Diagrama de flujo de Raspberry Pi Zero*



Nota: En el presente flujograma efectuado en el programa Thonny por medio de Python, se explica el proceso que realizará el código al momento de ejecutarlo, realizando el reconocimiento de la pelota.

## **7 RESULTADOS**

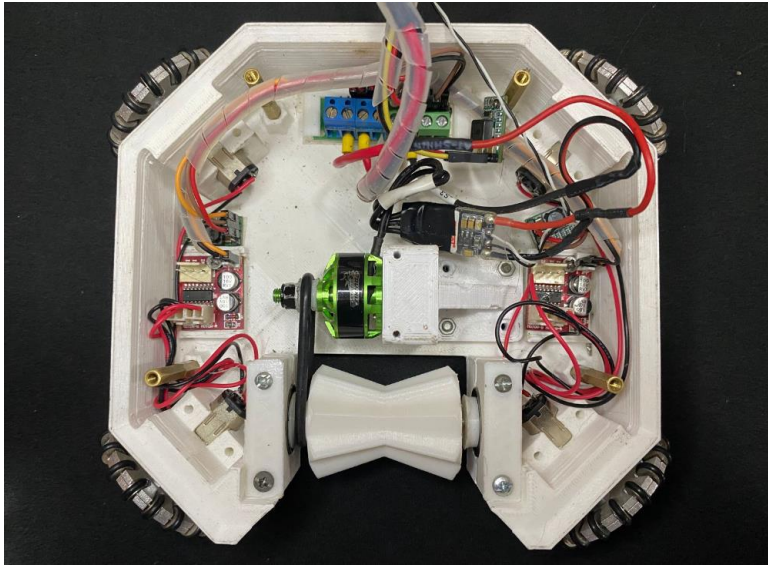
A continuación, se va a explicar el proceso de conexiones y ensamble del robot soccer, paso a paso.

En esta figura 19 se puede visualizar el primer nivel del robot de fútbol, en donde se ve la disposición organizada de los componentes previamente mencionados en este documento. Se ubican las ruedas omnidireccionales junto a sus correspondientes motores. Estos motores están interconectados a través de un puente H para control del movimiento de los motores, el cual a su vez se conecta a un regulador de voltaje. Este regulador de voltaje se enlaza al Arduino ubicado en la parte superior.

Adicionalmente, está presente el mecanismo de disparo. Este mecanismo se encuentra unido a una banda que proporcionará movimiento al motor sin escobillas. Dicho motor se conectará a un regulador y, a su vez, este regulador se conectará al Arduino.

## Figura 19

*Parte Inferior del Robot con los componentes de fuerza*

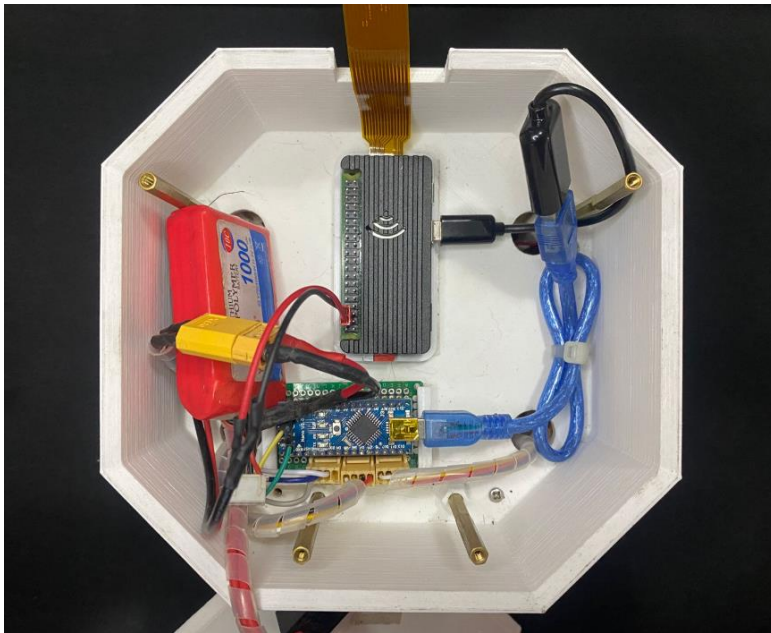


*Nota:* Se visualiza todas las conexiones explicadas en sus lugares correspondientes para un buen funcionamiento.

En la figura 20 se puede visualizar, el segundo nivel del robot, una serie de componentes interconectados. La batería de 12V desempeña un papel fundamental al proporcionar energía a todo el sistema. Esta energía se distribuye al Arduino Nano y, a su vez, al Raspberry Pi Zero. El Raspberry Pi Zero se enlaza a una cámara que proveerá información crucial para la ejecución de las funciones necesarias.

