



**Escuela Universitaria  
Politécnica** - La Almunia  
Centro adscrito  
**Universidad Zaragoza**

**ESCUELA UNIVERSITARIA POLITÉCNICA  
DE LA ALMUNIA DE DOÑA GODINA (ZARAGOZA)**

# **MEMORIA**

**EUROBOT 2025**

**Trabajo fin de grado**

Autor: Julio Cesar Balanta Narvaez

Director: Javier esteban escaño

Fecha: 14/09/2025

## INDICE DE CONTENIDO BREVE

1. RESUMEN	1
2. ABSTRACT	2
3. INTRODUCCIÓN	3
4. DESARROLLO	59
5. EJECUCION DE PRUEBAS Y PROBLEMAS SURGIDOS	139
6. CUMPLIMIENTO DE OBJETIVOS	141
7. CONCLUSIONES	144
8. BIBLIOGRAFIA	145

## INDICE DE CONTENIDO

### Contenido

EUROBOT 2025	1
Trabajo fin de grado	1
1. RESUMEN	14
2. ABSTRACT	2
3. INTRODUCCIÓN	3
3.1. Contextualización y motivación	3
3.2. Normativa y objetivos del proyecto	4
3.3. Metodología que se llevara a cabo	5
3.4. Antecedentes	8
3.4.1. "Feed the world" EUROBOT 2010	8
3.4.2. "Treasure Island" EUROBOT 2012	9
3.4.3. "Treasure Island" EUROBOT 2012	10
3.4.4. "Robomovies" EUROBOT 2015	13
3.4.5. EUROBOT 2016 - THE BEACH BOTS	15
3.4.6. EUROBOT 2021 – SAIL THE WORLD	17
3.4.7. EUROBOT 2023 - THE CHERRY ON THE CAKE	19
3.4.8. INTEGRACIÓN DE VISIÓN ARTIFICIAL EN UN ROBOT INDUSTRIAL	21
3.4.9. ROBOT SEGUIDOR	23

3.4.10.	Desarrollo de un sistema de control de un robot móvil.....	24
<b>3.5. Marco Teórico .....</b>		<b>27</b>
3.5.7.	Cinemática diferencial. ....	27
3.5.8.	Tecnología Lidar.....	34
3.5.9.	Motores de corriente continua y reguladores correspondientes. ....	35
3.5.10.	Otros modelos cinemáticos .....	42
3.5.11.	Sensores de entorno.....	43
4.2.7.	Materiales y cálculos estructurales .....	44
3.5.13.	Sistema de almacenamiento de energía .....	45
4.2.10.	Control de locomoción en Robots cuadrúpedos .....	46
4.2.11.	Visión Artificial y Procesamiento de Imágenes .....	48
4.2.12.	Diseño y Fabricación de Estructuras Robóticas .....	50
4.2.13.	Sistemas de Control y Electrónica.....	51
4.2.14.	Estado del arte: Estudio de tecnologías y limitaciones.....	53
<b>4. DESARROLLO.....</b>		<b>59</b>
<b>4.1. DIAGRAMA DE BLOQUES .....</b>		<b>59</b>
4.1.1.	Diagrama de alto nivel .....	59
4.1.2.	Diagrama de bajo nivel .....	59
4.1.3.	Diagrama de bajo nivel con referencias comerciales .....	64
4.1.4.	Layout .....	65
<b>4.2. Diagramas de funcionamiento .....</b>		<b>68</b>
4.2.1.	UML casos de uso .....	68
4.2.2.	UML actividad .....	71
4.2.3.	Diagrama de arquitectura (clases).....	75
<b>4.3. CALCULO DE MOTORES, ACTUADORES Y BATERIA .....</b>		<b>77</b>
4.3.1.	CALCULO DEL SISTEMA DE LOCOMOCIÓN.....	77
4.3.2.	Calculo motor paso a paso.....	80
4.3.3.	Calculo ventosa de succión.....	82
4.3.4.	Calculo alimentación.....	83
<b>4.4. ELECCIÓN DE MATERIALES.....</b>		<b>84</b>
4.4.1.	Elección servomotor .....	84
4.4.2.	Sensor presión pinza .....	86
4.4.3.	Elección ruedas .....	86
4.4.4.	Elección del botón de encendido.....	87
4.4.5.	Final carrera cordón arranque .....	88
4.4.6.	Sensor de ultrasonidos .....	88
4.4.7.	Sensor de color .....	88
4.4.8.	Sensor de infrarrojos .....	89
4.4.9.	Elección de pinza .....	89

4.4.10.	Elección bomba de vacío .....	90
4.4.11.	Elección sensor de presión .....	90
4.4.12.	Plataforma de elevación .....	91
4.4.13.	Elección de cámara .....	92
4.4.14.	Elección BMS .....	92
4.4.15.	Elección microcontrolador .....	93
4.4.16.	Driver motor paso a paso .....	93
4.4.17.	Final de carrera .....	93
4.4.18.	Puerto de carga .....	94
<b>4.5.</b>	<b>ESQUEMA ELECTRICO .....</b>	<b>95</b>
<b>4.6.</b>	<b>DISEÑO MECANICO .....</b>	<b>97</b>
<b>4.7.</b>	<b>Esquema electrónico .....</b>	<b>109</b>
4.7.1.	Fuente de alimentación .....	110
4.7.2.	Driver motor paso a paso .....	113
4.7.3.	Driver motores motrices .....	115
4.7.4.	Indicadores Leds .....	117
4.7.5.	Sensor de ultrasonidos .....	119
4.7.6.	Adaptador de nivel.....	119
4.7.7.	Conectores servomotores .....	120
4.7.8.	Final carrera plataforma .....	121
4.7.9.	Final de carrera cordón de arranque.....	121
4.7.10.	Conector sensor infrarrojo .....	122
4.7.11.	Conector bomba de vacío con sensor de presión .....	122
4.7.12.	Sensor de color .....	123
4.7.13.	ESP32 CAM .....	124
4.7.14.	ESP32 S3 WROON 1 .....	124
<b>4.8.</b>	<b>Diseño de la PCB.....</b>	<b>125</b>
<b>4.9.</b>	<b>COMO ORIENTAR LA PROGRAMACIÓN .....</b>	<b>131</b>
4.9.1.	ESP32 CAM.....	131
4.9.2.	Conexión I2C .....	134
4.9.3.	Driver motores motrices .....	135
4.9.4.	Driver motor paso a paso .....	136
4.9.5.	Bomba de vacío.....	136
4.9.6.	Sensor de ultrasonido .....	136
4.9.7.	Sensor infrarrojo.....	136
4.9.8.	Programación servomotores .....	137
4.9.9.	Sensor de color .....	137
<b>5.</b>	<b>EJECUCIÓN DE PRUEBAS Y PROBLEMAS SURGIDOS .....</b>	<b>139</b>

6. CUMPLIMIENTOS DE OBJETIVOS.....	141
6.1. REGLAMENTO EUROBOT.....	141
6.2. OBJETIVOS GENERALES DEL TFG .....	141
7. CONCLUSIONES .....	144
8. BIBLIOGRAFÍA .....	145

## INDICE DE ILUSTRACION

Ilustración 1: robot completo 2012 .....	9
Ilustración 2: micro implementación.....	10
Ilustración 3: Estructura robot.....	11
Ilustración 4: locomocion.....	12
Ilustración 5: motores con ruedas.....	13
Ilustración 6: locomoción.....	14
Ilustración 7: servo con pinzas .....	14
Ilustración 8: evento de competencia 2016 .....	15
Ilustración 9: pinza con servomotor .....	16
Ilustración 10: Sensor presión.....	16
Ilustración 11: Montaje completo .....	17
Ilustración 12: Evento de competencia 2021.....	18
Ilustración 13: evento de competencia 2023.....	20
Ilustración 14: cámara navegación.....	21
Ilustración 15: pinza sijeccion .....	22
Ilustración 16: Chasis completo .....	25
Ilustración 17: Sensores de navegación .....	25
Ilustración 18: Pico Board .....	26
Ilustración 19: Tracción diferencial.....	28
Ilustración 20: Representación triciclo .....	29
Ilustración 21: Metodo de orugas.....	29
Ilustración 22: Ackermann.....	30
Ilustración 23: Representacion modelo omnidireccional .....	30
Ilustración 24: funcionamiento general Lidar.....	34
Ilustración 25: funcionamiento de un motor CC.....	36
Ilustración 26: Esquema interno del Brushless .....	38
Ilustración 27: Motores motrices.....	39
Ilustración 28: Métodos para el rastreo de objetos.....	49
Ilustración 29: Diagrama de alto nivel .....	59
Ilustración 30: Diagrama de bajo nivel.....	60
Ilustración 31: Interfaz (interacción usuario) .....	61
Ilustración 32: Sistema de manipulación .....	61
Ilustración 33: Sistema locomoción .....	62
Ilustración 34: Sistema navegación .....	62
Ilustración 35: Sistema alimentación .....	63
Ilustración 36: Diagrama bajo nivel con referencias comerciales .....	64
Ilustración 37: Vista delantera.....	65
Ilustración 38: Elementos de plataforma .....	66
Ilustración 39: Vista frontal.....	66

Ilustración 40: Sistema locomoción .....	67
Ilustración 41: Unidad de control .....	67
Ilustración 42: Diagrama genera.....	68
Ilustración 43: Diagrama de celebración .....	68
Ilustración 44: Diagrama de navegación .....	69
Ilustración 45: Diagrama de manipulación.....	70
Ilustración 46: Diagrama general .....	71
Ilustración 47: Diagrama celebración.....	72
Ilustración 48: Diagrama de navegación .....	73
Ilustración 49: Diagrama de manipulación.....	74
Ilustración 50: Diagrama general clases.....	75
Ilustración 51: Funcionamiento general.....	76
Ilustración 52: Pololu 37D.....	80
Ilustración 53: Nema 17.....	81
Ilustración 54: Calculo ventosa .....	82
Ilustración 55: Eleccion ventosa.....	82
Ilustración 56: Ventosa de succión .....	82
Ilustración 57: SM-14303R .....	84
Ilustración 58: FETEECH SCS225 .....	84
Ilustración 59: conexiones intentadas .....	85
Ilustración 60: Micro servomotor .....	85
Ilustración 61: Sensor resistivo .....	86
Ilustración 62: Rueda goma .....	86
Ilustración 63: Rueda bola .....	87
Ilustración 64: Interruptor .....	87
Ilustración 65: Cordón arranque .....	88
Ilustración 66: HC-SR04.....	88
Ilustración 67: TCS230 TCS3200 .....	89
Ilustración 68: Sensor infrarrojo .....	89
Ilustración 69: Pinzas de ABS.....	90
Ilustración 70: Bomba vacio VN-T1 .....	90
Ilustración 71: sensor de presión ventosa .....	91
Ilustración 72: Plataforma elevación.....	91
Ilustración 73: ESP32 CAM .....	92
Ilustración 74: BMS Lisolect .....	92
Ilustración 75: ESP32 S3 WROOM 1 .....	93
Ilustración 76: DRV8848 .....	93
Ilustración 77: Final carrera plataforma .....	94
Ilustración 78: Puerto de carga DCJ200 .....	94
Ilustración 79: ESQUEMA ELÉCTRICO LOCOMOCIÓN Y CONTROL .....	95
Ilustración 80: Verificacion perimetro .....	97
<i>Ilustración 81: verificación de altura.....</i>	98
Ilustración 82: Primera planta, vista 1 .....	99
Ilustración 83: Primera planta, Vista 2.....	100
Ilustración 84: Primera planta, vista inferior .....	101
Ilustración 85: Segunda planta, vista 1.....	102
<i>Ilustración 86: Segunda planta, vista 2.....</i>	102
Ilustración 87: Segunda planta, parte inferior .....	103
Ilustración 88: Tercera planta .....	104
Ilustración 89: Plataforma, vista 1 .....	105
Ilustración 90: Plataforma vista lateral.....	106
Ilustración 91: Robot completo, vista 1 .....	107
<i>Ilustración 92: robot completo, vista 2.....</i>	108

Ilustración 93: Tercera planta, parte delantera.....	109
Ilustración 94: Fuente de alimentación .....	110
Ilustración 95: DRV8848 Kicad .....	113
Ilustración 96: TB67H420FTG Kicad .....	115
Ilustración 97: Codificador y señal de pulso .....	117
Ilustración 98: Conectores ultrasonido .....	119
Ilustración 99: Adaptadores de nivel .....	119
Ilustración 100: Conector servo pancarta .....	120
Ilustración 101: Conector servo pinza .....	120
Ilustración 102: Sensor resistivo Kicad .....	120
Ilustración 103: Final de carrera plataforma .....	121
Ilustración 104: Final de carrera cordón arranque.....	121
Ilustración 105: Conector sensor infrarrojo .....	122
Ilustración 106: Conector bomba de vacío .....	122
Ilustración 107: Sensor de presión ventosa .....	123
Ilustración 108: Sensor de color con adaptador .....	123
Ilustración 109: ESP32 CAM Kicad.....	124
Ilustración 110: ESP32 S3 WROOM 1 Kicad .....	124
Ilustración 111: Calculo ancho de la pista.....	125
Ilustración 112: PCB vista superior.....	126
<i>Ilustración 113: PCB vista inferior .....</i>	<i>127</i>
Ilustración 114: PCB completa.....	128
Ilustración 115: PCB vista 3D capa superior.....	129
<i>Ilustración 116: PCB vista 3D capa inferior .....</i>	<i>130</i>
Ilustración 117: Peso robot.....	142

# 1. RESUMEN

En el siguiente proyecto se pretende hacer un documento para crear un robot de manera teórica, en la competición de eurobot 2025. Entonces lo que tengo que hacer en primer lugar es buscar los antecedentes para saber cómo han sido los proyectos de años anteriores, su tecnología, método de diseño; y así poder ver cómo podemos proceder en la competición de este año. También buscamos alguna referencia respecto a otra competición diferente al eurobot, para así poder tener una visión más amplia sobre la robótica y poder tener ideas innovadoras y algunas formas diferentes de implementación. En este proceso también podremos ver las dificultades que han tenido los proyectos de otros años y cómo los han solucionado, y así podemos hacernos una idea de cómo nos podremos enfrentar las dificultades y qué dificultades nos pueden surgir.

Una vez que ya hayamos investigado los antecedentes, entonces ya realizamos un estudio de las diferentes tecnologías que se hayan implementado, con el fin de elegir el sistema que mejor se adapte a nuestro proyecto y que sea lo más sencillo posible.

A la hora de realizar el diseño hay que tener en cuenta la normativa de eurobot 2025, ya que nos dan limitaciones por ejemplo en cuanto al tamaño del robot, al peso del mismo, algunos componentes electrónicos tampoco están permitidos. Entonces todo esto condiciona el desarrollo del robot, el cual tiene que conseguir ser lo más funcional posible y fácil de fabricar; y cumplir su objetivo en la competición que es conseguir la mayor cantidad de puntos posibles

## 2. ABSTRACT

This project aims to design and develop a theoretical framework for constructing an autonomous robot for the Eurobot 2025 competition, themed "Robot Rock Tour." The objective is to create a functional system that can perform specific tasks defined by the competition rules, such as organizing musical instruments, setting up decorations, managing storage spaces, and executing celebratory movements. The process begins with the analysis of previous Eurobot editions and similar robotics contests to identify technological trends, design strategies, and challenges encountered. This comparative study provides valuable insights into existing solutions and common errors, which are critical for optimizing our own design. Subsequently, a detailed study of the technologies involved is conducted, including locomotion systems, sensors (e.g., LIDAR, ultrasonic), controllers, power supply, and navigation algorithms. This allows for the selection of components and architectures that best suit the project's objectives and constraints imposed by the Eurobot 2025 rules, such as weight limits, maximum dimensions, and component restrictions.

A structured methodology is followed, encompassing theoretical modeling, 3D design, schematic diagrams, and system simulations. Although the construction of a physical prototype is not within the scope of this phase, the design ensures feasibility for future implementation.

Ultimately, this project serves as a comprehensive exploration of robotic system design, emphasizing adaptability, efficiency, and compliance with technical regulations. It reflects a practical application of engineering knowledge and contributes to the educational development of robotics in academic environments.

## 3. INTRODUCCIÓN

### 3.1. Contextualización y motivación

A día de hoy la robótica se ha convertido en algo fundamental de la ingeniería, bueno por razones obvias, porque al fabricar un robot tocas diferentes campos como lo que es la parte mecánica, la electrónica, electricidad y programación. Esto utilizado en diferentes áreas como por ejemplo puede ser la medicina entre otras cosas, lo interesante de aquí es que tienes que montar circuitos electrónicos con unos determinados componentes viendo sus especificaciones para ver si aguantan o si tiene la potencia necesaria para hacer la tarea que se le quiere encomendar. La competición como eurobot nos permiten poder fabricar un robot en función a una normativa específica, y con unas especificaciones técnicas determinadas. Esto es importante ya que a la hora de fabricar el robot vamos a usarlo en un terreno físico real. En este tipo de robot tengo que utilizar sistema de navegación que puede ser por sensores de posición cámaras o sensores que detecten objetos; también usaré algoritmos específicos para la navegación (Siciliano & Khatib, 2016). Este trabajo de fin de grado tiene como objetivo construir un prototipo de robot, como si se fuera a competir en la competición de eurobot, por lo tanto este proyecto no es meramente teórico sino que se llevará a cabo el montaje del mismo y su verificación de funcionamiento.

Este tipo de proyecto puede resultar un reto, ya que en la teoría todo funciona muy bien pero en la práctica suelen venir todos los problemas. Por eso hay que hacer pruebas, proceso de validación y verificar el funcionamiento antes de avanzar al siguiente paso si no es imposible.

Hay que tener en cuenta las tecnologías de el sistema de navegación y el sistema de manipulación; y con ello los sensores y actuadores, se debe de tener en cuenta de que estaremos en un entorno ambiental que puede tener condiciones cambiantes, como por ejemplo la temperatura entonces hay que hacer el robot que tenga un determinado margen a ciertas variaciones.

#### ¿Qué es Eurobot?

**Eurobot** es una competición internacional de robótica autónoma creada en 1998

por la asociación **Planète Sciences** (Francia), en colaboración con la **Eurobot Association**. Está dirigido a estudiantes universitarios principalmente, su principal objetivo es promover la investigación y la innovación entre las personas más jóvenes (Planète Sciences, 2024).

En cada edición se propone un nuevo reto temático. Los robots deben realizar tareas específicas sobre un tablero de juego de 3×2 metros en un tiempo limitado (90 segundos). La puntuación se obtiene según el cumplimiento exitoso de las misiones planteadas. Se compete en modo uno contra uno, con un robot por equipo (a veces dos), bajo normativa rigurosa.

El lema de **Eurobot 2025** es "*Robot Rock Tour*". Las tareas suelen incluir organizar instrumentos musicales, organizar conciertos, dejar macetas, agarrar determinados objetos dándole preferencia a algunos que tienen un color específico debido al puntaje; todo eso se debe hacer en función de unas normas rigurosas

## 3.2. Normativa y objetivos del proyecto

### Normativa técnica Eurobot 2025

El reglamento de Eurobot se actualiza cada año, y los equipos deben cumplir estrictamente los límites técnicos y de seguridad. Las especificaciones generales para 2025 son:

- **Autonomía completa:** No se debe controlar el robot ni mediante ningún mando ni nada, el robot debe tener decisión propia.
- **Perímetro máximo:** no debe exceder 1200 mm y con la pinza desplegada se permite hasta 1400 mm.
- **Tiempo de operación:** 100 segundos por partida.
- **Elementos obligatorios:** sistema de arranque con cuerda (al menos 500 mm de largo) y botón de emergencia.
- **Componentes prohibidos:** fuentes de fuego, líquidos, humo, radiofrecuencia no homologada, proyectiles, piezas cortantes.
- **Altura máxima:** El robot no debe exceder 350 mm de altura, se tolera una altura de hasta 25 mm adicionales para la parada de emergencia es decir hasta un máximo de 375 mm.

(Planète Sciences, 2024)

<https://www.eurobot.org/wp->

[content/uploads/2024/10/Eurobot\\_General\\_Rules\\_EN.pdf](content/uploads/2024/10/Eurobot_General_Rules_EN.pdf)

Estas restricciones hacen que el diseño deba ser compacto, resistente, ligero y eficiente. El sistema debe moverse rápidamente, detectar objetos y obstáculos, manipular elementos y ejecutar una estrategia autónoma sin errores.

## Objetivos del proyecto

### Objetivo general:

Desarrollar un prototipo funcional de robot autónomo que cumpla con los requisitos técnicos y estratégicos de la competición Eurobot 2025, optimizado para obtener la máxima puntuación.

### Objetivos específicos:

1. Diseñar una estructura mecánica ligera, resistente y modular, con un peso inferior a 10 kg.
2. Implementar un sistema de propulsión con una aceleración mínima de  $0,8 \text{ m/s}^2$  y una velocidad máxima de  $1.2 \text{ m/s}$ .
3. Incorporar un mecanismo de captura (ventosa o brazo) capaz de agarrar objetos en menos de 3 segundos.
4. Integrar sensores (ultrasónicos, infrarrojos o cámara) y sistemas de navegación autónoma eficientes.
5. Simular y probar el robot en entornos controlados antes de construir el prototipo físico.
6. Validar el prototipo a través de pruebas reales sobre tablero, siguiendo el reglamento oficial.

## 3.3. Metodología que se llevara a cabo

La metodología propuesta para el desarrollo de este Trabajo de Fin de Grado sigue un enfoque **iterativo, estructurado y progresivo**, donde cada etapa alimenta a la siguiente mediante realimentación técnica, análisis crítico y pruebas constantes. El objetivo final es llegar a un **prototipo funcional**, pasando por una fase sólida de estudio y planificación.

Las etapas principales son las siguientes:

## 1. investigación antecedentes

En primer lugar se llevará a cabo la investigación de proyectos anteriores como este o similares a éste, para así saber cuáles fueron los componentes que han cogido otros alumnos y que errores han sólido tener para intentar no cometer los mismos errores

## 2. Marco teórico

Luego se analizan de manera teórica en matemáticamente las tecnologías que se vayan a usar para fabricar el robot, teniendo en cuenta cuáles son sus limitaciones cuáles son los alcances que tienen en la actualidad cuáles son sus ventajas y desventajas frente a otros componentes.

## 3. Modelado funcional del sistema

Luego se realizarán la elaboración de diagramas de alto y bajo nivel, para así hacernos una idea general del funcionamiento del circuito, sobre todo de cara a las conexiones que vamos a hacer.

Luego también se harán la elaboración de los diagramas UML y los diagramas de arquitectura para ver cuál será el comportamiento del robot y tener una idea general de cómo se debe programar.

## 4. Selección de referencias comerciales

A partir de los diagramas elaborados se deben elegir componentes, que ya sean componentes comerciales, que son los que vamos a usar para fabricar el robot, obviamente hay que tener en cuenta sus especificaciones técnicas y ver si el microcontrolador o los drivers que cojamos pueden aguantar esos componentes.

## 5. Diseño estructural y esquemático

- **Layout físico:** esa es la primera parte del diseño, no debe ser un diseño exacto con todo ensamblado de manera exacta, es algo que da una idea de dónde deben ir los componentes y la forma más o menos general que va a tener el robot
- **Diseño 3D:** Luego usando el software de inventor ahí sí que crearemos de manera exacta como debe ser el robot, con todos sus anclajes con todos los componentes puesto en la posición exacta y con un tamaño más o menos parecido y lo más exacto posible al de la realidad.
- **Esquema eléctrico:** es el conexionado eléctrico y se realizará mediante el

software QelectroTech.

- **Esquema electrónico:** diseño del circuito electrónico lógico mediante el programa Kicad
- **Diseño de PCB:** creación de la placa electrónica personalizada del sistema, utilizando KiCad.

## 6. Fabricación e integración

Una vez validadas las simulaciones, se realiza la fabricación física: impresión 3D de piezas, ensamblado de la estructura, soldado de componentes y montaje del hardware completo. Para hacer la fabricación de la PCB usaré el fabricante JLCPCB

## 7. Validación y pruebas

Se efectúan ensayos funcionales para ver la funcionalidad. Se mide el rendimiento en tiempo real del robot en tareas como navegación, detección de objetos, manipulación y retorno a zona segura. Se evalúan métricas como precisión, velocidad, consumo energético y estabilidad.

## 3.4. Antecedentes

### 3.4.1. "Feed the world" EUROBOT 2010

#### 3.4.1.1. introducción

La prueba de este año en Eurobot 2010 se denomina "Feed the World" y consiste en recolectar alimentos: cereales (maíz), frutas (naranjas) y verduras (tomates); para hacer acopio de ellos y redistribuirlos entre los que más los necesitan.

Los alimentos están representados por dos tipos de bolas, rojas (tomates) y naranjas (naranjas), de distintos pesos, y por unos cilindros de polipropileno (maíz). Estos elementos están disponibles para los robots en distintos sitios de la mesa, ya sea en el suelo en posiciones fijas o aleatorias, o en posiciones elevadas. Los alimentos más difíciles de coger son los más pesados.

Los alimentos recolectados deben ser depositados en unos contenedores situados en el frente de la mesa. Gana el equipo que tenga más peso en su contenedor pasado el minuto y medio que dura la prueba. Compiten dos equipos compuestos por un robot autónomo cada uno.  
(SÁNCHEZ, 2010)

#### 3.4.1.2. Análisis del robot

Se utilizó un sistema de balizas, las cuales muestran la posición del robot en el terreno, evitando así colisiones.

Para el movimiento del robot se utilizó un sistema de tracción diferencial, que consiste en dos ruedas principales y una rueda-bola que es el apoyo para facilitar el movimiento y los giros. En cuanto al sistema de navegación se utilizó motores de corriente continua Bernio con sensores de proximidad y láseres para detectar obstáculos.

para evitar posibles inconvenientes en la superficie de competición, la arquitectura del chasis del robot se hizo mediante dos perfiles de aluminio con base plana y bordes rectos. para controlar la velocidad y giro se escogió un sistema de realimentación con encoder y un microcontrolador Renesas H8S/2638 para procesar la señal, debido a que es robusto y con un buen rendimiento.

Para evitar que los elementos de juego salieran del interior del robot durante algún movimiento o manipulación, se eligió un sistema de palas móviles (actuadores) que estaban accionadas por motores de corriente continua. Para elevar los objetos y poder depositarlos en los contenedores, se implementó un sistema elevador con guías y un motor de corriente continua para su uso.

**Trabajo fin grado**

En conclusión, se eligió un sistema de tracción diferencial para el movimiento, sensores y láseres para detectar objetos, sistema de balizas para saber la posición de los robots, para la manipulación se escogió palas móviles con un sistema elevador con guías. y todo esto cogió y deposita elementos en los sitios correspondientes (SÁNCHEZ, 2010)

### 3.4.2. "Treasure Island" EUROBOT 2012

#### 3.4.2.1. Introducción

En la edición de Eurobot 2012, titulada *Treasure Island*, Los robots debían coger monedas, lingotes de oro y otros objetos que estaban distribuidos en la isla; con el objetivo de acumular la mayor cantidad de puntos posible en 90 segundos. Cada equipo podía competir con hasta dos robots autónomos y debía interactuar con distintos elementos del terreno, como tótems, barcos y mapas. Por lo tanto se debería hacer una navegación precisa y un sistema de manipulación muy efectivo (Halvorsen, Myren, & Sperre, 2012, p. 12).

#### 3.4.2.2. Análisis del robot

Tenía una arquitectura hueca y contaba con dos brazos en los laterales para empujar los discos a su interior y así coger la mayor cantidad posible, en sus laterales también tenía unos brazos para poder coger las monedas. (Halvorsen, Myren, & Sperre, 2012, p. 54).

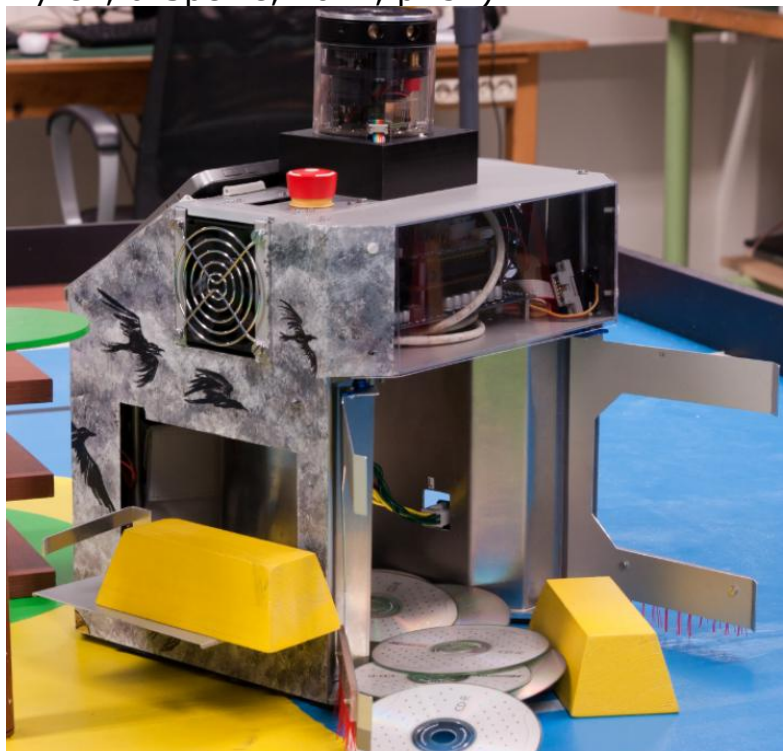


Ilustración 1: robot completo 2012

En cuanto al sistema de locomoción, se usaron dos ruedas motrices con una rueda de apoyo, un sistema de tracción diferencial básicamente; en donde las ruedas

**Trabajo fin de grado**

motrices estaban cerca del centro. Se utilizaron motores brushless Maxon EC-45 Flat, controlados por EPOS2 24/2, lo que ofrecía precisión y velocidad en los movimientos (Halvorsen, Myren, & Sperre, 2012, p. 57).

El sistema de navegación se basó en la fusión de sensores mediante un filtro de Kalman, combinando datos de los sensores Hall de los motores con un sistema de posicionamiento absoluto. Para ello se empleó un sistema de torres láser rotatorias, capaz de estimar ángulos y distancias, permitiendo localizar tanto al propio robot como a los oponentes (Halvorsen, Myren, & Sperre, 2012, pp. 58–60).

En cuanto al sistema de control, se implementó una arquitectura híbrida:

La unidad de control principal fue una tablet HP Slate 2 con arquitectura x86, corriendo Linux y programada en lenguaje Go

El control de bajo nivel se realizaba mediante un microcontrolador montado en una placa basada en la norma EPIC/PC104-plus, encargada de gestionar motores, servos, sensores y alimentación eléctrica. La comunicación entre ambos sistemas se hizo mediante CAN-bus. (Halvorsen, Myren, & Sperre, 2012, pp. 33–34).

### Hardware stack

A “PC104-plus” stack consisting of: A custom power-supply, capable of delivering 24V/12V/ 5V/3V3 @ 5A; a motor controller card; breakout-board.

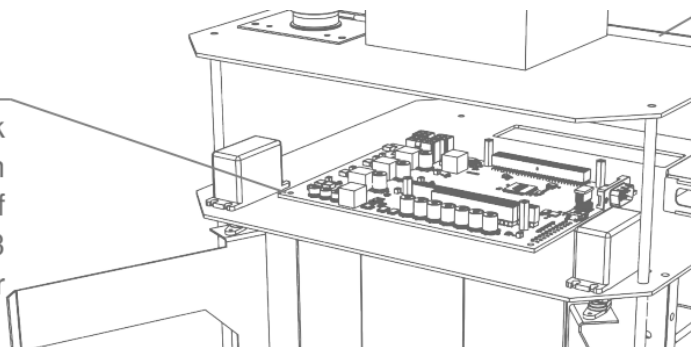


Ilustración 2: micro implementación

El sistema de manipulación se resolvió con servomotores que accionaban los brazos y alas, permitiendo recolectar objetos sin detenerse. También se implementó un sistema de evitación de obstáculos, basado en detección láser y estrategias de parada si un oponente se acercaba demasiado. (Halvorsen, Myren, & Sperre, 2012, p. 60)

En conclusión, se eligió un diseño compacto y funcional, centrado en garantizar la homologación y recolección de puntos seguros. El sistema de tracción diferencial junto con controladores PID permitía una navegación ágil. El uso del sistema láser y el filtro de Kalman proporcionaba una localización precisa. La estrategia de control basada en una tablet x86 y un microcontrolador separado permitió separar claramente la lógica de alto nivel del control físico. En conjunto, el robot logró una implementación robusta y modular, que culminó en la participación exitosa en la competición, finalizando en el puesto 23 de 43 equipos. (Halvorsen, Myren, & Sperre, 2012).

## 3.4.3. "Treasure Island" EUROBOT 2012

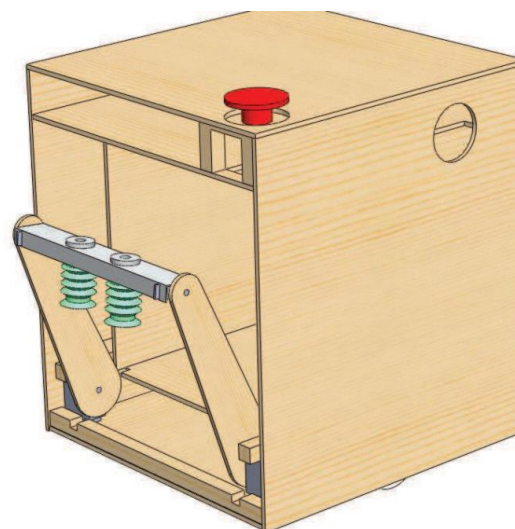
### 3.4.3.1. Introducción

### Trabajo fin grado

Esta competición de eurobot consistía en recolectar objetos en su interior que vendrían a hacer monedas que aquí se representaron con discos (discos CD sobre cubos de madera) y "lingotes" (bloques de madera) repartidos en el área de juego. El robot debía identificar y recoger estos elementos con precisión para almacenarlos en su interior (Halvorsen, Myren, & Sperre, 2012, pp. 58–60). Cada participante se tendría que enfrentar a dos contrincantes, el objetivo era que el que recolectara la mayor cantidad de objetos posibles en el interior del robot era el ganador. (Studený, 2012, pp. 9–10).

#### 3.4.3.2. Análisis del robot

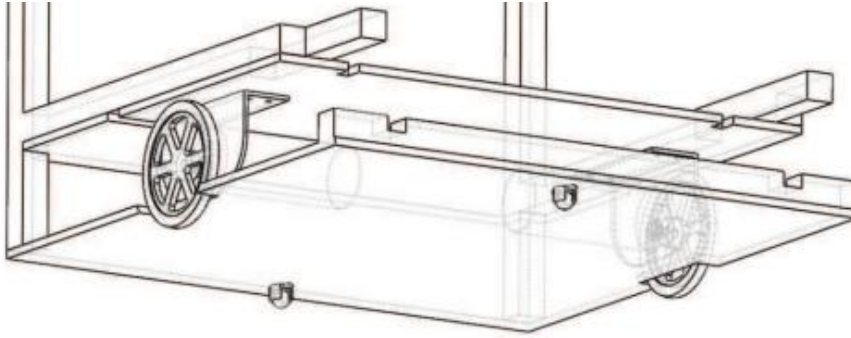
El robot fue diseñado con una estructura de madera (contrachapado y listones de abeto), buscando rigidez y simplicidad en el ensamblaje. El chasis permitía un amplio espacio interior para recoger la mayor cantidad de objetos posible. (Studený, 2012, p. 15).



*Ilustración 3: Estructura robot*

Para el sistema de locomoción se implementó un sistema de tracción diferencial con dos ruedas motrices EMG30 (con encoder para odometría) y Dos ruedas de apoyo para su estabilidad. (Studený, 2012, pp. 18–19).





*Ilustración 4: locomocion*

En el sistema de manipulación se utilizaron dos brazos articulares que estaban accionados con servos, luego tenemos los brazos que tenían ventosas para agarrar los objetos como los discos. La muñeca del brazo integraba dos ventosas con una separación ajustada para garantizar el agarre de los objetos, y el brazo estaba controlado por servos estándar y microservos (Studený, 2012, p. 26).

Usa una bomba de vacío de 12 voltios que iba conectada a un conector en T, una parte del conector iba conectado a un extremo donde había una válvula que se accionaba con un servo y eso permitía que se hiciera vacío y cuando hacía vacío se podía hacer la succión. Y de esa manera el sistema era capaz de agarrar el objeto sin hacer consumo continuo (Studený, 2012, p. 30–32).