## Figura 20

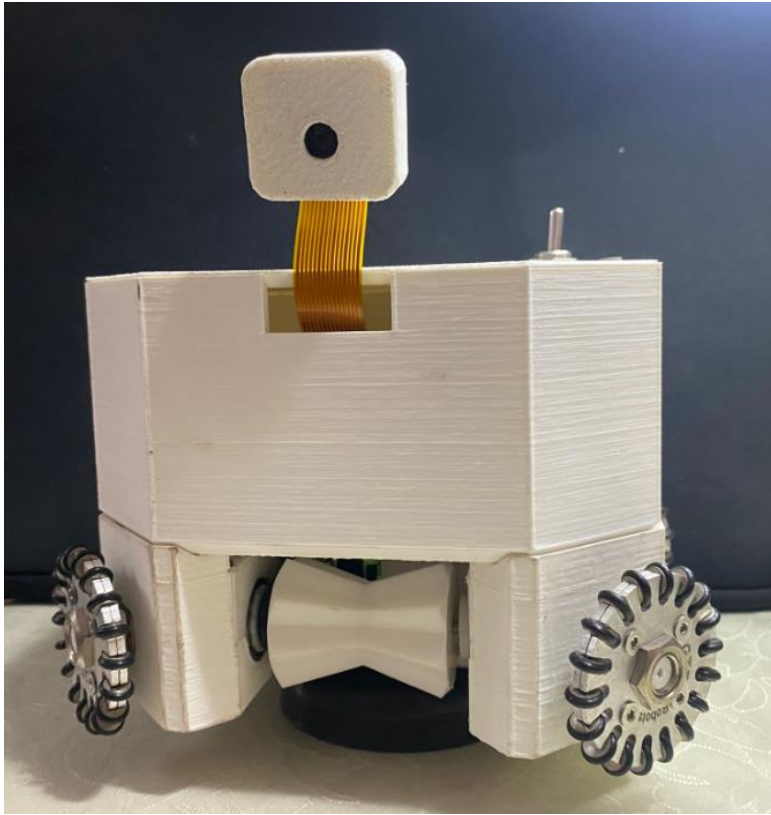
*Parte Superior del Robot con los componentes de control*



*Nota:* Se visualiza todas las conexiones, para su respectivo funcionamiento.

En la figura 21 en esta etapa, se visualiza nuestro robot de fútbol, completamente ensamblado y finalizado. Al observar sus componentes externos, destaca las ruedas omnidireccionales, el mecanismo de disparo, un pequeño interruptor que permite la activación y la cámara encargada de recolectar datos.

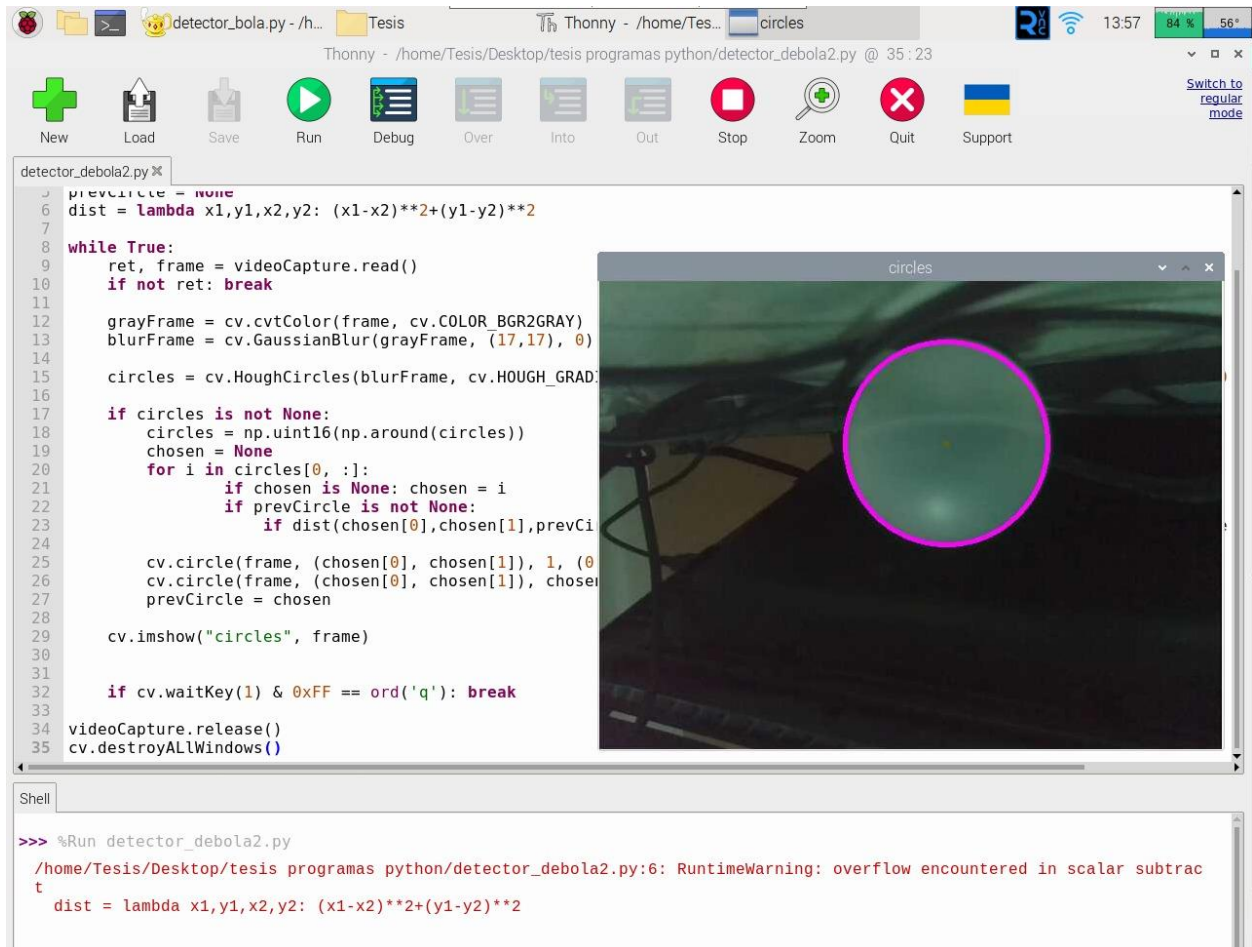
**Figura 21**



*Nota:* Se visualiza al robot soccer ensamblado listo y funcional.

En la figura 22, para esta sección, se muestra el código abierto en el Raspberry Pi Zero, permitiendo la visualización de la esfera o pelota. Este código tiene la función de segmentar los colores entre blanco y negro, además de aplicar un efecto difuminado para eliminar los ruidos y de esa manera detectar con facilidad al balón. Este proceso interno se ejecuta imagen por imagen, y su objetivo final es detectar la esfera; una vez que se ha identificado su objetivo, se activa el motor DC sin escobilla.