La navegación del robot se apoyaba en sensores ultrasónicos (HC-SR04) ubicados en los extremos del robot, junto con la odometría (encoders) y una cámara web para guiar con precisión el manipulador. (Studený, 2012, p. 33).

Usa un Arduino nano para controlar los sensores y los actuadores y la comunicación con un netbook a bordo. La energía se suministraba con baterías Li-Poly de 3 celdas (12.6V) y 1000mAh, con sistemas de monitorización de voltaje y corriente. (Studený, 2012, p. 34, p. 38–40).

En conclusión, el robot se movía por tracción diferencial, detectaba el entorno con sensores ultrasónicos y cámara, manipulaba los objetos con un sistema de succión montado en un brazo articulado, y todo el sistema era controlado por una placa Arduino comunicada con un netbook central (Studený, 2012, p. 45).

### 3.4.4. "Robomovies" EUROBOT 2015

#### 3.4.4.1. Introducción

La edición del Eurobot 2015 se celebró en Yverdon-les-Bains, Suiza, se debastó en El Mundo del cine donde básicamente lo que había que hacer era por ejemplo organizar los escenarios poner los focos de luz poner alfombras rojas, para así poder hacer escenas de cine típicas.

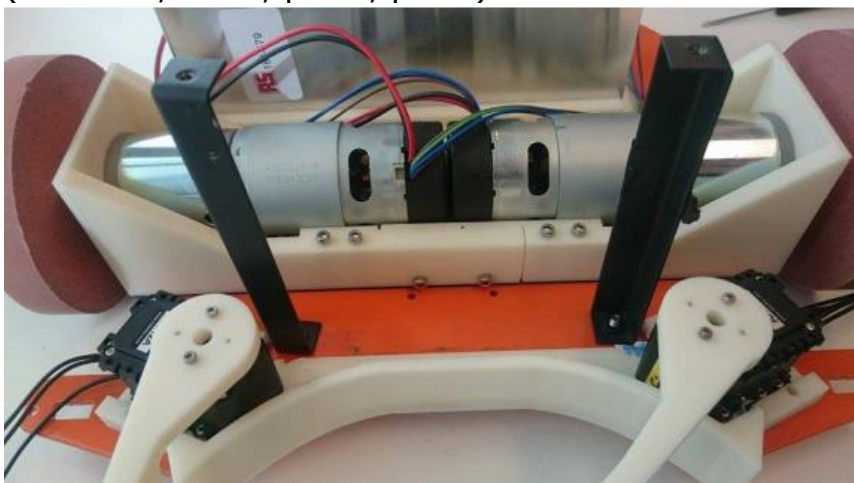
Debía tener un tamaño específico 3X2 metros, cada equipo podría presentar dos modelos de robot uno principal y uno secundario, en donde deberían cumplir obviamente con la reglamentación interruptor de emergencia, cuerda de arranque...

Las tareas puntuán individualmente salvo algunas excepciones que eran compartidas (como las palomitas) y otros específicos por equipo (como los focos o alfombra roja). Al finalizar la competición pues los puntos totales eran las sumas de todas las tareas (Wikheim, 2015, p. 1-4).

#### 3.4.4.2. Análisis del robot

Para el sistema de locomoción el robot principal se diseñó con una configuración de tracción diferencial, utilizando dos ruedas motrices a los lados y ruedas de soporte (rueda bola) para estabilidad. Esta arquitectura fue elegida por su simplicidad, precisión y bajo coste. Las ruedas se fabricaron mediante impresión 3D y se les añadieron neumáticos de silicona moldeada para mejorar la tracción sobre la superficie lisa del tablero. También se cogieron motores Maxon EC45 con un driver Devantech MD49

(Wikheim, 2015, p. 43, p. 47)



*Ilustración 5: motores con ruedas*



Figure 2.8: Common robot wheel types.

*Ilustración 6: locomoción*

Para el sistema de navegación inicialmente se planeó el uso de un sistema de posicionamiento global mediante balizas infrarrojas, pero fue descartado por su complejidad y baja fiabilidad. Finalmente, el robot se guió utilizando encoders y sensores de proximidad, implementando un patrón de movimiento "point-and-shoot", donde el robot gira hacia su objetivo y luego avanza en línea recta. (Wikheim, 2015, p. 34, p. 45).

En el sistema de manipulación el robot incorporó un sistema de pinzas motorizadas controladas por servomotores Dynamixel, montadas en una estructura elevadora accionada por un motor paso a paso. Esa elevadora estaba conectada a una correa que permitía subir y bajar los objetos recolectados. Además, para tareas específicas como las claquetas, se instalaron brazos robóticos simples en los laterales, que se desplegaban mediante servos al pasar por delante de los objetivos. (Kaiser, 2001, p. 11, p. 59).

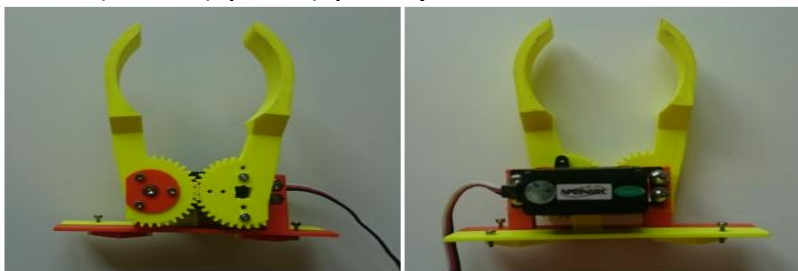


Figure 4.23: Gripper arms mounted on servo, connected by gear teeth.



(a) Servo

Specifications	
Operating Voltage	12V
Max Current	900mA
Stall Torque	1.5 Nm
No-load Speed	59 RPM
Angular Resolution	0.29°
Dimensions	32x50x40 mm
Weight	55g
Other Features	Sophisticated position controller

(b) Specifications

Figure 4.24: **Dynamixel** AX-12A servo motor.

*Ilustración 7: servo con pinzas*

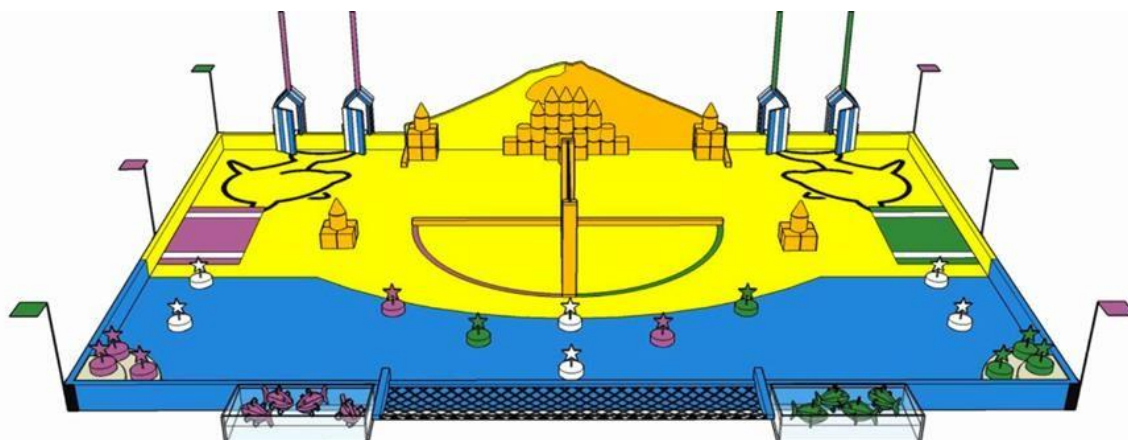
Se utilizó un portátil de 32 bits para la unidad de control ya que tenía bibliotecas permitía monitorizar y realizar todos los cambios pertinentes. Se utilizó Linux como sistema operativo y la comunicación entre módulos se manejó mediante ZeroMQ. El sistema de control estaba dividido en módulos como AI, navegación (NAVSYS) y sistema de control (CTRLSYS), cada uno operando de forma autónoma pero sincronizada. Para la unidad de control secundaria se usó un arduino mega 2560 y un arduino UNO, para leer los sensores y actuadores básicos. (Kaiser, 2001, p. 63).

En conclusión, para el Eurobot 2015, se optó por una arquitectura de bajo coste y fácil implementación. Se eligió una tracción diferencial por su sencillez y fiabilidad, descartando sistemas complejos como orugas o patas. La navegación se basó en encoders y sensores, abandonando el uso de triangulación por balizas debido a su baja efectividad en la práctica. La manipulación se realizó mediante palas elevadoras y brazos simples, adecuados para las tareas requeridas. El uso de un portátil como unidad central resultó práctico y económico. En conjunto, el diseño permitió al robot completar varias tareas del concurso con éxito, aunque con limitaciones frente a equipos con presupuestos más elevados. (Wikheim, 2015, p. 94)

### 3.4.5. EUROBOT 2016 - THE BEACH BOTS

#### 3.4.5.1. Introducción

En este proyecto se pedía fabricar robots para que atrapara los peces de los contenedores de agua, y se debe dejar en los lugares específicos. El robot que más peces deje es el que gana. Los peces se pueden ver representados en la imagen, a la izquierda esta en color morado y a la derecha en color verde; cada una para cada equipo. (Utrilla Bustamante, 2016, p. 7).



*Ilustración 8: evento de competencia 2016*

#### 3.4.5.2. Característica

Trabajo fin de grado

- Ascensor: Sistema de tornillo-tuerca accionado por motor paso a paso (Utrilla Bustamante, 2016, p. 9).
- Pinza: Mecanismo de sujeción mecánica con servomotor Hitec-805BB (Utrilla Bustamante, 2016, p. 9).
- Sensor de color: TCS3200 para distinguir objetos por color (Utrilla Bustamante, 2016, p. 10).

### 3.4.5.3. Análisis del robot

En esta edición se utilizó la tracción diferencial para el desplazamiento de los robots, que poseen motores individuales para cada rueda y mide la distancia y la velocidad mediante encoders. Para el sistema de agarre, se rechazó las pinzas por vacío y succión debido a su complejidad; se usó en su lugar una pinza de succión mecánica con un mecanismo de tornillo-tuerca controlado por un motor paso a paso. El sistema de detección de agarre de la pinza se basa en medir la presión ejercida (FSR o micro celdas de carga) o midiendo la corriente del servomotor de la pinza (Utrilla Bustamante, 2016, p. 9, p. 10).

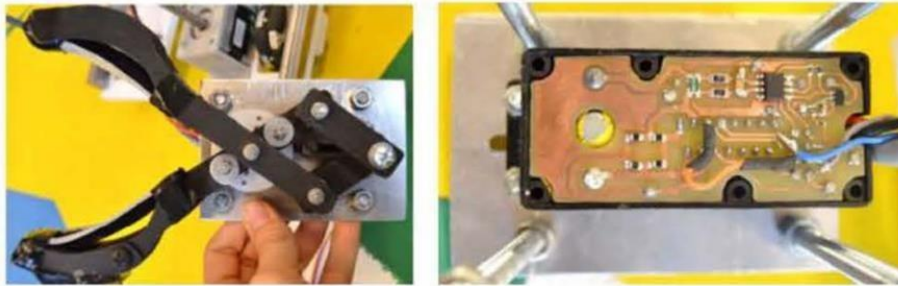


Ilustración 9: pinza con servomotor

Para que el robot transportara discos se plantearon alternativas: preconfigurar el robot o crear un sistema de detección de colores, se escogió la primera debido a la sencillez. Se escogió una arquitectura maestro-esclavo, donde el maestro era una Raspberry pi y utilizaba un controlador DRV8825 para el motor paso a paso del ascensor (Utrilla Bustamante, 2016, p. 10).



Ilustración 10: Sensor presión

En conclusión, la competición se baso en atrapar peces y dejarlos en un lugar específico, por lo que se pudo ver se intento usar sistemas y métodos no tan complejos de usar, y métodos innovadores para demostrar que funcionaria de manera igual. se terminó usando sistema de tracción diferencial, pinzas de sujeción mecánica y motores paso a paso. (Utrilla Bustamante, 2016, p. 79).

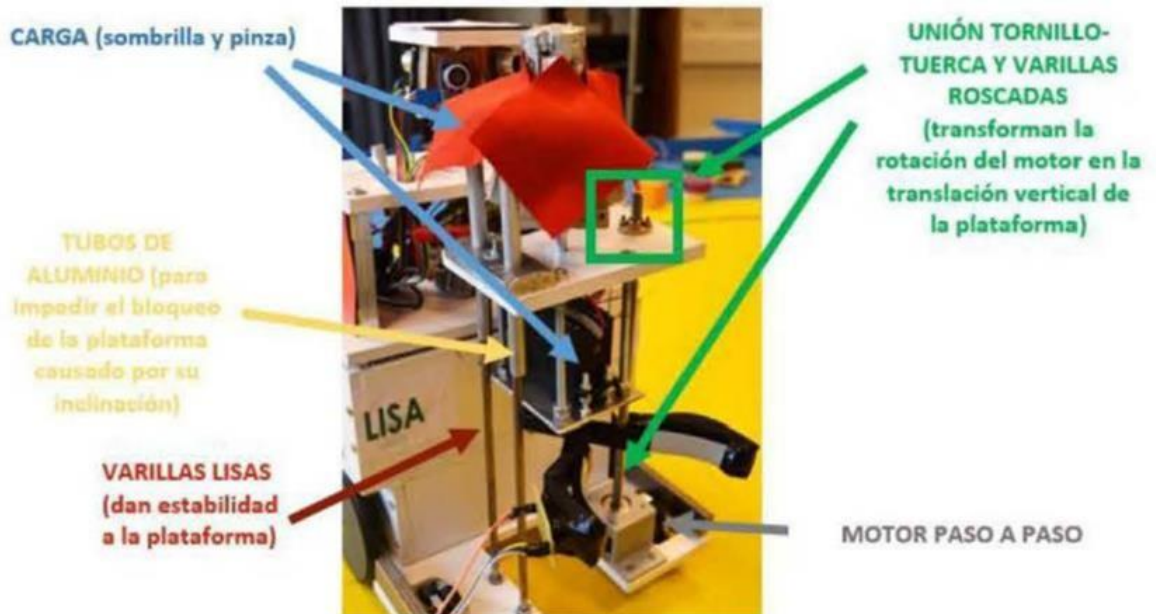


Ilustración 11: Montaje completo

### 3.4.6. EUROBOT 2021 – SAIL THE WORLD

#### 3.4.6.1. Introducción

La competición de este año se centra en la recolección de boyas, que se pueden ver representadas en la imagen como troncos huecos, el campo de entrenamiento está simulado en el mar y el objetivo es colocar las boyas en el puerto específico (Mamblona Nieto, 2021, p. 15-16).

En el campo hay un total de 50 boyas; de las cuales 20 están en dispensadores, 24 están repartidas por todo el campo de manera concreta y 6 están repartidas de manera aleatoria en una zona cercana a la torre. Cada una de estas boyas debe colocarse en los puertos y en función de su disposición correcta por color; Si se colocan en línea en el puerto y en función del color que componga la boya se obtendrá más o menos puntos (Mamblona Nieto, 2021, p. 16).



Ilustración 12: Evento de competencia 2021

### 3.4.6.2. Análisis del robot

Pasando al diseño del robot empezamos por el tipo de robot, en este caso el autor eligió un motor maxon DC de 15 W, Debido a su compacto tamaño y alta eficiencia. Se incorporó un encoder de 256 pulso por vueltas (ppv) y 3 canales, que proporciona información de manera más precisa de la posición al sistema de control para así poder realizar una navegación más exacta (Mamblona Nieto, 2021, p. 49-50).

La elección de una reductora planetaria de cerámica 26:1 se escogió en base a su capacidad para reducir la velocidad y aumentar el par del motor. Se utilizaron servos controladores ESCON 70/10 para así poder garantizar un control mucho más preciso del motor para que así pueda tener una mejor autonomía (Mamblona Nieto, 2021, p. 49-50).

La arquitectura del robot se configuró de forma paralela, debido a su simplicidad a pesar de las posibles restricciones de tamaño. Entonces esto simplifica la estructura del robot y de esta manera facilita su diseño y construcción debido a la facilidad de la implementación. También se implementaron engranajes y poleas para poder hacer frente a la posible inestabilidad del robot de esta manera se consiguió un par óptimo y una velocidad de transmisión en las ruedas para que el robot tendrá un mejor movimiento. (Mamblona Nieto, 2021, p. 64-66).

En este caso el autor eligió un LIDAR, como un sistema de corrección de movimiento, el LIDAR proporciona una información más detallada y también se utiliza para el aprendizaje del entorno. Las

ruedas motrices seleccionadas tienen una especie de cilindro con ranuras y juntas tóricas para una mejor sujeción y minimizar las pérdidas por tracción (Mamblona Nieto, 2021, p. 69).

El robot tiene un diseño de tal forma que el espacio está optimizado y reutilizable. El control del movimiento se hizo con los motores Maxon y un Arduino, la gestión de la odimetría se llevó a cabo mediante Raspberry pi para los cálculos estimados de la posición (Mamblona Nieto, 2021, p. 69-70).

En conclusión para poder implementar de manera correcta el robot y evitar ciertos problemas el autor recalcó que lo hizo mediante la optimización del diseño. También hizo énfasis en la estructura modular que facilitó el reensamblaje y el diagnóstico visual rápido en caso de los fallos, Básicamente lo que se puede sacar de esto es que si se hace un diseño optimizado luego es mucho más fácil encontrar los fallos y solucionarlo más rápidos.

### 3.4.7. EUROBOT 2023 - THE CHERRY ON THE CAKE

#### 3.4.7.1. introducción

La competición de este año se hizo con el objetivo de que los robots organizarán fiestas de cumpleaños, para ello tendrán que servir los pasteles y las guindas sobrantes colocarlas en una mesa.

Sobre la mesa hay cuatro ingredientes bizcocho (marrón oscuro), crema (naranja), Helado (rosa) y guindas (rojas). El equipo que consiga dejar más pasteles y guindas es el equipo que consiga más puntos y por lo tanto el ganador. Los robots también pueden conseguir puntos si vuelven a su lugar de inicio y hacen una pequeña celebración.



*Ilustración 13: evento de competencia 2023*

En la competición el árbitro se encargará de dar un clic A la zona de la casilla correspondiente del tablero, La puntuación final se calculará en base a estos clics y dependiendo de la zona y del tipo de ficha que hay en el tablero. La normativa de este año indica que si se juntaba el mayor número de fichas en un menor tiempo se obtenía más puntuación (Balanta Narváez, 2023, p. 5-7).

### 3.4.7.2. Análisis del robot

El sistema de arranque se hará mediante una cuerda que actúa como un empujador precargado con goma elástica, para el sistema de arranque también se puso un interruptor, el cual se debe de activar primero y junto con la cuerda se activa el robot (Balanta Narváez, 2023, p. 92).

El sistema de alimentación estaba compuesto por una batería BMS y pulsador de emergencia. El sistema de dispensado de guindas este compuesto por un driver para controlar el motor paso a paso, para realizar un desplazamiento vertical (ascensor). También tiene un sensor de origen para saber la posición del motor (Balanta Narváez, 2023, p. 92).

El sistema de locomoción estaba compuesto por dos motores de locomoción, que dispondrá de un encoder y reductora; este sistema moverá el robot por el tablero (Balanta Narváez, 2023, p. 92).

El sistema para coger y dejar tartas: compuesto por un servomotor unido a una pinza, habrá dos láseres para guiar el robot. Dispondrá de un sensor para detectar el número de capas para cambiar de altura (Balanta Narváez, 2023, p. 92).

Para el sistema de localización y navegación se usará una cámara en la torre y conectada mediante wifi. El microcontrolador recibe información de los sensores y la envía a los actuadores (Balanta Narváez, 2023, p. 92).

En conclusión en este proyecto de fin de grado se ha hecho un robot de competición autónomo, que detecta las tartas y las deja en un determinado lugar, para ello se han usado sensores, controladores, drivers... se hizo un diseño optimizado para que fuera sencillo y evitar problemas durante la fabricación; de aquí se puede concluir el aprendizaje del funcionamiento de manera colectiva de los componentes y la importancia individual de cada uno.

## 3.4.8. INTEGRACIÓN DE VISIÓN ARTIFICIAL EN UN ROBOT INDUSTRIAL

### 3.4.8.1. Introducción

El trabajo de visión artificial nos puede resultar interesante para la competición de eurobots ya que detecta unos objetos específicos en una zona y nos puede ayudar a diferenciar objetos para manipular e interactuar con los mismos. El Trabajo de Fin de Grado de Lucía Díaz García (2018) aborda precisamente este desafío, desarrollando un sistema que combina una cámara de profundidad Intel RealSense D435 con un sistema de manipulación basado en una pinza industrial. (Díaz García, 2018, p. 7).

Aunque está orientado al sector industrial, nos viene a nosotros interesantes ya que toca temas como la detención de objetos por forma, por su color, y también por el eje cartesiano tridimensional del objeto es decir su posición. Se pueden usar perfectamente en el contexto competitivo de Eurobot, donde los robots deben detectar, clasificar y mover objetos con precisión bajo restricciones de tiempo y espacio. Este TFG ofrece una base tecnológica aplicable al diseño de robots autónomos que deban tomar decisiones en tiempo real basadas en datos visuales, lo que lo convierte en una referencia valiosa para el desarrollo de estrategias de navegación y manipulación en el marco de esta competición.

### 3.4.8.2. Análisis del robot

Para el sistema de navegación se usó una cámara RealSense Depth Camera D435, para poder detectar los objetos en función de sus características, esto es lo que detecta el objeto por su color, forma y el eje de coordenadas cartesianas. Cuenta con dos sensores de profundidad, un sensor RGB y un proyector de infrarrojos (Díaz García, 2018, p. 22-24).



*Ilustración 14: cámara navegación*

En cuanto al sistema de manipulación se usó una pinza PGM 60 de la marca Schunk, la cual tiene en sus extremos una especie de 2 dedos con la cual se pueden agarrar los objetos; el control de estas pinzas se realiza mediante señales digitales, con dos estados (abierto o cerrado), Como la pinza tiene dos estados al final se hacía un sistema bastante

sencillo de controlar (Díaz García, 2018, p. 21)



*Ilustración 15: pinza sijeccion*

y por último mediante la programación y el comportamiento del brazo del robot se debe determinar la posición en la que va a terminar el brazo. Teniendo en cuenta la información dada por el vector de 6 datos (Díaz García, 2018, pp. 27–29).

En conclusión, de este proyecto podemos extraer materiales e información relevante para nuestro sistema de manipulación y navegación, ya que usa unas pinzas para coger unas piezas que están desordenadas, dichas pinzas funcionan mediante señales digitales de dos estados, lo cual hace sencilla su programación e implementación. Por otra parte tenemos el sistema de navegación, donde usa una cámara D435, la cual es bastante completa; ya que nos indica la profundidad, detección de objetos y por colores, la cual puede ser buena opción para nuestro sistema de navegación

## 3.4.9. ROBOT SEGUIDOR

### 3.4.9.1. Introducción

Desarrolló un robot móvil autónomo en donde el robot se tenía que desplazar en base a una línea que estaba puesto en la zona donde iba a circular el robot. Este trabajo de fin de grado nos puede servir bastante aunque no tenga que ver con eurobot, porque usa sensores por ejemplo para detectar las líneas un sensor de infrarrojos que me puede servir sin problema, sobre todo porque nuestro robot tiene que detectar un objeto y orientarse hacia él y acercarse hacia él; entonces podemos usar esta metodología para centrar nuestro robot hacia la pieza que tenemos que coger (Gutiérrez Jiménez, 2020, p. 2-4)

Usando sensores para la detección de líneas de infrarrojos, sensores ultrasónicos para detección de obstáculos, y microcontroladores con múltiples periféricos integrados, pueden ser reutilizados o adaptados para las fases de navegación, localización y detección de objetos en Eurobot. Es un tfg muy interesante que me puede servir bastante.

### 3.4.9.2. Análisis del robot

Cogió 5 pilas alcalinas que tiene autonomía de hasta 1 hora; este tipo de batería son de menor capacidad que las de litio y más pesadas, pero como este proyecto es de pruebas, eligió pilas alcalinas debido a que en el corto plazo son más barato (Gutiérrez Jiménez, 2020, p. 10)

Para el sistema de control se eligió un microcontrolador TM4C1294NCPDT de Texas Instrument. Se optó por este tipo de microcontrolador ya que tiene gran cantidad de periféricos (Gutiérrez Jiménez, 2020, p. 11)

Para el sistema de navegación escogió un sensor de línea QTR-8RC, el cual detecta objetos de colores claros y de colores oscuros. Y un sensor de distancia HC-SR04, elegir este componente tiene inconvenientes como el tiempo de medida y la precisión para medir distancia entre objetos, que depende de factores ambientales; pero se optó por este sensor ya que solo se necesita detectar presencia de objetos, y no la distancia precisa entre ellos (Gutiérrez Jiménez, 2020, pp. 12-13).

En cuanto al sistema de locomoción se eligieron dos motores de corriente continua DG01G-L, con reductora 48:1. A la salida de la reductora va una rueda 65x25mm. También eligió dos encoders, el cual está conformado por un disco con 12 periféricos, este encoder mide con precisión la velocidad y trayectoria (Gutiérrez Jiménez, 2020, pp. 13-14).

Para la adaptación de potencia y el control de motores se escogió un integrado de L298N, este se encuentra en una placa propia e incluye un regulador de 5V (Gutiérrez Jiménez, 2020, p. 14-15).

La velocidad de los motores se encontraba mediante el PWM, al variar el ciclo de trabajo hacía que los motores aumentarían o disminuirán su velocidad

En conclusión de este proyecto se pueden extraer materiales e información relevante para nuestro proyecto, como usar pilas alcalinas en lugar de las de lipo, ya que el eurobot no durará demasiado, el inconveniente sería el peso que agregaría al robot. Para el sistema de navegación se escogió sensor QTR-8RC para detectar color y un sensor de distancia (ultrasonidos) HC-SR04, que aunque no sea preciso para detectar la distancia entre objetos nos sirve, ya que nos consigue con facilidad una medida entre el sensor y el objeto, y se maneja con dos pines digitales. Para el sistema de locomoción se escogió un sistema de dos ruedas con dos motores y encoders en cada rueda, es un sistema de triciclo, y este sistema nos sirve para nuestro robot

## 3.4.10. Desarrollo de un sistema de control de un robot móvil

### 3.4.10.1. Introducción

El Trabajo de Fin de Grado titulado *Desarrollo de un sistema de control de un robot móvil*, presentado por Carlos Jiménez Jiménez en la Universidad de Valladolid (2017), describe el desarrollo completo de un robot autónomo para tareas de navegación y percepción del entorno. Me sirve a mí ya que es un robot autónomo, es decir, que va solo, navega por el entorno y detecta obstáculos, entonces podemos extraer información valiosa de aquí.

### 3.4.10.2. Análisis del robot

está diseñado con un chasis de aluminio, el aluminio es un material que es interesante ya que es fácil trabajar con él si queremos darle una determinada forma o algo (ventaja en resistencia, ligereza y facilidad de mecanizado). El formato hexagonal es una forma buena porque nos facilita la colocación de sensores en múltiples ángulos, algo especialmente útil para robots que requieren detección perimetral continua (Jiménez Jiménez, 2017, p. 14).

Trabajo fin grado

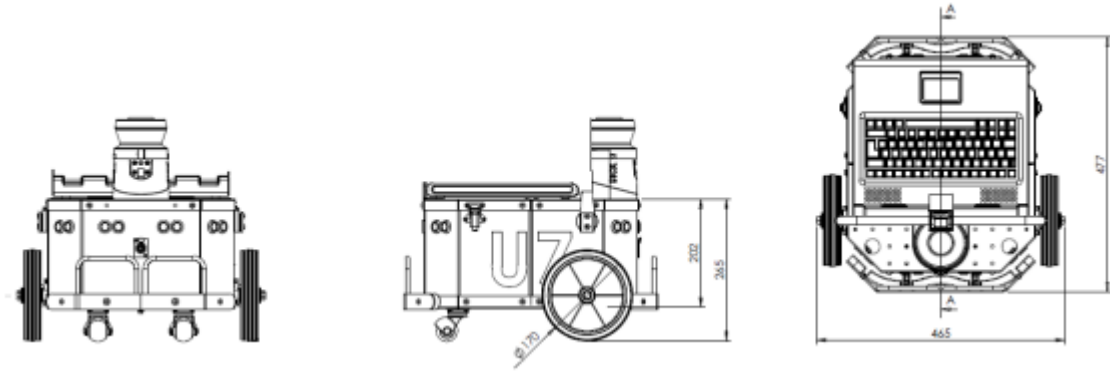


Ilustración 16: Chasis completo

Cómo se pueden ver en las imágenes usa un sistema de tracción diferencial en donde usó dos ruedas motrices con dos ruedas de apoyo para una buena estabilidad. Este tipo de tracción, combinada con motores DC IG-42GM de 12V, que tienen dos encoders ópticos, proporciona un control preciso de velocidad y dirección. En particular, los encoders permiten implementar técnicas de odometría y control PID, y eso nos viene bien ya que con eso podemos saber la posición en la que está el robot mediante pulsos que ha dado y también podemos controlar la posición (Jiménez Jiménez, 2017, p. 23).

Encontrar sistema de navegación el robot incorpora varios sensores: sensores ultrasónicos para evitar colisiones, sensores de contacto tipo bumper, cámara para tareas de visión artificial, y un escáner láser Hokuyo, fundamental para mapeo y localización. Gracias a esta configuración, el robot puede implementar algoritmos de SLAM y mantener una localización continua, técnica ampliamente aplicada también en Eurobot para navegación autónoma y evasión de obstáculos (Jiménez Jiménez, 2017, pp. 27–31).



Figura 27 Hokuyo



Figura 26 Cámara de la Raspberry

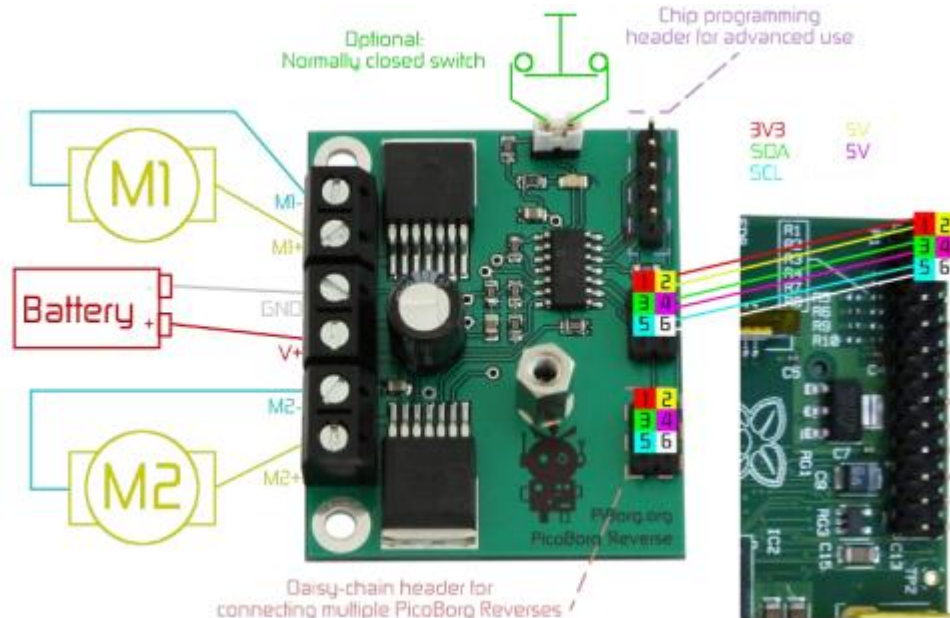


Figura 21 Ultrasonidos

Ilustración 17: Sensores de navegación

**Trabajo fin de grado**

En cuanto a la unidad de control se adoptó por: el control de bajo nivel es gestionado por una placa PicoBorg que controla los actuadores, mientras que el control de alto nivel lo realiza una Raspberry Pi 3 que ejecuta ROS (Robot Operating System). Esta separación permite modularidad en el sistema, Reduce la latencia en la ejecución de tareas y habilita algoritmos para la ejecución de planificación (Jiménez Jiménez, 2017, pp. 31–37).



*Ilustración 18: Pico Board*

La navegación autónoma fue implementada a través del stack de navegación, integrando sensores, odometría y control PID para generar trayectorias seguras y adaptables al entorno. Esto es totalmente aplicable eurobot dónde se requieren movimientos autónomos y detectar objetos de manera precisa (Jiménez Jiménez, 2017, pp. 66–74).

En conclusión, este antecedente proporciona una base sólida para entender cómo integrar de forma eficiente sensores, actuadores y controladores dentro de una estructura robótica modular, ligera y funcional, en condiciones que exigen autonomía, precisión y versatilidad. Muchas de sus soluciones son aplicables o adaptables al entorno técnico y reglamentario de la competición Eurobot.

## 3.5. Marco Teórico

### 3.5.7. Cinemática diferencial.

#### 3.5.7.1. Cinemática diferencial

La cinemática diferencial es el estudio de cómo las velocidades y aceleraciones de las articulaciones y enlaces del robot se relacionan con las velocidades y aceleraciones del efecto final del robot. La cinemática diferencial es importante para diseñar e implementar sistemas de control de retroalimentación que regulan el movimiento del robot, así como para optimizar el rendimiento y la eficiencia del robot. Para realizar la cinemática diferencial, es necesario derivar la matriz jacobiana, que es una aproximación lineal de la relación entre las velocidades de las articulaciones y las velocidades del efector final. La matriz jacobiana también se puede utilizar para analizar la singularidad, la destreza y la manipulabilidad del robot.

#### 3.5.7.2. Tipos de modelos cinemáticos

Hoy por hoy existen varios sistemas de locomoción como sistema de orugas, ruedas convencionales y sistemas de patas. en vehículos de ruedas convencionales pueden tener 2,3 o 4 ruedas

#### 3.5.7.3. Tracción diferencial

Es un sistema de tracción que permite que un objeto se puede mover hacia atrás hacia adelante y quiere en su lugar mediante el control y dependiente de las ruedas o grupo de ruedas en cada lado de los objetos. Este tipo de sistema es muy simple y eficaz en una variedad de aplicaciones como en los robots móviles.

##### Principio de funcionamiento:

para explicar el funcionamiento hay que entender los tipos de movimiento que puede tener la creación diferencial para ello veremos los siguientes puntos:

- la tracción diferencial utiliza al menos 2 ruedas que están en cada lado del vehículo. En donde cada grupo de ruedas se conecta a un motor independiente el cual permite controlar la velocidad y la dirección de giro.

- Por otra parte esta el movimiento hacia adelante y atrás, que es cuando ambos motores al haber tenido la misma dirección y a una misma velocidad entonces se mueve hacia adelante; si gira en sentido

contrario el robot se mueve hacia atrás

- y luego tenemos otro movimiento que es el giro en el propio lugar del que se encuentra, esto se basa en que uno de los motores se detiene y otro continúa girando entonces esto hace que una de las ruedas no se mueva, Y eso termina en el giro del robot alrededor de su eje central

- Y en el último tipo de movimiento tenemos graduales, es decir, que no tiene un movimiento brusco sobre su propio eje, Si un motor gira más lento que otro tenderá a girar hacia el lado que el motor sea más lento

(Misiego Lopez, 2015, p. 20)

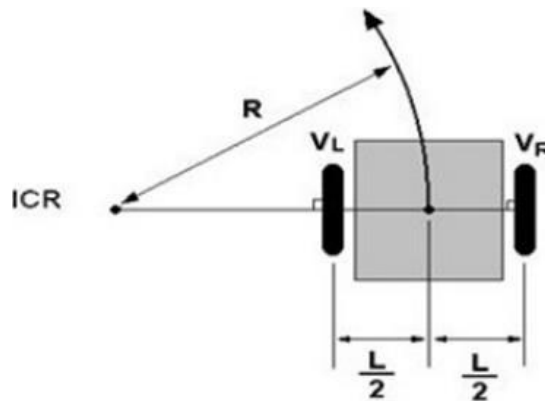


Ilustración 19: Tracción diferencial

### 3.5.7.4. Triciclo

en la tracción de un triciclo lo que nos indica es que disponen de 3 ruedas, en donde una actúa como directriz y esa es la responsable de facilitar el movimiento durante la maniobra, a continuación se muestran todas las posibilidades que hay

3.5.7.4.1. La primera opción es que la rueda trasera actúa como función de motriz y las ruedas delanteras como dirección

3.5.7.4.2. en segundo lugar, tenemos que la rueda delantera desempeña el papel tanto de motriz como de dirección, mientras que la trasera permanece como rueda fija.