Figura 22



*Nota:* En este código da a entender el paso a paso de la visualización del robot para poder ubicar la pelota.

## 8 CRONOGRAMA

En la tabla 1, se muestra el plan de trabajo que se ha ejecutado durante los meses de mayo, junio y julio, donde se especifica las actividades que se realizaron para la elaboración del Proyecto de Titulación correspondiente al periodo académico 62.

**Tabla 1.**

*Cronograma de actividades para el desarrollo del proyecto de Titulación*

Plan De Trabajo De Proyecto De Tesis	Abr-23				May-23				Jun-23				Jul-23							
	Semanas																			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
Realizar Reuniones Con El Tutor Para Sugerencias E Ideas Del Proyecto																				
Diseño Del Modelo 3d E Impresión De La Estructura Del Robot Soccer																				
Adquisición Y Compras De Componentes Electrónicos.																				
Ensamblaje, Implementación Y Pruebas Del Robot																				
Desarrollo Y Programación Del Software Para Programar El Reconocimiento De Imágenes Del Robot Soccer																				
Verificación Del Funcionamiento Del Robot Soccer Y Prueba De Programación.																				
Realización Del Informe Y Presentación Del Proyecto.																				



## 9. PRESUPUESTO

En la tabla 2, se presenta el presupuesto estimado para la construcción de un robot soccer, donde se ha considerado los gastos para el ensamblaje y programación del prototipo.

**Tabla 2.**

*Presupuesto para la construcción de un robot soccer autónomo.*

Materiales	Unidades	Precio Unitario Sin impuestos	Envío	Precio Total
Impresión 3D	1	\$50.00	\$0.00	\$50.00
Raspberry Pi Zero 2 W	1	\$149.99	\$0.00	\$149.99
Módulo Camera Raspberry Pi	1	\$19.99	\$0.00	\$19.99
Micro SD 64 GB	1	\$7.50	\$0.00	\$7.50
Módulo regulador de Voltaje de 5 v	1	\$6.00	\$0.00	\$6.00
Ruedas omnidireccionales de 50 mm	4	\$35.00	\$0.00	\$140.00
Micro Motor HPCB 50:1 a 12V	4	\$20.65	\$6.45	\$89.05
Motor DC sin escobilla (Brushless)	1	\$25.00	\$5.00	\$30.00
Arduino nano	1	\$14.00	\$0.00	\$14.00
Puente H (Driver) para motores (Paquete de 4)	0.5	\$6.99	\$6.99	\$10.49
Batería 12 v (Alimentación)	1	\$30.00	\$5.00	\$35.00
Jumpers	1	\$5.00	\$0.00	\$5.00
			Subtotal	\$557.02
			Impuestos	56.19
			Total	613.205

## 10. CONCLUSIONES

1. Se ha logrado diseñar e implementar con éxito un robot de fútbol autónomo utilizando la combinación de la Raspberry Pi Zero y el Arduino Nano. El sistema embebido Raspberry Pi Zero proporcionó capacidades de visión artificial, mientras que el Arduino Nano se utilizó para el control y la comunicación del robot.

2. El robot desarrollado demostró capacidades de detección de la pelota, así como navegación autónoma. Estas funcionalidades son fundamentales para un juego autónomo efectivo y prometen aplicaciones en el campo del fútbol robótico.

3. Durante el desarrollo del proyecto, se superaron diversos desafíos técnicos, como la integración de los componentes electrónicos, la programación de los algoritmos de visión artificial y la coordinación de las acciones del robot.

4. Las aplicaciones potenciales del robot soccer autónomo son amplias y abarcan áreas como la robótica, la inteligencia artificial y la automatización industrial. Los avances logrados en este proyecto pueden ser utilizados para desarrollar sistemas autónomos en otros contextos y contribuir al avance de la tecnología en general.

## 11. RECOMENDACIONES

1. Mejora del rendimiento del sistema de visión: A pesar de los resultados positivos obtenidos, se recomienda explorar técnicas de optimización para mejorar el rendimiento del sistema de visión. Esto podría incluir la implementación de algoritmos más eficientes, el uso de técnicas de aceleración de hardware o la exploración de enfoques alternativos para la detección y reconocimiento de objetos.

2. Integración de técnicas de aprendizaje automático: Se sugiere investigar la integración de técnicas de aprendizaje automático en el sistema de visión y control del robot. El uso de algoritmos de aprendizaje automático podría mejorar la capacidad de detección, el reconocimiento de patrones y el rendimiento general del robot en situaciones de juego complejas.

3. Implementación de estrategias de juego más sofisticadas: El robot soccer autónomo desarrollado en este proyecto se centró en las funcionalidades básicas del juego. Se recomienda explorar y desarrollar estrategias de juego más avanzadas, como el análisis táctico, la toma de decisiones basada en el contexto del juego y la coordinación eficiente con otros robots en el equipo.

4. Evaluación en entornos reales: Aunque se realizaron pruebas exhaustivas del robot en diferentes escenarios, se recomienda llevar a cabo evaluaciones en entornos de juego reales. Esto permitirá evaluar el rendimiento del robot en condiciones más realistas y analizar su capacidad para adaptarse a situaciones cambiantes y dinámicas propias del juego de fútbol.

5. Documentación y compartición de resultados: Se recomienda documentar y compartir los resultados y el código fuente desarrollado durante el proyecto. Esto puede ser útil para otros investigadores y desarrolladores interesados en trabajar en proyectos similares, promoviendo así el intercambio de conocimientos y el avance de la comunidad.

## 12. BIBLIOGRAFÍA

*Arduino, Programar Fácil.* (18 de Abril de 2017). Recuperado el Agosto de 2023, de Detector de bordes Canny cómo contar objetos con OpenCV y Python:

<https://programarfacil.com/blog/vision-artificial/detector-de-bordes-canny-opencv/>

Bisson, S. (4 de Noviembre de 2021). *Revisión de Raspberry Pi Zero 2 W: el dispositivo de placa única de bajo costo obtiene una actualización de cuatro núcleos.* Recuperado el Agosto de 2023, de <https://www.zdnet.com/article/raspberry-pi-zero-2-w-review-performance-in-a-small-form-factor/>

*ecuarobot.* (20 de Febrero de 2020). Obtenido de Comenzando con el módulo de cámara

Raspberry Pi: <https://ecuarobot.com/2020/02/19/comenzando-con-el-modulo-de-camara-raspberry-pi/>

*fablab.uchile.* (9 de Abril de 2018). Recuperado el Agosto de 2023, de Robot Omnidireccional:

<http://www.fablab.uchile.cl/proyectos/268/robot-omnidireccional/>

*puntoflotante.* (2023). Recuperado el Agosto de 2023, de Driver L298N, dual, PWM, puente H, para control de velocidad y dirección de giro en motores de corriente directa.:

<https://www.puntoflotante.net/DRIVER-PWM-PUENTE-H-MOTORES-CD.htm>

Python Software Foundation. (2023). *Python Software Foundation.* Recuperado el Agosto de 2023, de Python: <https://docs.python.org/es/3/tutorial/>

ROBOWORKS. (Abril de 2023). *ROBOWORKS.* Recuperado el Agosto de 2023, de

Tags:Plataforma de Robot Móvil Autónomo para Interior:

<https://manuals.plus/es/roboworks/mecabot-autonomous-mobile-robot-platform-for-indoor-manual>

*studiopieters*. (4 de Octubre de 2022). Recuperado el Agosto de 2023, de Arduino® Nano – Pinout: <https://www.studiopieters.nl/arduino-nano-pinout/>

*Tardigrade Outdoors*. (2021). Recuperado el Agosto de 2023, de Overhauling a Monoprice MP Select Mini V2 MPSM V2 with SKR 1.4 board, octoprint, and more.: <https://www.tardigradeoutdoors.com/blog/overhauling-a-monoprice-mp-select-mini-v2-mpsm-v2-with-skr-14-board-octoprint-and-more>

*tytorobotics*. (3 de Noviembre de 2021). Recuperado el Agosto de 2023, de ¿Qué es un motor sin escobillas?: <https://www.tytorobotics.com/blogs/articles/what-is-a-brushless-motor>

*Universidad tecnica Federico Santa María*. (20 de Marzo de 2018). Recuperado el Agosto de 2023, de Sansanos clasifican a mundial de fútbol de robots autónomos RoboCup Soccer League 2018: <https://noticias.usm.cl/2018/03/20/sansanos-clasifican-a-mundial-de-futbol-de-robots-autonomos-robocup-soccer-league-2018/>

*visualgdb*. (29 de Septiembre de 2015). Recuperado el Agosto de 2023, de Using OpenCV with Raspberry Pi 2 Camera: <https://visualgdb.com/tutorials/raspberry/opencv/camera/>

## 13 ANEXOS

A continuación, se presentará en este apartado las siguientes imágenes:

**Anexo 1.** Plano estructural para la base y protección del módulo de la cámara ARDUCAM IMX219.

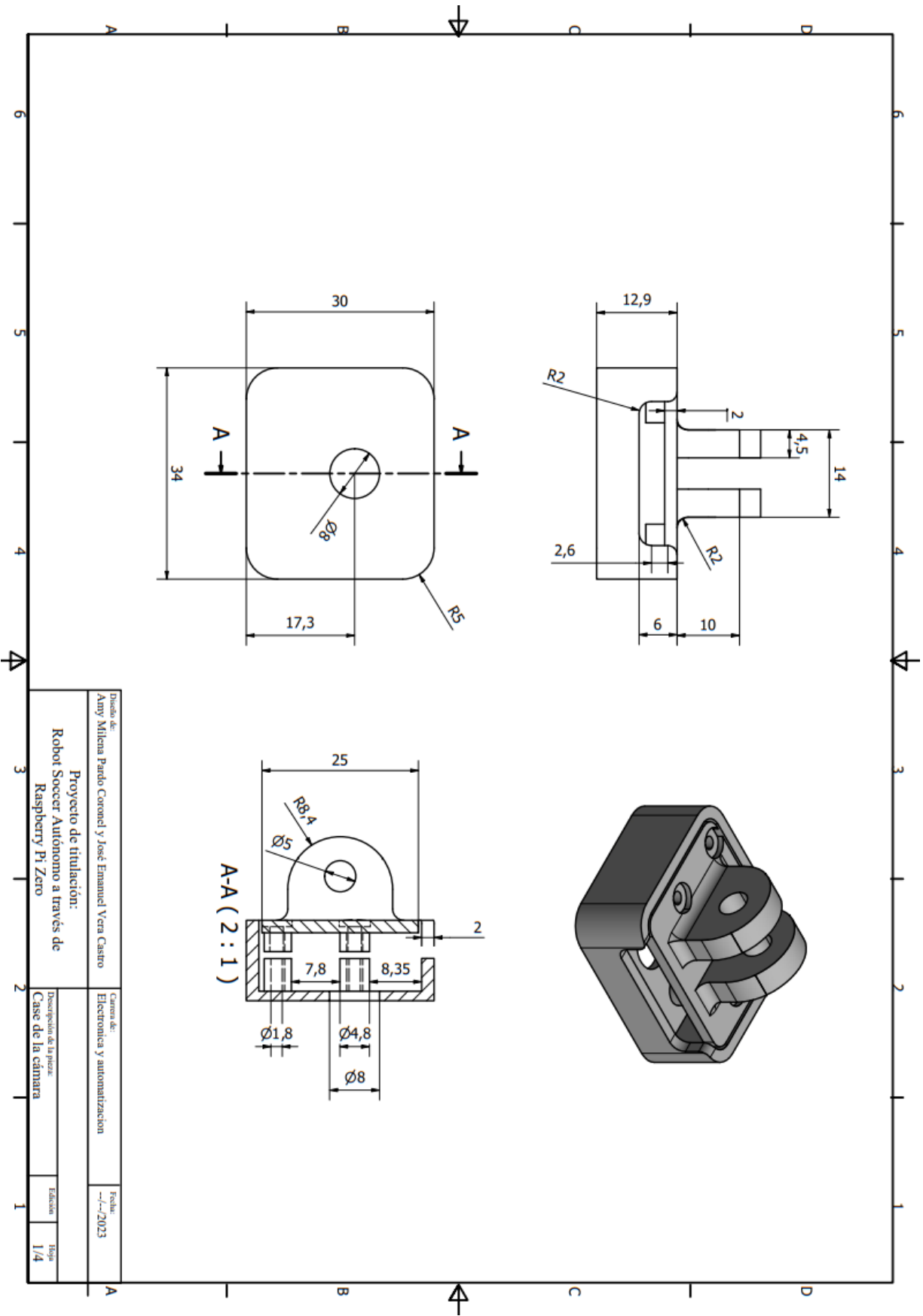
**Anexo 2.** Plano estructural de la tapa y soporte de la base para la cámara.