3.5.7.4.3. Y por último tenemos que las ruedas delanteras harían el papel tanto de motrices como de directrices y la rueda independiente trasera sería una rueda bola fija.

(Universitaria de Investigación y Desarrollo, 2009).

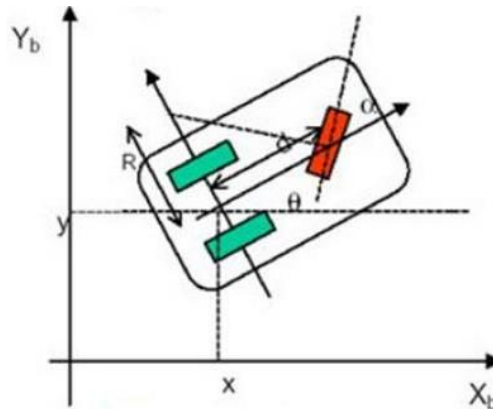


Ilustración 20: Representación triciclo

### 3.5.7.5. Cinemática de oruga

Este sistema tiene mayor contacto con el suelo, en lugar de ruedas utiliza cintas continuas que giran alrededor de un conjunto de ruedas y rodillos. Estas orugas distribuyen el peso del vehículo sobre una mayor superficie, lo que proporciona una mejor tracción y estabilidad en terrenos difíciles. El robot dispone en su cinta de más de 2 ruedas en cada lado en donde el giro de las ruedas no está permitido, entonces se debe llevar a cabo de una manera similar como se hace con la configuración de diferencial (Jiménez Jiménez, 2017, p. 16).

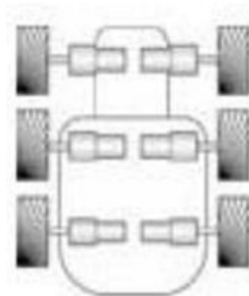


Ilustración 21: Metodo de orugas

### 3.5.7.6. Dirección Ackermann

Coordina mecánicamente el ángulo de 2 ruedas delanteras que están fijadas a un eje común utilizado para la dirección y dos ruedas para aceras fijadas en otro eje para la conducción.

Este enfoque se usa generalmente para robots rápidos al aire libre que requieren una excelente distancia al suelo y tracción.

Este es un sistema de varillas que hace que la rueda interior en una curva tenga un ángulo más pronunciado que la exterior; el resultado es que las dos ruedas describen circunferencias concéntricas al girar y de esta manera reduce el rozamiento y realizan giros más suaves

(Jiménez Jiménez, 2017, p. 17).

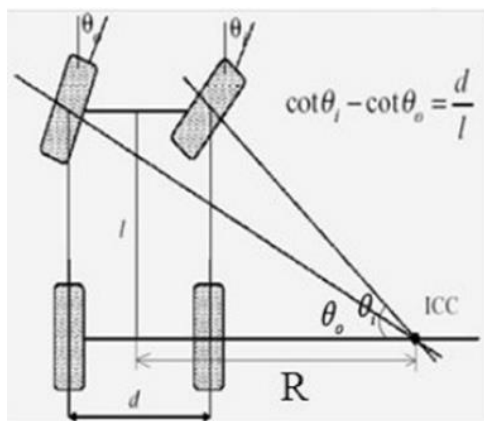


Ilustración 22: Ackermann

### 3.5.7.7. cinemática omnidireccional (ruedas Mecanum o esféricas)

Están fabricadas de ruedas especiales que se pueden mover en cualquier dirección. Este tipo de sistema es similar al ackerman, solo que en este caso el eje delantero no gira, tiene una plataforma rectangular y la ruedas se colocan en los lados paralelos que generalmente son lados más grandes (Jiménez Jiménez, 2017, p. 15).

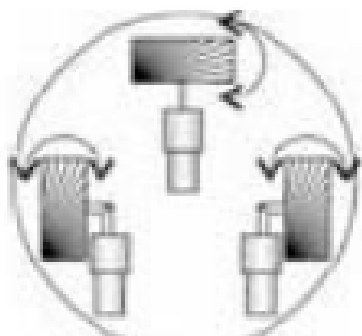


Ilustración 23: Representacion modelo omnidireccional

### 3.5.7.8. técnicas de navegación autónoma y sensores de entorno

Esta parte es fundamental ya que nos dice la posición de los robots en el terreno de juego, y también tenemos la parte de los sensores que detentan los obstáculos y los objetos que se deben de tomar, y el sitio exacto donde se deben de dejar para obtener los puntos. En donde debemos tener en cuenta lo siguiente:

### Componentes clave:

- En primer lugar tenemos los sensores, que en esta competición permiten emplear una variedad de sensores para percibir lo que pasa en el entorno y poder recibir información importante. Los sensores son las cámaras, lidar, encorder, sensores de ultrasonidos. En cuanto a la función de cada uno, las cámaras permiten la visión del robot, el lidar y los sensores de ultrasonido permiten saber la distancia y detectar los objetos, y los encorder son necesarios para saber la posición y el movimiento de las ruedas del robot
- Por otra parte tenemos los cálculos de trayectoria, en donde se debe realizar cálculos de las rutas para que el robot se mueva desde su posición actual hacia el objetivo que se desee entonces para ello se debe de usar un algoritmo adecuado, el cual realice la búsqueda y encuentre la ruta más corta considerando los obstáculos y restricciones de juegos.
- Otro punto esencial es evitar los obstáculos de esta manera se garantiza que en robot se desplaza de manera segura, y así se evita colisiones con objetos en tiempo real, para ello se usan algoritmos para que el robot a una cierta distancia evite los obstáculos ajustando su trayectoria.
- SLAM (Localización Simultánea y Mapeo): esta técnica permite crear un mapa del entorno mientras determina su posición dentro de ese mapa. Esto se logra combinando los datos de los sensores con algoritmos de localización en tiempo real. Los sensores como el lidar y el de ultrasonidos son fundamentales para la recopilación de datos del mapeo y los algoritmos permiten determinar su posición de manera rápida (Universidad de Zúrich, 2025).

### TECNOLOGÍAS MÁS RELEVANTES:

- Para planificar la ruta se usa un algoritmo tipo A\*, calcula la ruta más corta teniendo en cuenta obstáculos y restricciones
- Controladores PID: para seguimiento de manera precisa de las trayectorias, de esta forma el robot se mueve de manera suave y controlada hacia su objeto
- Las redes neuronales (CNN) y la visión por cámara permiten al robot reconocer objetos a partir de imágenes y videos.
- Sensores LiDAR: detección de obstáculos y creación de mapas tridimensionales de entorno para facilitar la navegación
- ROS (Robot Operating System): facilita la programación, comunicación y gestión de sensores en robots.

(Fernández León, 2023).

### 3.5.7.9. sistema de navegación

#### estimadores explícitos

En el sistema de los estimadores explícitos se pueden ver dos grupos: que es la estimación basada en las medidas internas y la estimación basada en la transmisión

La estimación que se basa en las medidas internas la manera más sencilla de poder estimar la posición y la orientación del robot consiste en poner las trayectorias del recorrido a partir de una serie de medidas internas: como las vueltas que se han dado, velocidades, aceleraciones, cambios de orientación, etc. Se pueden distinguir dos grupos fundamentales, sistemas por odimetría y sistema de navegación inercial

- Sistemas odimétricos: el objetivo principal es estimar la posición y la orientación del robot a partir del número de vueltas que se ha dado mediante las ruedas. En la actualidad para saber el número de vueltas que da una rueda se utilizan codificadores ópticos de mucha precisión montado sobre los propios ejes de la rueda. Obviamente para poder hacer esta estimación se tiene que realizar un registro odimétrico de al menos un par de ruedas del robot

Odiometría tiene una ventaja y es su simplicidad y bajo costo. Sin embargo aparte de necesitar una frecuencia de calibración debido al desgaste y pérdida de presión de las ruedas del ajuste de los ejes; esta técnica es vulnerable a imprecisiones causadas fundamentalmente por el deslizamiento o irregularidades del suelo. Otra cosa que afecta la estimación es de la carga transportada por el robot. Aunque es posible diseñar un modelo a partir del peso para que distribuya la carga y permita corregir desviaciones introducidas.

- Sistema de Navegación Inercial: Los Sistemas de Navegación Inercial (INS) estiman la posición y orientación del robot usando medidas de las aceleraciones y ángulos de orientación. La primera integración de las aceleraciones proporciona la velocidad y la segunda la posición.

A diferencia de la odimetría la navegación inercial no se ve afectada por los problemas de interacción del robot con el suelo, además que pueden corregir los efectos de las ondulaciones e irregularidades del suelo. Por lo tanto son mucho más fiables y precisos que los sistemas odimétricos pero lo malo es que son más frágiles y son mucho más caros de implementar. Por lo tanto si tienes que hacer un proyecto sencillo lo más fácil es usar la odimetría

**Trabajo fin grado**

La estimación basada en estación de transmisión, este tipo de sistema de posicionamiento absoluto ha venido siendo muy usado desde hace varias décadas en aplicaciones marítimas y aeronáuticas. También se usa mucho en la robótica por ejemplo en los robots móviles para exteriores y en particular en vehículos de navegación autónoma. La principal ventaja de esta técnica es que proporciona la localización absoluta del vehículo en un área suficientemente grande y sin requerir estructuración alguna del entorno, esto hace especialmente adecuada la utilización para aplicaciones donde por lo general el vehículo ha de moverse por zonas muy diversas y variantes y recorriendo grandes distancias (carreteras, caminos forestales, campo a través, parajes semidesérticos, etc.) (Lima & Ribeiro, 2022).

estimadores basados en la percepción del entorno

- Otra alternativa a los anteriores métodos consiste en la estimación de posición que lo que hace básicamente es poner en el robot móvil un sistema sensorial que es capaz de proporcionar suficiente información del entorno para que pueda saber su localización. Este sistema puede trabajar en base a distinto tipo de sensores (cámaras CCD, sónares, escáner láser, etc.) Y seleccionando determinados tipos de datos u objetos a partir de una información adquirida (marcas naturales, puntos de interés, entorno completo percibido, etc.). De igual manera la localización del vehículo se determina a partir de los datos extraídos del entorno por el sistema sensorial, y se compara con datos que previamente se conocían de éste. Esta información previa la podemos obtener de un modelo conocido del entorno, o de anteriores observaciones. Asimismo, este emparejamiento de datos se puede realizar según distintas estrategias.
- Sensores activos: Proporciona medida de la distancia del entorno lo cual lo hace para saber la posición del robot y a qué posición está de cada objeto. Los más utilizados son sin duda los sónares y los sensores láseres. El principio de funcionamiento más utilizado es el denominado "tiempo de vuelo", consistente en medir el tiempo transcurrido entre la emisión de una señal (pulso, onda modulada, etc.) y la recepción del eco correspondiente. A partir de esta medida y teniendo en cuenta la velocidad de propagación por ejemplo en el aire se puede estimar la distancia a la que se encuentra el objeto.
- Sensores pasivos: los sensores denominados pasivos se limitan a captar la energía ya existente en el medio. Por ejemplo, de este tipo son los sensores de infrarrojos, que examinan el entorno y si ha habido un cambio de luminosidad, hay sombra o hay algo ahí es cuando detecta el objeto. (Lima & Ribeiro, 2022).

## 3.5.8. Tecnología Lidar.

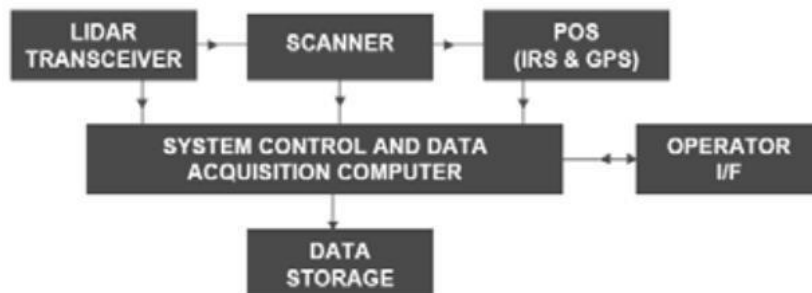
### 3.5.8.1. Definición

Un aspecto importante que debe tener nuestro robot es la capacidad de analizar las características del entorno, por ello la tecnología Lidar (Light Detection and Ranging) es una opción muy conveniente para lograr este fin, utilizando pulsos de luz láser sobre superficies y midiendo el tiempo que tardan en regresar permite generar una nube de puntos que describa las superficies del objeto.

Esta tecnología es útil en todos los sectores que precisen obtener medidas de la superficie como por ejemplo: Cartografía y topografía, automóviles autónomos, agricultura, conservación ambiental, Ingeniería civil, etc

### 3.5.8.2. Principio de funcionamiento

Esencialmente un sistema LIDAR está conformado por un sistema transmisor, un receptor, uno de control y uno de procesamiento, con el siguiente diagrama de bloques.



*Ilustración 24: funcionamiento general Lidar*

Los pasos del funcionamiento del LIDAR son los siguientes:

1. **Emisión de pulsos de láser:** Un dispositivo LIDAR emite pulsos de luz láser hacia un objeto o superficie en el entorno. El tipo de láser utilizado suele ser en el espectro infrarrojo con una longitud de 1064nm lo hace invisible al ojo humano.
2. **Reflexión del pulso:** Cuando la luz llega al objeto destino parte de su luz rebota y llega de nuevo al sensor LIDAR
3. **Detección del pulso reflejado:** Este sensor capta pulsos, que luego cuando llegan al receptor se devuelven entonces mide el tiempo que ha pasado entre ir al receptor y volver de vuelta y así sabe la distancia.
4. **Cálculo de la distancia:** Utilizando la velocidad de la luz, el sistema calcula la distancia entre el LIDAR y el objeto. Esta

distancia se determina mediante la fórmula:

$$\text{Distancia} = \frac{c * t}{2}$$

siendo:

- c la velocidad de la luz
- t es el tiempo desde que se emite el pulso por el transmisor hasta que es recibido por el receptor, es por ello que se divide entre 2 ya que calcula el tiempo de ida y de vuelta, suponiendo que son iguales.

5. **Generación de un modelo 3D:** Entonces el LIDAR envía un haz de luz en todas las direcciones y la señal cuando encuentra un obstáculo se devuelve. Entonces analiza cada punto por ejemplo si está en una habitación cada esquina dónde está cada pared y todo lo analiza y lo guarda y así hace una simulación 3D del entorno en el que está.

6. **Procesamiento de datos:** con los datos obtenidos realiza un mapa en 3 dimensiones, es decir un modelo tridimensional del entorno. Dependiendo de la aplicación, estos datos pueden ser utilizados para tareas como la navegación autónoma, la planificación de infraestructura o la detección de objetos.

(Salamanca Céspedes & Pérez Castillo, 2022).

## 3.5.9. Motores de corriente continua y reguladores correspondientes.

### 3.5.9.1. Definición

Los motores están hechos por un mecanismo que es capaz de transformar la energía ya sea eléctrica, de combustión... en energía mecánica con el objetivo de hacer funcionar un sistema o una maquinaria para que realice un trabajo. El proceso de transformación de energía se realiza mediante la variación de campos magnéticos también llamado electromagnetismo, su principio básico para este funcionamiento es la fuerza existente en la bobina que es donde circula la corriente y un imán.

### 3.5.9.2. Características

Existen muchos tipos de motores y cada uno tiene sus propias características, pero las características generales serían:

- **Estabilidad:** capacidad para mantenerse funcionando a altas velocidades sin consumir una cantidad excesiva de combustible o energía eléctrica.
- **Potencia:** es el trabajo que generan los motores en un tiempo a

Trabajo fin de grado

una velocidad determinada se expresa en caballos (CV)

- **Rendimiento:** es el resultado entre la potencia útil y la potencia absorbida.
- **Par motor:** momento de fuerza que se ejerce sobre el eje de los motores y es el que determina su giro.
- **Velocidad nominal:** es el número de revoluciones por minuto al que giran los motores.

TIPOS DE MOTORES DE CORRIENTE CONTINUA:

Los motores de corriente continua son dispositivos electromecánicos que transforman la energía eléctrica en energía mecánica (movimiento mecánico) mediante el principio de la acción electromagnética. Hay dos tipos de tecnologías esenciales en este tipo de motor que son los motores de corriente continua sin escobillas (brushless) y los motores de corriente continua con escobilla (Brushed).

### 3.5.9.3. Principio de funcionamiento

3.5.9.3.1. Ley de Ampere: Una corriente eléctrica genera un campo eléctrico.

3.5.9.3.2. Fuerza de Lorentz: Campo magnético que experimenta una fuerza perpendicular en la dirección de la corriente y campo magnético.

3.5.9.3.3. Ley de la fuerza electromotriz: que viene dada por la ley de Faraday mediante la ecuación  $E = k * w$

donde

$E$ =fuerza electromotriz

$k$ =constante

$w$ =velocidad angular

3.5.9.3.4. Generación de torque: es la interacción entre el campo magnético creado por la corriente en la bobina y el campo magnético externo el cual produce un par motor (torque).

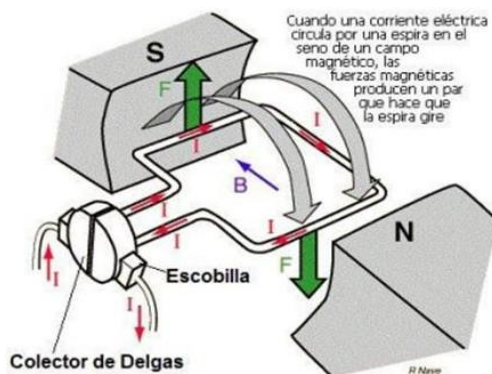


Ilustración 25: funcionamiento de un motor CC

### 3.5.9.4. Tipos de motores de corriente continua

1- Motores con escobillas (Brushed): este tipo de motores presentan una ventaja frente a los motores sin escobilla y es que son económicamente más viables, entre otras cosas, porque no necesitan de un control eléctrico para su funcionamiento. Pero también tienen inconvenientes ya que el ciclo de vida de estos motores es más limitado debido a la presencia de componentes clave, como las escobillas. En el caso de este tipo de motores al generar fricción durante el arranque y posteriormente cambiar su polaridad experimentan una pérdida de capacidad por el calor generado y las chispas; esto hace que sí acelere su deterioro y requieren un mantenimiento periódico para su reemplazo, aunque este mantenimiento eleve los costos es fundamental para poder mantener el ciclo de vida del motor.