**Anexo 3.** Plano estructural de la base principal o piso bajo del robot soccer.

**Anexo 4.** Plano estructural de la base secundaria o piso alto del robot soccer.

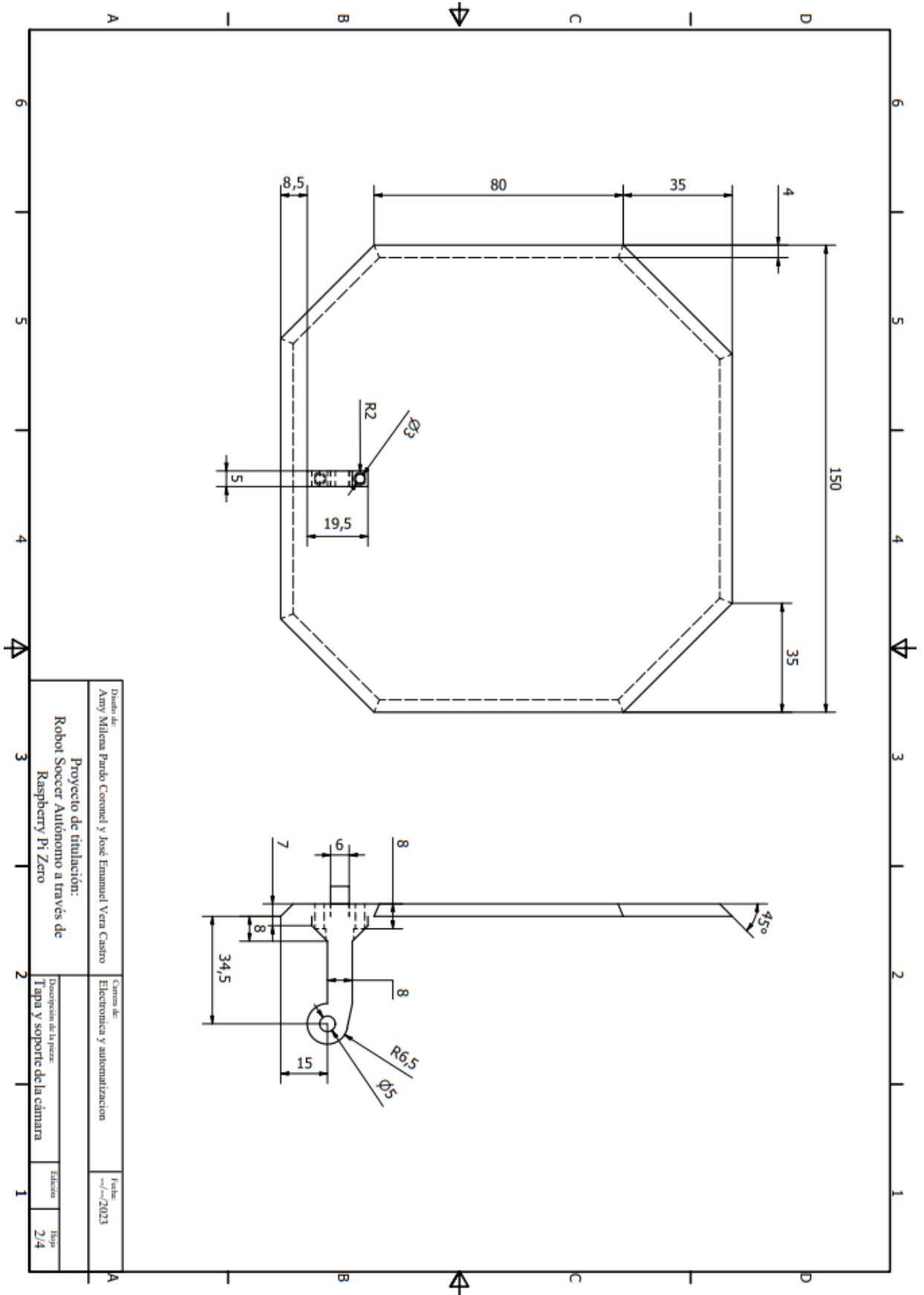
**Anexo 5.** Código de programación de control de motores Pololu y activación del motor brushless en Arduino.