Estos motores ofrecen una relación favorable entre potencia y tamaño, esto hace que sea adecuado para aplicaciones que requieran mucha potencia en un espacio muy reducido. Un ejemplo de implementación serían los ventiladores del ordenador, sistemas robóticos y otras aplicaciones que requieran eficiencia en dimensiones reducidas (Pascual Vega, 2019, p. 27).

Componentes: carcasa, piñón de ataque, rotor, estator y colectores

2- motores con (Brushless): utilizan un control eléctrico para cambiar la dirección de la corriente en el estator, por eso los variadores empleados en este tipo de motores son más complicados ya que se debe analizar la respuesta y los datos de funcionamiento del motor en tiempo real. Al no tener escobillas requieren de menos mantenimiento y a su vez que sean más eficientes.

En lugar de usar escobillas este tipo de motores usan imanes permanentes, en donde el proceso de conmutación se lleva a cabo electrónicamente mediante los controladores (drivers), esto hace que tenga un rendimiento óptimo. De manera semejante a los servomotores estos motores cuentan con 3 cables de cobre que están enrollados, en donde su conexión se realiza de forma típica en estrella; aunque estos tipos de aplicaciones varían en función del tipo de operación y de las especificaciones del motor.

Internamente se pueden encontrar un conjunto de cables que están conectados en el estator, los cuales están rodeados por imanes permanentes dispuestos en un patrón específico que dependen del número de polos que se requiera.



*Ilustración 26: Esquema interno del Brushless*

(Pascual Vega, 2019, p. 29)

3.5.9.4.1. reguladores, control de posición, velocidad y dirección Control proporcional-integral-derivado (PID): Son ampliamente utilizados para controlar los motores de corriente continua, la salida del controlador se ajusta en función del error proporcional, la derivada del error y la integral del error

3.5.9.4.2. Lazo de control de velocidad: en el control de motores de corriente continua, sobre todo en aplicaciones de velocidad constante se usa un alto control de velocidad. Este lazo lo que hace es comparar la velocidad real del motor con la velocidad que se desea y ajusta la señal de control en consecuencia. La velocidad de un motor de corriente continua se controla variando la tensión o la corriente.

3.5.9.4.3. Control de posición: se implementa un lazo de control de posición que compara la posición real con la deseada

3.5.9.4.4. Control de dirección: en el caso de los motores de corriente continua con escobillas, la corriente a través de las escobillas controla la dirección del movimiento. En el caso de los motores sin escobillas se ajusta la secuencia de conmutación del controlador eléctrico de esta manera se cambia la dirección.

En resumen las aplicaciones comunes se encuentran en una amplia variedad de aplicaciones por ejemplo vehículos eléctricos, dispositivos eléctricos autónomos... Lo cual hace que sean capaces de un control preciso y se pueden implementar de manera eficiente en la robótica y automatización industrial.

(IGNACIO IRANZO, 2021).

### 3.5.9.5. Motor paso a paso

Dispositivo que mediante una serie de impulsos eléctricos realiza desplazamientos controlados angularmente de manera discreta, es

Autor: Julio Cesar Balanta Narvaez

- 38 -

decir, que es capaz de girar un determinado número de grados en concreto dependiendo de sus entradas de control será un desplazamiento de paso o de medio paso. Este motor es controlado mediante impulsos que son generados por sistemas digitales actuando de manera similar a un conversor digital.

Los motores paso a paso comparten algunas similitudes con los motores de corriente continua, de hecho, son un tipo de motor de corriente continua. Tienen semejanzas con los motores de corriente continua, pero tienen una estructura y un funcionamiento distinto que los hace únicos.

En vez de depender de la fuerza electromotriz que es inducida por la rotación lo que genera el movimiento, los motores avanzan en pasos discretos o ángulos predefinidos, este movimiento se logra por la activación secuencial de las bobinas del motor.

Su funcionamiento para que se citen las bobinas se tiene que controlar de manera externa mediante un controlador (driver).

(Malo Yagüe, 2011, p. 12-13).



*Ilustración 27: Motores motrices*

### 1. principio de funcionamiento

- Disponen de un rotor y un estator de dientes magnéticos, El rotor se mueve en pasos discretos en donde cada paso se corresponde con una posición angular específica

- La energización secuencial de las bobinas del estator inducen campos magnéticos que atraen el rotor hacia posiciones predeterminadas; la secuencia más común son full-step y half-step.

### 2. tipos de motores paso a paso

Existen 3 tipos de motores paso a paso:

- Motor de reluctancia variable: posee un rotor multipolar de hierro y un estator devanado, el cual puede ser laminado, el motor rota cuando los dientes más cercanos al rotor son atraídos a las bobinas del estator.

- Motor de imanes permanentes: permite un par diferente de cero cuando el motor no está energizado y dependiendo de la

**Trabajo fin de grado**

construcción se pueden obtener diferentes pasos angulares. El ángulo de rotación del motor viene determinado por el número de polos que contiene el estator

- motor híbrido: es una combinación de las dos tecnologías anteriores, y se caracteriza por tener dientes en el estator. El rotor se compone por dos tapones de acero dentados con 50 dientes que se magnetizan y un estator de 48 dientes por lo que nunca llegan a estar todos los dientes alineados con la energización de las bobinas.

### 3. Reguladores y aplicaciones

- control de posición: controla la posición de un motor basándose en una entrada específica utiliza señal de retroalimentación para corregir el movimiento y asegurar que el motor paso a paso está en la posición requerida. En este tipo de motores implementa el microstepping, que lo que hace es dividir en fracciones más pequeñas un paso lo cual hace que se mejore la resolución, por tanto los controladores deben ser capaces de generar se puede ser de microsteps.

- Control de velocidad: la velocidad se controla ajustando la frecuencia la cual viene dada por la fórmula  $\omega = 2\pi \cdot f \cdot N$ . donde N es el número de pasos por revolución

(Malo Yagüe, 2011, p. 15-17).

## 3.5.9.6. Servomotores

Un servomotor es un motor diseñado para controlar movimientos lineales o rotativos, emplean mecanismos de retroalimentación para una posición exacta y un movimiento suave.

son controlados por pulsos eléctricos (pwm), si se controla por medio de cables tiene un pulso máximo y un mínimo, y una frecuencia de repetición

### 1. principio de funcionamiento

- señal de control: es el punto de partida, se trata de una señal eléctrica (pwm) enviada por controlador, que especifica posición o movimientos deseados
- motor y mecanismo de engranaje: cuando recibe la señal acciona el motor para que se mueva a la posición deseada.
- sistema de retroalimentación: como encoders, supervisa la posición actual del eje de salida del motor, compara esta posición con la deseada.
- Corrección de errores: compara la posición actual con la posición deseada a partir de la señal de control. Sí hay un error el controlador ajusta la potencia de salida al motor corrigiendo la posición

- controladores de posición y velocidad: controlador PID, la mayoría

### Trabajo fin grado

de los servomotores los usa para ajustar la posición y mantenerla estable, los servomotores utilizan una retroalimentación para mantener la posición, y como el PID ajusta la corriente pues ajusta la posición y por tanto el error entre la posición deseada y la real.

(Morales Ceballos & López Morales, 2014, p. 34-43)

#### 2. componentes

- Motor (CC o CA): para aplicaciones de mayor velocidad y potencia, se usa para un control más preciso
- Circuito de control: procesa la señal entrante y emite órdenes. determina velocidad, posición y par en función de las entradas recibidas.
- potenciómetro: proporciona información sobre la posición del eje del motor.
- eje del motor: transmite la potencia mecánica del motor a la aplicación.
- Servo accionamiento: suministra potencia necesaria para el motor para lograr el movimiento deseado, ajustando la tensión y la corriente.
- engranajes: amplificar el par de salida, para movimientos más precisos.
- servoamplificador: refuerza señales de control para asegurar que sean fuertes.
- encoders: proporciona información sobre la posición o velocidad al circuito de control.
- sistema de refrigeración: para evitar sobrecalentamiento (Morales Ceballos & López Morales, 2014, p. 27-36)

#### 3. características y aplicaciones

- precisión y resolución: alta precisión en la posición y control angular, lo que los hace ideales para aplicaciones de control fino.
- velocidad y par ajustable: mediante el control de la corriente suministrada al motor
- amplia gama de aplicaciones: robótica, automatización industrial, posicionamiento y control de movimiento
- retroalimentación continua: respuesta rápida a cambios en la carga o condiciones del entorno.

(Morales Ceballos & López Morales, 2014, p.18-19)

#### 4. consideraciones en controladores

- filtros y suavización de señales: se implementan filtros para

**Trabajo fin de grado**

- evitar oscilaciones y mejorar estabilidad del sistema, esencial para aplicaciones de baja velocidad o alta precisión
- adaptación a cargas variables: los controladores tienen que adaptarse a cambios en la carga del motor, ajustando la corriente suministrada
  - interfaz de comunicación: permite configuración remota y monitoreo del sistema
  - protección y detección de fallos: garantizar la seguridad y la integridad del sistema
  - optimización energética: mediante el control de energía, como la regeneración de frenado, elegir y configurar bien los controladores para el rendimiento óptimo.

### 3.5.10. Otros modelos cinemáticos

A Continuación se explorarán diferentes alternativas de modelos cinemáticos que se pueden aplicar al robot.

cinemática omnidireccional (ruedas Mecanum o esféricas):

ventajas: se puede mover en cualquier dirección sin necesidad de girar, permite rotación y traslación simultáneamente, permiten maniobras complejas en entornos complicados

desventajas: costo elevado, más complejas a la hora de controlar y programar, y menor eficiencia energética por el deslizamiento lateral

- cinemática tracción de orugas:

ventajas: proporciona mayor estabilidad y tracción, mayor durabilidad y movimiento en terrenos difíciles, y capacidad de realizar giros en el lugar.

desventajas: baja eficiencia energética, menor precisión en el control de movimientos y mantenimiento más complicado

- cinemática Ackermann:

ventajas: reduce el desgaste, es mucho mejor para mover en la curva y da una respuesta más precisa en la dirección

desventajas: no puede girar sobre su eje y mayor complejidad mecánica

- tracción diferencial:

ventajas: Es simple ya solo se necesitan dos motores para funcionar. Tiene mucha maniobrabilidad ya que puede girar sobre su propio eje y encima solo consume la potencia necesaria en aquellos motores que se necesitan.

desventajas: Lo malo es que pierde mucha tracción en superficies

irregulares o resbaladizas, por eso hay que ajustar bien la rueda y ver qué tipo de rueda y qué tipo de material coger para cierto tipo de suelo. (Jiménez Jiménez, 2017).

### 3.5.11. Sensores de entorno

A Continuación se explorarán diferentes alternativas de modelos cinemáticos que se pueden aplicar al robot.

- sensor LIDAR:

ventajas: preciso en la detección de distancias y obstáculos, Es ideal para hacer un mapeo tridimensional, de una zona en concreto con las distancias precisas y funciona con poca luz

desventajas: es caro y requiere un procesamiento más elevado.

- sensor de ultrasonidos:

ventajas: es más barata, detecta muy bien obstáculos cercanos y funciona muy bien en ambientes de mucha luminosidad.

desventajas: En menos precisos que el LIDAR tiene menos alcance y no es capaz de medir la distancia que hay entre objetos, tampoco reconoce un objeto de manera directa.

- sensor infrarrojo:

ventajas: Es barato ideal para proyectos que no requieren de mucha complejidad.

desventajas: solo detectan objetos cercanos y es sensible a la luz ambiental.

- SLAM:

ventajas: Combina diferentes sensores con lo cual sirve para analizar el entorno de una manera precisa y desde varias funciones diferentes, es optimo para un proyecto completo.

desventajas: es muy complejo ya que requiere un procesamiento muy alto

(Panigrahi & Bisoy, 2021)

### 3.5.12. Técnicas de navegación

Esto es fundamental en el área de la robótica sobre todo cuando se va a hacer un robot autónomo que debe tomar sus propias decisiones, debemos ver qué tipo de sensores le vamos a poner para que analicen el entorno.

- navegación odométrica: Estima en la posición mediante la distancia que ha recorrido el robot

ventajas: barata, sencilla de implementar y buena precisión

desventajas: lo malo es que si hay deslizamientos o irregularidades no lo detecta ya que solo detecta si la rueda gira, o sea detecta datos de pulsos.

- visión artificial:

ventajas: La principal ventaja que tiene sobre lo demás sensores es que puede detectar un objeto específico mediante una forma, un color determinado y nos dice el eje coordenadas donde se encuentra.

desventajas: el procesamiento de imágenes a veces suele ser muy pesado y ocupa mucho espacio, y si no tiene muy buena resolución en la Cámara o si está a muchísima distancia no va a detectar bien el objeto.

- navegación GPS:

ventajas: fácilmente actualizable, tamaño reducido permitiendo fácil transporte, fácil intercambio de información de manera más rápida y precisa

desventajas: El inconveniente más grande que tiene es que no funcionan en el momento, y claro en una competición obviamente necesitas un funcionamiento al instante.

(Panigrahi & Bisoy, 2021)

## 4.2.7. Materiales y cálculos estructurales

A continuación se van a exponer los diferentes materiales con los que se puede construir el robot.

- Acero inoxidable (AISI 304):

Ventaja: sus ventajas principales son su rigidez y que tiene una gran capacidad de carga

Desventaja: los inconvenientes que tienen son que tiene un elevado peso y una enorme dificultad de fabricación

- Aluminio (6061-T6):

Ventaja: sus ventajas principales son su rigidez y su reducido peso en comparación con el acero.

Desventaja: el principal inconveniente que tiene es su dificultad de fabricación

- Madera (Pino):

Ventaja: bajo coste, reducido peso y adecuada rigidez

Desventaja: dificultad de fabricación

- Materiales de impresión 3D ABS GF:

Tiene una resistencia a la tracción de 78 MPa y modulo de 5.63 GPa.

Ventaja: excelente rentabilidad, fácil Unión, fácil de mecanizar, resistente a la radiación.

Desventaja: Menor resistencia al impacto, tiende a deformarse en condiciones altas de temperatura, superficie rugosa por las fibras.

Cálculos estructurales:

Acero inoxidable (AISI 304):

- Resistencia a la tracción 515 MPa aproximadamente
- Densidad:  $7.930 \text{ kg/m}^3$

Aluminio (6061-T6):

- Resistencia a la tracción 310 MPa aproximadamente
- Densidad  $2.7 \text{ kg/m}^3$

Madera (Pino):

- Resistencia a la tracción 40 MPa aproximadamente
- Densidad  $550 \text{ kg/m}^3$

### 3.5.13. Sistema de almacenamiento de energía

Para el sistema de almacenamiento de energía usaremos las baterías, que se pueden definir como dispositivos electroquímicos que almacenan energía en forma de electricidad y la liberan cuando sea necesario. Son cruciales para alimentar al robot ya que proporcionan energía portable. Para escoger la batería se han de considerar varios aspectos como la capacidad, el peso y la vida útil que tenga la batería.

Hay diferentes tipos de baterías que están hechas de distinto tipo de material para almacenar energía, dependiendo del tipo de material tienen unas determinadas capacidades y limitaciones, a continuación, voy a expresar las diferentes alternativas de baterías que hay:

- Batería de fosfato de hierro-litio:

Ventajas: la durabilidad tiene un ciclo de vida largo y da mucha seguridad, ya que permite descargar la batería hasta el 100% sin dañarla. El umbral de fuga térmica es de alrededor  $260 \text{ }^\circ\text{C}$  lo cual hacen seguras.

Desventaja: Tienen una energía específica muy baja y su rendimiento puede disminuir cuando hay temperaturas bajas.

- Baterías Óxido de litio y cobalto:

Ventaja: tiene una alta energía específica lo cual permite entregar energía durante largos periodos de tiempo siempre y cuando sean aplicaciones de carga baja.

Desventaja: Tienen una vida útil muy corta y encima el cobalto es muy caro.

- Baterías de óxido de magnesio de litio:

Ventaja: se carga rápidamente y ofrece una alta potencia específica esto quiere decir que entrega una corriente alta, también ofrece mejor estabilidad térmica entonces funciona a temperaturas elevadas

Desventaja: Tiene una corta duración de vida.

- Baterías Óxido de níquel manganeso cobre de litio:

ventajas: alta densidad energética y una vida útil larga a un precio más bajo

desventajas: tiene un voltaje ligeramente más bajo que las baterías basadas en el cobre.

- Baterías de óxido de níquel cobalto aluminio litio:

Ventaja: Alta energía y vida útil

Desventaja: son caras y algo menos seguras

- Baterías de Titanato de litio:

Ventaja: carga más rápida, amplio rango de funcionamiento en temperatura, larga vida útil y una gran seguridad por su estabilidad

Desventaja: tiene una baja densidad de energía y su precio es muy elevado

(sunFields, 2025).

## 4.2.10. Control de locomoción en Robots cuadrúpedos

### 1. Introducción

Este tipo de robot son inspirados en animales a cuatro patas está hecho para zonas irregulares en donde un robot con tracción diferencial le costaría moverse. Se pueden desplazar de manera más sencilla en escaleras, lugares empinados con holguras en los que los robots de ruedas no podrían ni siquiera acceder. Esto se debe a que su distribución mecánica es uniforme y les permiten distribuir el peso y así poder acceder a zonas que son complicadas.

Actualmente con los avances tecnológicos que hemos tenido se ha hecho posible mediante técnicas como el aprendizaje, tecnología con sensores, y actuadores; que los robots ya puedan empezar a caminar.(Seok et al., 2015, p. 1118).

### 2. Tipos de Marcha (Gaits)

La marcha se refiere al patrón cíclico de movimiento de las patas. Algunos de los más utilizados en robótica son:

Marcha estática: En esta el robot tiene 3 patas estables en el suelo, lo malo de esto es que es más lenta (ideal para entornos peligrosos).

Marcha dinámica: En esta hay dos patas en contacto con el suelo, el problema de

### Trabajo fin grado

esto es que requiere una mayor estabilidad pero lo bueno es que es más rápida (usada en robots como Spot de Boston Dynamics).

Trote: Dos patas opuestas (diagonalmente) se mueven simultáneamente. Aquí se requiere velocidad y estabilidad.

Galope: Las patas traseras son las que empujan. Es propio de robots ágiles como Cheetah del MIT.

Aquí cada marcha se define por el ciclo que es el tiempo que cada pata está en el suelo y la distancia que hay entre las patas (Seok et al., 2015, p. 1117-1129).