**Anexo 6.** Código de programación de reconocimiento de balón en el programa Thonny.

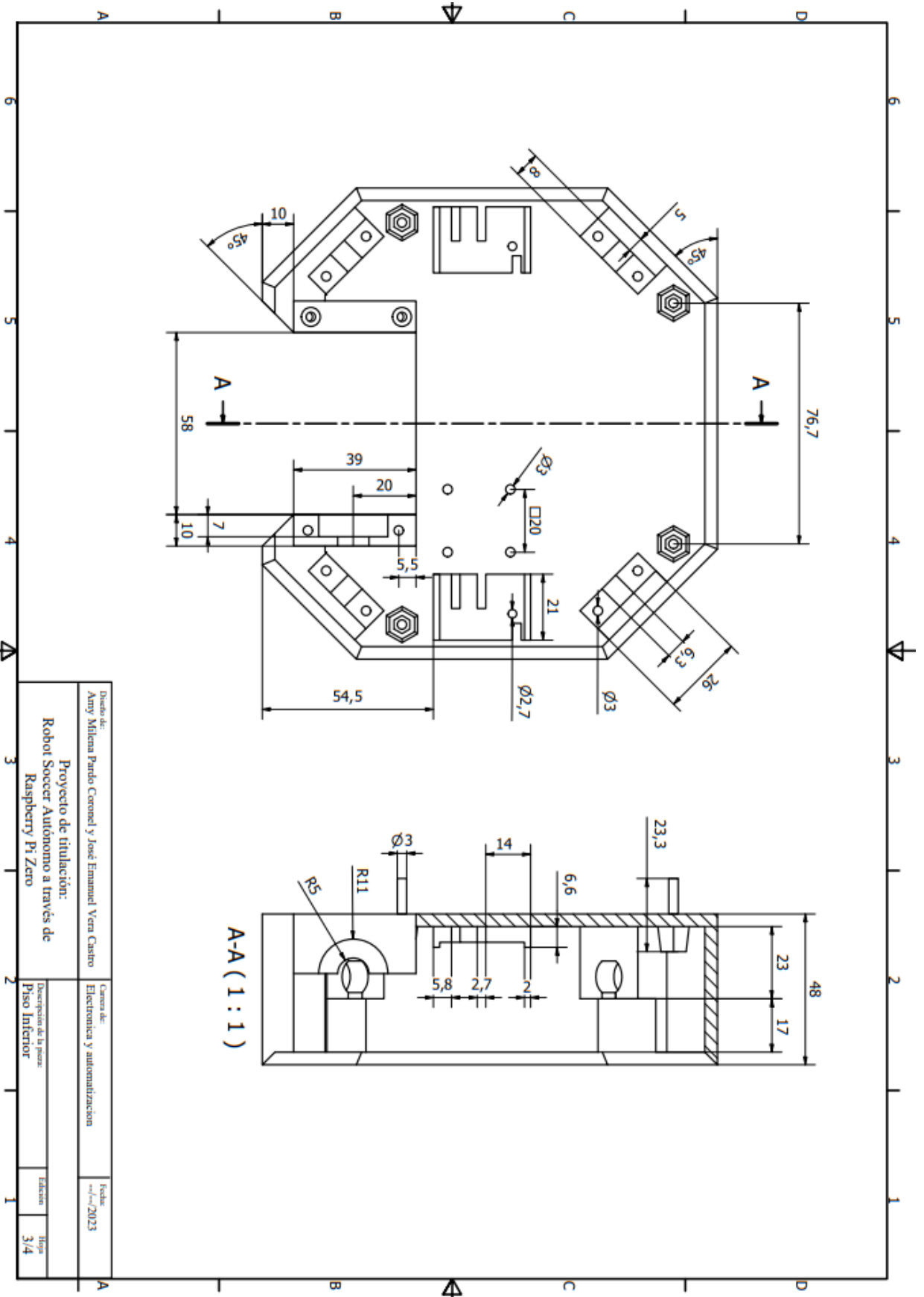


Diseño de: Amy Miliana Pardo Coronel y José Emanuel Vera Castro		Carrera de: Electrónica y automatización		Fecha: --/--/2023	
Proyecto de titulación: Robot Soccer Autónomo a través de Raspberry Pi Zero		Departamento de la pieza: Cáscara de la cámara		Edición: 1/4	

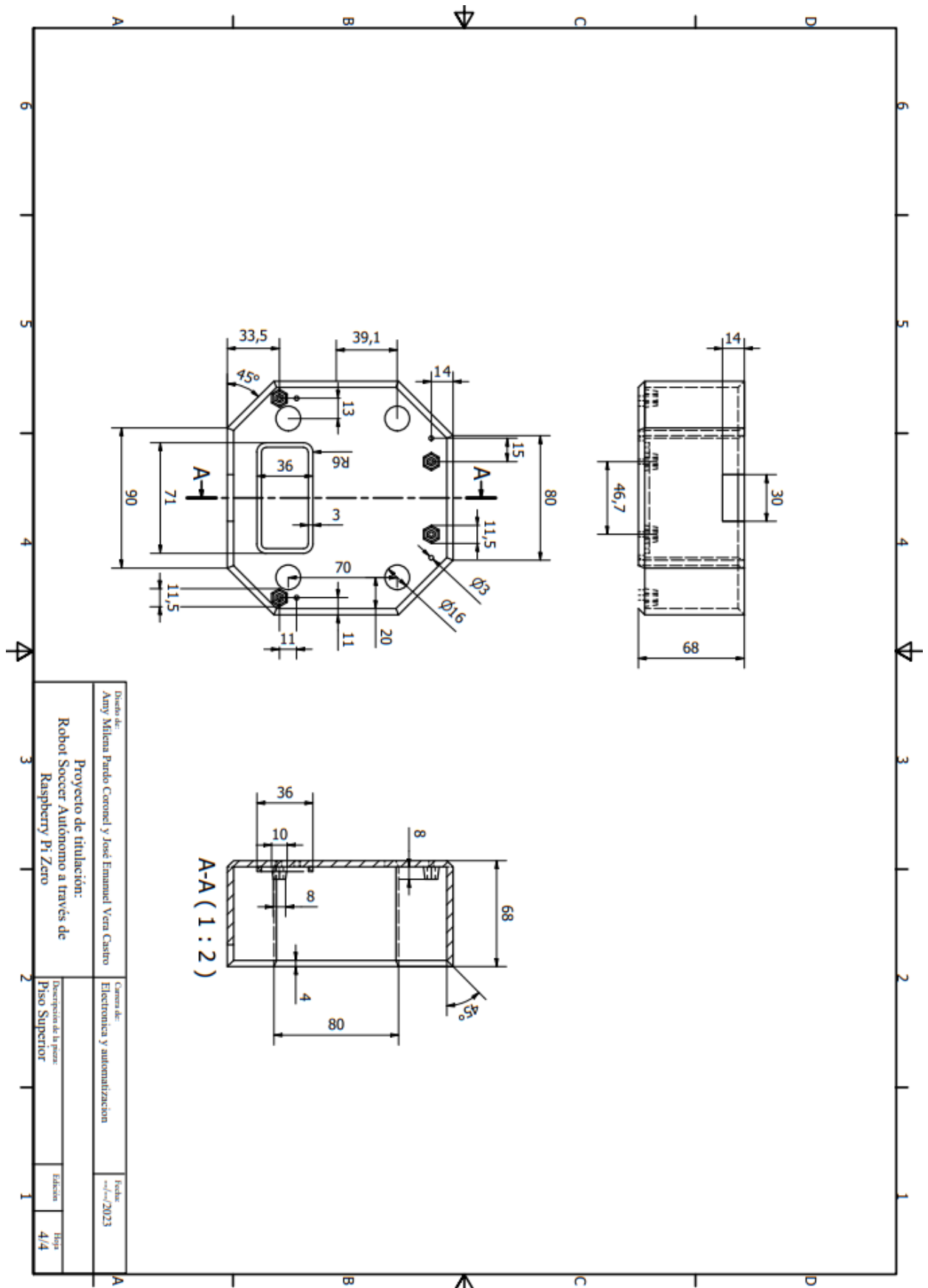




Diseño de: Amy Milena Pardo Coronel y José Emmanuel Vera Castro		Carrera de: Electrónica y automatización		Fecha: --/--/2023	
Proyecto de titulación: Robot Soccer Autónomo a través de Raspberry Pi Zero				Descripción de la pieza: Tapa y soporte de la cámara	
3				2	
A				1	



Diseño de: Amy Milena Pardo Coronel y José Emmanuel Vera Castro		Curso de: Electrónica y automatización		Fecha: **/**/2023	
Proyecto de titulación: Robot Soccer Autónomo a través de Raspberry Pi Zero			Descripción de la parte: Piso Inferior		
3			1		
Edición:			Hoja:		
			3/4		



Diseño de: <b>Robot Soccer Autónomo a través de Raspberry Pi Zero</b>		Carrera de: <b>Electrónica y automatización</b>		Fecha: <b>11/04/2023</b>	
Proyecto de titulación: <b>Robot Soccer Autónomo a través de Raspberry Pi Zero</b>		Descripción de la parte: <b>Piso Superior</b>		Edición: <b>4/4</b>	