### 3. Control de Movimiento

Existen varios enfoques para controlar la locomoción de robots cuadrúpedos:

#### a) Mediante modelos dinámicos

Se emplean las ecuaciones de Newton-Euler o Lagrange para calcular la fuerza y la trayectoria que llevará en esas patas, lo malo de esto es que requiere de los valores exactos.

#### b) mediante las redes neuronales

Se entrenan para la adaptación del entorno, esto permite que aprendan cuáles son las condiciones del entorno y pueden operar sin problemas en ella. Por ejemplo, si se usa el aprendizaje permite que el robot pueda caminar en un entorno que no tiene por qué ser el mismo y puede ser cambiante.

#### c) Osciladores Centrales Generadores (CPG)

Generan patrones rítmicos de activación para así dar pulsos a las patas y que se pueda mover. Son de bajo costo y a la vez son fáciles de usar debido a que si se modifican ciertos parámetros (velocidad, frecuencia...) ya se puede modificar la forma de movimiento y flexibilidad.

#### d) Control híbrido

Es una mezcla de los dos métodos anteriores, utilizando CPGs que lo que hace es que genera patrones dinámicos y realiza el control así dando equilibrio en tiempo real.

(Ijspeert, 2008).

### 4. Sensores y Percepción

Se usan este tipo de componentes ya que se requiere una percepción muy precisa del entorno. Los sensores más comunes son:

IMU (Unidad de Medición Inercial): Estima orientación, aceleración y velocidad angular.

Encoders rotacionales: Miden los ángulos articulares de cada pata.

Fuerza y presión en patas: Detectan cuándo una pata está en contacto con el suelo y la fuerza ejercida.

LIDAR y cámaras: Permiten mapear el entorno para generar rutas seguras y evitar obstáculos.

(Ijspeert, 2008)

### 5. Casos de Estudio

#### **Raibo:**

El robot cuadrúpedo Raibo puede correr a 3.9 m/s, saltar obstáculos, subir escaleras y girar bruscamente, todo sin un mapa de su alrededor. Visualiza

imágenes en tiempo real para analizar el entorno y usa estas imágenes para mediante la red neuronal, que es de profundidad hacer un aprendizaje y de cómo interactuar con ese mismo entorno.

### **ANYmal:**

Desarrollado por ETH Zurich, Aquí usa algoritmos de aprendizaje y se apoya con controladores no lineales para tener una navegación mucho más efectiva en cualquier entorno ya sean lineales o subterráneos.  
(El país, 2025).

## 6. Ventajas y Desventajas

### Ventajas:

En superficie que son irregulares tiene buena estabilidad  
Puede superar obstáculos como los escalones, terrenos con mucha holgura  
estabilidad biomecánica en diferentes terrenos

### Desventajas:

lo malo es que tiene mucha complejidad mecánica  
Tiene demasiado grados de libertad lo cual hace que consuma mucha energía y necesita un sistema avanzado para el control

## 4.2.11. Visión Artificial y Procesamiento de Imágenes

### 1. ¿Qué es la Visión Artificial?

Se basa en el reconocimiento mediante el procesamiento de imágenes o vídeos. En donde básicamente usa un algoritmo para estudiar aquellas imágenes y así poder realizar un reconocimiento de los objetos ya sea por su forma o color (IBM, 2021).

### 2. Tipos de Sistemas de Visión Artificial

Dependiendo de la información que reciben se pueden clasificar las siguientes categorías:

- Visión 2D: capturan imágenes planas, donde usan un eje de coordenadas en dos dimensiones; esto se usa en donde la profundidad no es necesaria, y solo necesitamos saber la forma de un objeto o su color.
- Visión 3D: Emplea sensores que capturan información de profundidad, como cámaras estéreo o sensores LIDAR. Esto es fundamental si queremos hacer un sistema de manipulación completo, para saber el sitio exacto donde se encuentra un objeto y así poder manipularlo (Cognex, n.d.)

### 3. Técnicas de Detección de Objetos.

Se usa para que el robot identifique los objetos que se encuentran a su alrededor usando diferentes técnicas, como la combinación de cámaras con sensores:

- Segmentación de imágenes: Divide una parte de la imagen en donde en la parte de la imagen se encuentra el objeto que se quiere identificar y la analiza.
- Detección basada en características: Utiliza descriptores como SIFT, SURF u

ORB para identificar puntos clave en las imágenes.

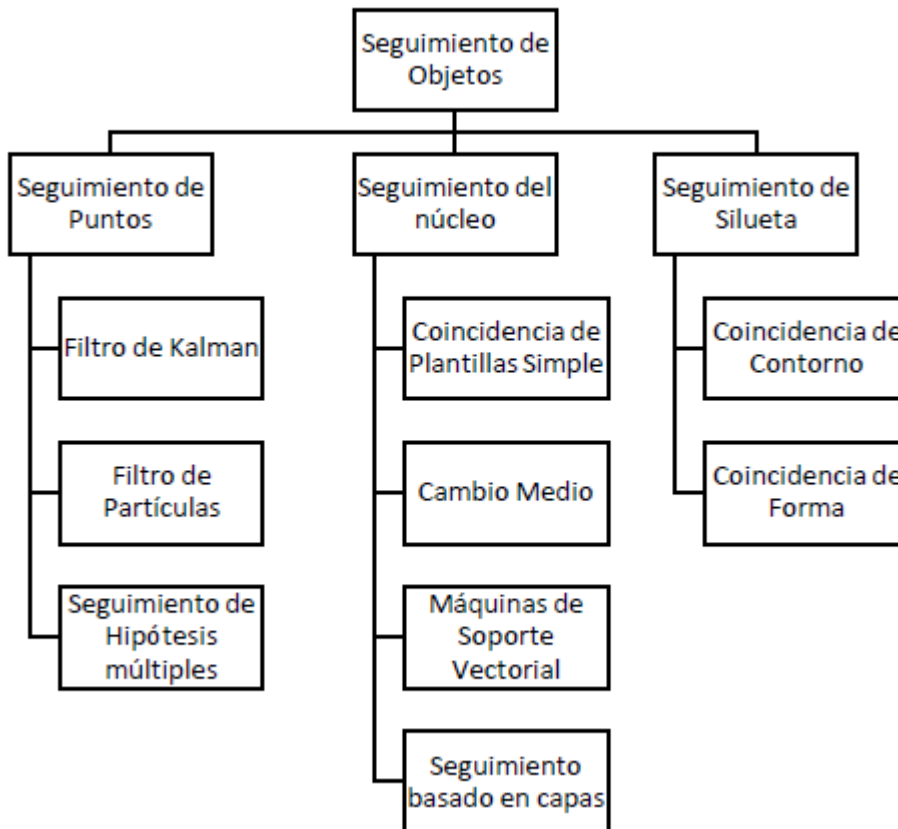


Ilustración 28: Métodos para el rastreo de objetos

- Redes neuronales convolucionales (CNNs): Aprendiendo a detectar objetos mediante un conjunto de datos, por ejemplo, segmentación de puntos que lo que permite es distinguir unos patrones de periféricos.

Una fórmula típica para distinguir un objeto mediante patrones, es el operador de Sobel:

$$G = \sqrt{G_x^2 + G_y^2}$$

#### 4. Seguimiento de Trayectorias

Esto permite monitorizar la trayectoria que lleva el robot permite monitorizar diferentes trayectorias o la de él mismo. Las técnicas más utilizadas son:

- Filtros de Kalman: Predecir la posición futura basándose en el estado actual y en los datos que se tengan anteriormente.
- Algoritmos de correlación: Comparan diferentes imágenes y consecutivas y así pueden predecir el siguiente movimiento
- Modelos de aprendizaje profundo: Como YOLOv8 combinado con ByteTrack, que permiten un seguimiento en tiempo real, este tipo de algoritmos se puede hacer por ejemplo por segmento de profundidad.

Una fórmula representativa del filtro de Kalman en una dimensión es:

$$x_k|_k = x_k|_{k-1} + K_k (z_k - Hx_k|_{k-1})$$

#### 5. Tipos de Trayectorias

cuando los robots están en movimiento la trayectoria se pueden clasificar en trayectorias simples o complejas:

- Lineales: trayectorias en línea recta
  - Curvilíneas: son trayectorias que pueden seguir una curva o una circunferencia
  - Poligonales: Son segmentos lineales que al final terminan cerrándose, es decir, es una combinación de segmentos lineales.
  - Trayectorias definidas por splines: utilizar funciones matemáticas para definir una trayectoria que pueda parecer compleja, de una manera más suave.
- Y para elegir el tipo de trayectoria pues depende de lo que queramos hacer, y de las condiciones del terreno.

## 6. Ventajas y Desventajas de la Visión Artificial

### Ventajas:

- Automatización: Los robots pueden realizar las tareas de forma autónoma sin que nosotros estemos controlándolos.
- Precisión: pueden detectar hasta el más mínimo detalle que para los humanos pueden ser imperceptibles.
- Velocidad: procesa una enorme cantidad de datos en tiempo real y los analizan de manera muy rápida.

### Desventajas:

- Dependencia de la iluminación: lo malo es que la luz puede afectar al procesamiento de imágenes ya que si tiene mala calidad no puede distinguir bien.
- Complejidad computacional: necesitan algoritmos muy complejos ya que para poder procesar toda esa información de manera precisa, requiere de mucha capacidad.
- Sensibilidad al entorno: si hay cambios en el entorno pueden afectar al rendimiento.

## 7. Aplicaciones en Eurobot

En Eurobot, la visión artificial se emplea para:

- Detección de objetos: Identificar un objeto específico ya sea por forma o color en la zona de juegos.
- Navegación autónoma: Permite que el robot se desplace de manera autónoma por el área de juegos evitando posibles obstáculos...
- Interacción con el entorno: Manipulación de objetos en el área de juegos, por ejemplo, agarrar un objeto por pinzas.

Implementar un sistema de visión artificial eficaz puede proporcionar una ventaja competitiva significativa en Eurobot.

(Santos, Dallos, Gaona, 2020).

## 4.2.12. Diseño y Fabricación de Estructuras Robóticas

### 1. Importancia del Diseño Mecánico

El diseño estructural es un componente esencial en el desarrollo de robots móviles, especialmente en el contexto de competiciones como Eurobot. Una estructura mecánica buena lo que permite es poder ubicar los sensores y los actuadores cómodamente sin tener que hacer ajustes incómodos, ello también permite que podamos redistribuir el peso de manera adecuada a lo largo de todo el robot, por ejemplo, sin ponerle peso de más a una rueda que a otra. Lo más

### Trabajo fin grado

cómodo es coger perfiles de aluminio, son muy ligeros o sea que no pesan mucho y encima tienen una capacidad de aguante bastante grande, soportan grandes pesos; y son bastante fáciles de manipular, es decir, si queremos hacer un agujero o cualquier cosa (Robotics Design Handbook, 2020).

#### 2. Selección de Materiales

La elección del tipo de materiales es fundamental en el robot, ya que en primera instancia le puede agregar más o menos peso y eso influiría en el tipo de motores que tendríamos que coger, por ejemplo, generalmente tienen que ser materiales que tengan bastante aguante y puedan soportar las colisiones. Los materiales óptimos suelen ser aluminio, plásticos para las pinzas o engranajes; y para los soportes podemos usar por ejemplo piezas de madera (Robotics Design Handbook, 2020).

#### 3. Criterios de Diseño

Cuando se diseña el robot se ha de considerar la rigidez, estabilidad, que tenga facilidad tanto para montarlo como desmontar el robot, y obviamente se ha de tener en cuenta también la normativa de la competición, por ejemplo, que el robot tiene que tener unas dimensiones específicas. También es importante prever pasos internos para el cableado y la protección de componentes electrónicos ante vibraciones o impactos (Robotics Design Handbook, 2020).

#### 4. Herramientas de Diseño y Simulación

Antes de empezar con la fabricación del robot lo primero que se ha de hacer es hacer una simulación en algún programa por ejemplo el inventor, en donde se fabrique el programa en tres dimensiones con sus medidas exactas y la medida de sus componentes también exactas; luego también se podría hacer una simulación con alguna herramienta para ver si los soportes y las plantas pueden soportar el peso de los materiales, sensores y actuadores; en nuestro caso era aluminio que puede soportar sin problema todo ese peso (Robotics Design Handbook, 2020).

#### 5. Mecanismos Comunes en Robots Eurobot

Bueno lo ideal es hacer un robot con el peso distribuido, y que más o menos si lo hacemos por plantas que cada planta tenga un determinado peso, se pueden sostener mediante barras de aluminio por ejemplo, luego también podemos usar conectores en forma de T y plataformas de móviles que puedan conectar una planta con la otra (Robotics Design Handbook, 2020).

#### 6. Aplicación en Eurobot

En eurobot bueno nosotros podemos diseñar las plantas mediante el aluminio por ejemplo, ya que es bastante resistente al peso entonces nosotros le podemos poner varios motores y los aguantará sin problema luego perfiles de aluminio para sostener las plantas; y si queremos obtener algo que no pese tanto pues podemos dar perfiles de madera, también podemos usar la madera para hacer algún soporte ya que no pesa mucho y tienen cierta resistencia. (Robotics Design Handbook, 2020).

## 4.2.13. Sistemas de Control y Electrónica

### 1. Introducción

Los sistemas de control y la electrónica son fundamentales en un robot móvil autónomo, ya que permiten la integración de sensores, actuadores y elementos

computacionales. En Eurobot, estos sistemas son responsables de:

- Detectar los datos sensoriales.
- Genera todas las acciones físicas como el movimiento y la manipulación.
- Coordina lo que viene a hacer el comportamiento autónomo.
- Gestiona toda la coordinación eléctrica del sistema.

## 2. Arquitectura Electrónica Típica

Una arquitectura electrónica modular típica incluye:

- Microcontroladores (MCU): Ej. Arduino Nano, ESP32 o STM32. Que ejecutan el control y toda la gestión de los periféricos.
- Microprocesadores: Como Raspberry Pi o Jetson Nano, estos son un poco más avanzados y se usan para las tareas como la visión artificial
- Drivers de motor: Ej. L298N, DRV8833, BTS7960, para controlar motores DC o paso a paso.
- Placas de expansión (shields): Permiten la conexión de los sensores y actuadores al microcontrolador
- Fuentes de alimentación: Baterías LiPo con reguladores buck o boost.

## 3. Diseño de Circuitos

Los circuitos deben cumplir criterios como:

- Tienen que tener bien aislada las señales de control y de potencia por ejemplo por obstaculizadores.
- Tienen que tener conectadas de manera correcta las masas y Unidas entre todos los circuitos, para que tengan una tierra común y pueda circular la corriente de manera correcta, evitando bucles de tierra.
- Luego también poner protecciones contra picos de tensión, como por ejemplo diodos para evitar dañar tanto el componente como el microcontrolador ya que actúan desviando la corriente excesiva.
- Diseño en PCB: Se puede usar KiCad.

## 4. Control de Bajo Nivel

Los sistemas de bajo nivel se encargan de mantener el comportamiento físico deseado. Algunos ejemplos:

### a) Control de velocidad con PID

Para regular la velocidad de los motores podemos usar el control PID:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int e(t)dt + K_d \cdot (de(t)/dt)$$

Donde:

$u(t)$ : señal de control (PWM).

$e(t)$ : error (velocidad deseada – velocidad real).

$K_p, K_i, K_d$ : constantes de ganancia.

### b) Control de posición

Similar al control de velocidad, pero basado en la distancia recorrida u orientación deseada (a menudo usando encoders).

## 5. Protocolos de Comunicación

Para conectar sensores y módulos:

- UART / Serial: Simple y común.
- I2C: Para sensores que no tengan una velocidad muy grande.
- SPI: Se usa para sensores que tengan una alta velocidad.

- CAN bus: Fiable en entornos con ruido eléctrico.

## 6. Aplicaciones

En Eurobot, los sistemas electrónicos deben ser:

- Lo más compacto posible y fáciles de acceder a ellos
  - Protecciones contra el cortocircuito por ejemplo poniéndoles fusibles.
  - Capaces de trabajar sin fallar por lo menos durante 90 segundos
  - En caso de que haya un fallo se tienen que poder parar.
- (IJSEM, 2024).

## 4.2.14. Estado del arte: Estudio de tecnologías y limitaciones

### 4.2.14.1. Sensores ultrasónicos

Este tipo de sensores utiliza el método *time-of-flight*, que lo que hace es emitir ondas que rebotan en el objeto y luego volver a regresar al sensor en un determinado tiempo

$$d = \frac{v_{\text{sonido}} * t}{2}$$

con  $v_{\text{sonido}} \approx 343 \text{ m/s}$  a  $20 \text{ }^\circ\text{C}$ . Tiene un rango de captación que puede ser bastante alto que es lo que nos hace falta, de 2 a 10 cm, para la competición que estamos haciendo es más que suficiente.

Limitaciones:

- tienen una resolución angular baja y el suelo a veces se puede detectar como un obstáculo ( $\sim 15^\circ$ ).
- Tiene demasiada sensibilidad a la temperatura, humedad y a la luminosidad.
- Si hay entornos irregulares puede haber mucho eco

(Wei, 2024)

### 4.2.14.2. Sensores infrarrojos

Los sensores IR miden la distancia mediante la intensidad reflejada de la luz infrarroja. En versiones ToF, calculan:

$$D = \frac{C * \Delta t}{2}$$

con  $c$  igual a la velocidad de la luz. Funcionan bien en interiores, pero son sensibles a:

- La luz ambiental, si hay poca luz ambiental no puede distinguir bien, también puede ser sensible a las texturas y el color
- por el contrario, si hay demasiado luz ambiental el sensor se puede ver saturado

**Trabajo fin de grado**

Para detectar objetos que son cercanos puede venir muy bien, y tiene la ventaja respecto al sensor de ultrasonido que el suelo no lo detecta como un obstáculo con lo cual no interfiere, pero tiene como contrapartida que no detecta la distancia hacia el objeto y que el rango de alcance es muy corto, suelen ser de más o menos 50 cm

### 4.2.14.3. Motores DC

Estos motores de corriente continua se modelan mediante dos ecuaciones mecánicas que describen tanto su comportamiento mecánico como eléctrico (Kuczmann, 2024):

Ecuación eléctrica:

$$R_a * i_a + L_a * \dot{i}_a + u_1 = u_a$$

Ecuación mecánica:

$$J\dot{\omega} = T - TL - k\omega$$

Y el par generado:

$$T = K i_a$$

Estas ecuaciones permiten representar el sistema en espacio de estados con:

$$\dot{i}_a = -\frac{R_a}{L_a} * i_a - \frac{K}{L_a} * \omega + \frac{1}{L_a} * u_a$$

$$\dot{\omega} = -\frac{K}{J} * i_a - \frac{k}{J} * \omega - \frac{1}{J} * TL$$

Y en forma matricial:

$$\frac{d}{dt} \begin{bmatrix} i_a \\ \omega \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K}{L_a} \\ \frac{K}{J} & -\frac{k}{J} \end{bmatrix} \begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} \\ 0 \end{bmatrix} u_a + \begin{bmatrix} 0 \\ -\frac{1}{J} \end{bmatrix} T_1$$

$$y = [0 \ 1] \begin{bmatrix} i_a \\ \omega \end{bmatrix} + 0u_a$$

Además, el modelo puede representarse en forma de espacio de estados, donde el vector de estado es:

$$X = [i_a \ \omega]^T$$

Y esto es lo que permite es diseñar controladores PID o también de retroalimentación de estados que se basan en los modelos reales obtenidos, donde también se implementan los filtros tipo SLAM que son filtros discretos que sirven para suavizar la señal de la corriente.