## Control\_ESC\_no\_pwmsoft §

```
#include <Servo.h>
#include <SoftPWM.h>
Servo ESC;

int pos = 0; //Sets position variable
int incomingByte = 0, fase_arma = 0, control_motores = 0, estado_motor2 = 0, f=0, vel=0;
bool arma_boolean = 0;

int speed=0;
#define IN_23 12 // Driver #2 In 1 motor Trasero Derecho (Marcha Frontal)
#define IN_24 11 // Driver #2 In 2 motor Trasero Derecho (Marcha Trasera)
#define IN_22 10 // Driver #2 In 3 motor Frontal Derecho (Marcha Trasera)
#define IN_21 9 // Driver #2 In 4 motor Frontal Derecho (Marcha Frontal)

#define IN_14 8 // Driver #1 In 1 motor Trasero Izquierdo (Marcha Trasera)
#define IN_13 7 // Driver #1 In 2 motor Trasero Izquierdo (Marcha Frontal)
#define IN_11 6 // Driver #1 In 3 motor Frontal Izquierdo (Marcha Frontal)
#define IN_12 5 // Driver #1 in 4 motor Frontal Izquierdo (Marcha Trasera)

void setup() {
  Serial.begin(9600);
  SoftPWMBegin();

  pinMode(3, OUTPUT);
  pinMode(IN_11, OUTPUT);
  pinMode(IN_12, OUTPUT);
  pinMode(IN_13, OUTPUT);
  pinMode(IN_14, OUTPUT);

  pinMode(IN_21, OUTPUT);
  pinMode(IN_22, OUTPUT);
  pinMode(IN_23, OUTPUT);
  pinMode(IN_24, OUTPUT);
}
```

```
ESC.attach(3); //Adds ESC to certain pin. arm();
Serial.println("Estamos Listos");
setSpeed(0); //Sets speed variable
//ESC.write(0);
delay(500);
}

void loop() {

  if (Serial.available() > 0) {
    incomingByte = Serial.parseInt();
    Serial.println(incomingByte);

    if (incomingByte == 199){
      Serial.println("Arma Encendido");
      fase_arma = 1;
    }
    else if (incomingByte == 198){
      fase_arma = 3;
      Serial.println("Arma Apagado");
    }
    else if (incomingByte > 100 && incomingByte < 190){
      control_motores = incomingByte;
    }
    else if (incomingByte > 0 && incomingByte <= 100){
      vel = incomingByte;
      Serial.print("velocidad actual=");
      Serial.println(vel);

      vel = map(incomingByte, 0, 100, 0, 255);
      //Sets servo positions to different speeds
    }
  }

  //////////////////////////////////////
```

```
        else if (control_motores == 119){
//    Serial.print("Derecha");
        turn_right();
        }

        else if (control_motores == 117){
//    Serial.print("Detenido");
        turn_left();
        }

        else if (control_motores == 115){
//    Serial.print("Detenido");
        stopRobot();
        }
}

void on_arma(){
    for(speed = 0; speed <= 35; speed += 1) {
        setSpeed(speed);
        delay(100);
    }
}

void off_arma(){
    for(speed = 35; speed > 0; speed -= 1) {
        setSpeed(speed);
        delay(100);
    }
}

void setSpeed(int speed){
    int angle = map(speed, 0, 100, 0, 180);
    //Sets servo positions to different speeds
    ESC.write(angle);
}
```

```
if (fase_arma == 0) {
    //ESC.write(0);
    setSpeed(0);
}else if (fase_arma == 1){
    on_arma();
    fase_arma = 2;
}else if (fase_arma == 2){
    setSpeed(35);
}else if (fase_arma == 3){
    off_arma();
    setSpeed(0);
    fase_arma = 4;

}else if (fase_arma == 4){
    //off_arma();
    // fase_arma = 0;
}

    if (control_motores == 118){
        forward ();
        // Serial.print("Avance");
    }
    else if (control_motores == 112){
        // Serial.print("Reversa");
        back();
    }
    else if (control_motores == 114){
        // Serial.print("Izquierda");
        left();
    }
    else if (control_motores == 116){
        // Serial.print("Derecha");
        right();
    }
}
```

```
void stopRobot() {
    SoftPWMSet(IN_11, 0);
    SoftPWMSet(IN_12, 0);
    SoftPWMSet(IN_13, 0);
    SoftPWMSet(IN_14, 0);
    SoftPWMSet(IN_24, 0);
    SoftPWMSet(IN_23, 0);
    SoftPWMSet(IN_22, 0);
    SoftPWMSet(IN_21, 0);
}

void forward () {
    SoftPWMSet(IN_11, vel);
    SoftPWMSet(IN_12, 0);
    SoftPWMSet(IN_13, vel);
    SoftPWMSet(IN_14, 0);
    SoftPWMSet(IN_24, 0);
    SoftPWMSet(IN_23, vel);
    SoftPWMSet(IN_22, 0);
    SoftPWMSet(IN_21, vel);
}

void back() {
    SoftPWMSet(IN_11, 0);
    SoftPWMSet(IN_12, vel);
    SoftPWMSet(IN_13, 0);
    SoftPWMSet(IN_14, vel);
    SoftPWMSet(IN_24, vel);
    SoftPWMSet(IN_23, 0);
    SoftPWMSet(IN_22, vel);
    SoftPWMSet(IN_21, 0);
}
```



```
void right() {
    SoftPWMSet(IN_11, vel);
    SoftPWMSet(IN_12, 0);
    SoftPWMSet(IN_13, 0);
    SoftPWMSet(IN_14, vel);
    SoftPWMSet(IN_24, 0);
    SoftPWMSet(IN_23, vel);
    SoftPWMSet(IN_22, vel);
    SoftPWMSet(IN_21, 0);
}

void left() {
    SoftPWMSet(IN_11, 0);
    SoftPWMSet(IN_12, vel);
    SoftPWMSet(IN_13, vel);
    SoftPWMSet(IN_14, 0);
    SoftPWMSet(IN_24, vel);
    SoftPWMSet(IN_23, 0);
    SoftPWMSet(IN_22, 0);
    SoftPWMSet(IN_21, vel);
}

void turn_right() {
    SoftPWMSet(IN_11, vel);
    SoftPWMSet(IN_12, 0);
    SoftPWMSet(IN_13, vel);
    SoftPWMSet(IN_14, 0);
    SoftPWMSet(IN_24, vel);
    SoftPWMSet(IN_23, 0);
    SoftPWMSet(IN_22, vel);
    SoftPWMSet(IN_21, 0);
}
```

```

void turn_left() {
    SoftPWMSet(IN_11, 0);
    SoftPWMSet(IN_12, vel);
    SoftPWMSet(IN_13, 0);
    SoftPWMSet(IN_14, vel);
    SoftPWMSet(IN_24, 0);
    SoftPWMSet(IN_23, vel);
    SoftPWMSet(IN_22, 0);
    SoftPWMSet(IN_21, vel);
}

```

```

import cv2 as cv
import numpy as np

videoCapture = cv.VideoCapture(0)
prevCircle = None
dist = lambda x1,y1,x2,y2: (x1-x2)**2+(y1-y2)**2

while True:
    ret, frame = videoCapture.read()
    if not ret: break

    grayFrame = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
    blurFrame = cv.GaussianBlur(grayFrame, (17,17), 0)

    circles = cv.HoughCircles(blurFrame, cv.HOUGH_GRADIENT, 1.2, 100, param1=100,param2=30,minRadius=75, maxRadius=400)

    if circles is not None:
        circles = np.uint16(np.around(circles))
        chosen = None
        for i in circles[0, :]:
            if chosen is None: chosen = i
            if prevCircle is not None:
                if dist(chosen[0],chosen[1],prevCircle[0],prevCircle[1]) <= dist(i[0],i[1],prevCircle[0],prevCircle[1]):
                    chosen = i

        cv.circle(frame, (chosen[0], chosen[1]), 1, (0,100,100), 3)
        cv.circle(frame, (chosen[0], chosen[1]), chosen[2], (255,0,255), 3)
        prevCircle = chosen

    cv.imshow("circles", frame)

    if cv.waitKey(1) & 0xFF == ord('q'): break

videoCapture.release()
cv.destroyAllWindows()

```