Limitaciones:

- Una de las partes negativas es que la inductancia en L y la propia inercia causan ralentización en la respuesta de control.
- La fricción y las cargas externas disminuyen la eficiencia del circuito en cuanto al movimiento y rendimiento.
- La linealidad del modelo no tiene en cuenta saturaciones ni comportamientos que no son lineales en los robots reales
- Si no se usa un filtro adecuado puede ser muy sensible al ruido.

(Kuczmann, 2024):

#### 4.2.14.1. Servomotores RC

Sus mecanismos son muy usados en la parte de la robótica sobre todo cuando se quiere trabajar con articulaciones o pinzas. Están compuestos por un motor DC, una caja de engranajes, reductores, un sensor de posición (potenciómetro o encoder) y una placa de control. Su funcionamiento se basa en el control del lazo cerrado, de esa manera pueden encontrar o alcanzar ángulos específicos (Autsou et al., 2024).

Los servomotores que suelen ser bastante conocidos son los de modulación por ancho de pulso (PWM). La posición angular se determina en función del ciclo de trabajo del pulso:

$$PW = D(PW_{max} - PW_{min}) + PW_{min}$$

Donde:

- $Pw$  es el ángulo de rotación,
- $pwmin$  y  $pwmax$  son límites angulares del servo,
- $D$  es ciclo de trabajo del pulso PWM.

En la práctica, los servos RC funcionan con pulsos de entre 1 ms ( $0^\circ$ ) y 2 ms ( $180^\circ$ ) en ciclos de 20 ms (frecuencia  $\approx 50$  Hz).

Limitaciones:

- Tiene una frecuencia de actualización bastante baja ( $\sim 50$  Hz).
- Si son aplicaciones críticas tiene baja presión angular eso es lo malo.
- El torque está limitado ( $\sim 1-2$  kg·cm).
- Pérdida de eficacia con carga excesiva o mala calibración del PWM.
- En nuestro caso por ejemplo si queremos agarrar un objeto no nos dice si lo ha agarrado ya que solo se mueve a una posición específica con lo cual no sabemos si la pinza agarrando el objeto.

En cuanto a este último punto esta última limitación, lo óptimo sería coger servomotores que te devuelvan la cantidad de corriente en función del

esfuerzo, con eso ya puedes saber si agarra un objeto.

#### 4.2.14.1. Control PID

La ley de un controlador PID es:

$$u(t) = Kp * e(t) + Ki \int_0^t e(T)dT + Kd * \dot{e}(t)$$

Limitaciones:

-Complicación a la hora de ajustar:

Ajustar un controlador PID puedes llegar a ser bastante complicado, sobre todo si lo vamos a hacer en sistemas con dinámicas que no son lineales o parámetros que cambian con el tiempo. Por eso es recomendable hacer pruebas para encontrar un balance adecuado entre la estabilidad y el rendimiento

-limitación en la robustez:

Los PID A veces no tienen por qué ser tan robusto si se usan como algoritmos más avanzados, y sobre todo si lo usamos en sistemas que tienen parámetros inciertos o alguna perturbación externa. Aunque existen otros métodos que permiten sintonizar de manera robusta, pero es que son mucho más complejos y requieren mucha experiencia.

-potencial para oscilaciones e inestabilidad:

Los parámetros obviamente se deben ajustar de manera correcta ya que si se ajusta mal el circuito es inestable y tiene lecturas erráticas, eso puede derivar en un funcionamiento incorrecto incluso que se dañe el sistema

-Saturación integral y el término derivativo sensible a errores de medición:

Problemas típicos en el PID es que la parte integral tiene errores acumulativos, es decir que tiene errores pequeños pero los acumulando los cuales se hacen grandes, y la parte derivativa es sensible a errores en la medición, eso hace inestable el circuito.

-No está muy claro cómo diseñar un sistema PID cuando el sistema no es SISO:

Eso quiere decir cuando el sistema tiene múltiples entradas y salidas entonces ahí la configuración del PID no es tan trivial (Vanneaux, 2024).

#### 4.2.14.1. Filtro de SLAM

Fase predicción:

$$x_k|_{k-1} = A_{k-1} * x_{k-1}|_{k-1} + B_{k-1} * u_{k-1}$$

Donde:

- $x_k|_{k-1}$  Estado que hay en el tiempo k hasta que llega al tiempo k-1.

- $A_{k-1}$ : Esta es la matriz que lo que hace básicamente es aplicar el movimiento a todo el sistema

$x_{k-1|k-1}$ : este es el estado estimado en el tiempo que es  $k-1$ .

$B_{k-1}$ : Matriz de control que mapea todas las entradas de control al estado.

$u_{k-1}$ : Esta parte determina cómo son las entradas de control.

(Thrun, S., Burgard, W., & Fox, D., 1999-2000)

Fase actualización:

$$x_{k|k} = x_{k|k-1} + K_k * (z_k - H_k * x_{k|k-1})$$

$x_{k|k}$ : Es el estado final estimado (corregido) en el tiempo  $k$ .

$K_k$ : Es la matriz de ganancia de Kalman, que determina cuánto "confiar" en la nueva medición frente a la predicción.

$z_k$ : Es la nueva medición recibida de los sensores (LIDAR y/o cámara) en el tiempo  $k$ .

$H_k$ : Es la matriz de observación, que proyecta el estado predicho al espacio de las mediciones esperadas.

$(z_k - H_k x_{k|k-1})$ : Es el residuo de la innovación o error de medición, la diferencia entre la medición real y la medición esperada.

(Thrun, S., Burgard, W., & Fox, D., 1999-2000)

Limitaciones:

-Sincronización y calibración precisa:

Cuando se sincroniza la cámara con el LIDAR, obviamente requiere un espacio temporal mientras limpia la información uno del otro, entonces claro si no se configura bien esto puede traducirse en que de mensajes o datos erróneos, con lo cual el sistema ya falla.

-Problemas en entornos complicados:

Si no hay mucha luz ambiental las cámaras no pueden captar bien la información. El LIDAR no depende de la luz ambiental, pero si tienen que captar muchos objetos a la vez, o si esos objetos están en movimiento ahí le cuesta ya captarlos bien en el entorno y hacer una distinción entre lo que tendría que ser estático o dinámico.

-Alta carga computacional:

El procesamiento conjunto de los datos del LIDAR y la cámara requiere mucha potencia. Entonces claro si resulta que tenemos que analizar un entorno pequeño pues lo puede procesar bien, pero si el entorno ya es más grande ya le empieza a

costar y puede tener dificultad para analizarlo todo de manera correcta sin que haya errores.

-Adaptación a cambios en el entorno:

Lo malo de esto es que al sistema le cuesta distinguir bien cuando hay un cambio de entorno, porque no sabe si ha habido un cambio de entorno o si lo nuevo que está captando es lo que debería captar, osea no tolera muy bien los cambios de entorno pudiendo confundir objetos que si que debería captar como objetos diferentes.

(Chen et al., 2025).

#### 4.2.14.1. LIDAR

Utiliza el mismo principio de time-of-flight:

$$\text{Distancia} = \frac{c * t}{2}$$

Proporciona alta precisión y cobertura de 360°. Pero:

- Son costosos y consumen mucha energía.
- Requieren procesamiento complejo en tiempo real

(Salamanca Céspedes & Pérez Castillo, 2022).

#### 4.2.14.1. Algoritmos de planificación

Algoritmo A\*:

- Descripción: Analiza el entorno y busca el camino más corto para llegar al objeto.
- Limitaciones: si hay un entorno que ha cambiado tiene que volver a recalcular desde cero la ruta, lo cual es computacionalmente muy costoso.
- Alcance Actual: Obviamente esto es ideal solo para entornos donde haya cambios mínimos, por ejemplo, en esta competición de Europa que siempre va a estar el mismo tablero y los mismos objetos para agarrar.

Algoritmo D\*:

- Descripción: es una versión mejorada del algoritmo A\* donde aquí sí que permite recalcular de manera rápida y eficiente la ruta hacia el destino final o el objeto.
- Limitaciones: Mejora la ruta es verdad en el momento ante cambios ambientales, pero sigue siendo muy pesado y computacionalmente ocupa mucho.
- Alcance Actual: Esto es fundamental en el entorno donde la planificación es crucial, ya que te permite hacer rutas cambiantes.

Algoritmo RRT (Rapidly-exploring Random Tree):

- Descripción: Construye un mapa muy completo para recorrer cada espacio del entorno, es decir para ver toda la ruta es posible.
- Limitaciones: Este algoritmo es bastante rápido esto es verdad, pero es que como contrapartida tiene de que no siempre encuentra la ruta más corta. Como tienen que hacer una exploración aleatoria y encima le toca verificar las colisiones en entorno que son muy densos y con muchos grados de libertad lo

hace requerir un alto poder de cómputo

- Alcance Actual: Bueno si estamos trabajando en altas dimensiones y lo que queremos es encontrar una solución rápida y no nos importa si es una solución eficiente esto nos viene bien.

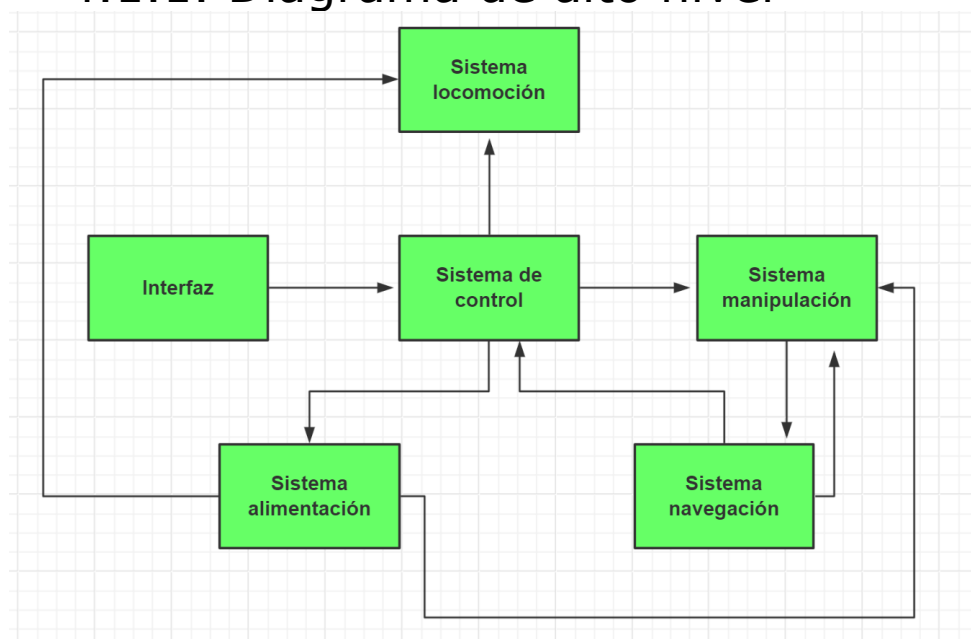
(Koenig, S., & Likhachev, M., 2005).

## 4. DESARROLLO

A continuación, en esta parte se llevaría al cabo lo que viene a ser la parte ya de la fabricación propia del robot, empezando por las ideas generales de cómo sería su funcionamiento y los diagramas pertinentes. Aquí es donde ya se elegirán todos los materiales necesarios, los componentes electrónicos y la parte de la programación, o sea esta es la parte ya del diseño total.

### 4.1. DIAGRAMA DE BLOQUES

#### 4.1.1. Diagrama de alto nivel



*Ilustración 29: Diagrama de alto nivel*

#### 4.1.2. Diagrama de bajo nivel

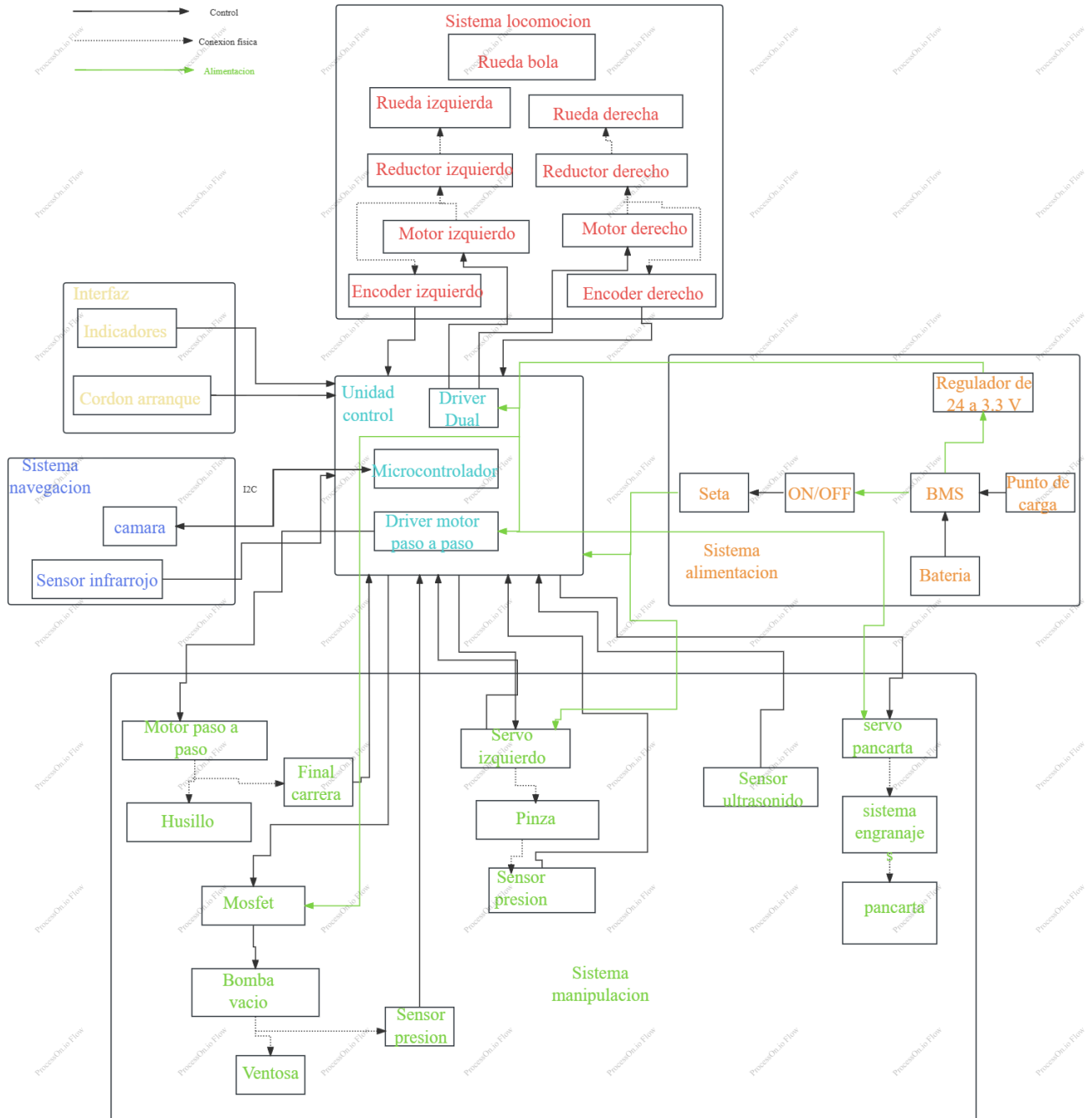


Ilustración 30: Diagrama de bajo nivel

- El sistema de arranque se encarga de poner en marcha el robot y advertir al usuario de que está funcionando.
- el sistema de energía se encarga de alimentar a todos los componentes del robot, es decir: microcontrolador, sensores, actuadores.
- El sistema de locomoción es el que se encarga de transmitir el movimiento del robot, aquí deben estar los motores para que generen movimiento y los elementos de control, como son los sensores y elementos de geolocalización para saber dónde está la posicionado el robot.

**Trabajo fin grado**

- El sistema de Navegación es el que le pasa la información al robot de aquellos elementos que se encuentren a su alrededor para tratar de no chocar con ellos y saber con cuáles debe interactuar.
- El sistema de manipulación es el que se encarga de recoger las columnas y la tabla de capas para dejarlas en una determinada posición.
- Y por último está la unidad de control que es lo que controla todos los actuadores, en esta se encuentra el microcontrolador, el driver dual y el driver del motor paso a paso

### 4.1.2.1. Interfaz

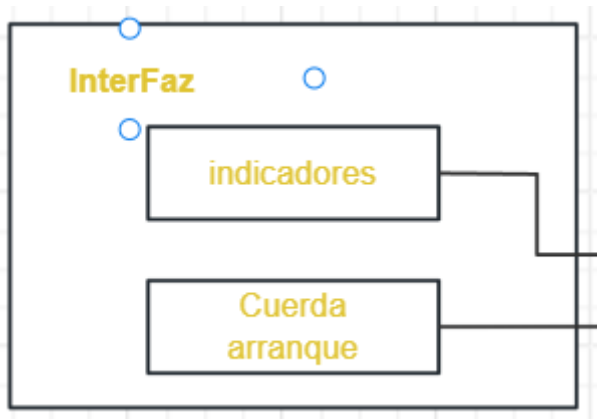


Ilustración 31: Interfaz (interacción usuario)

### 4.1.2.2. Sistema de manipulación

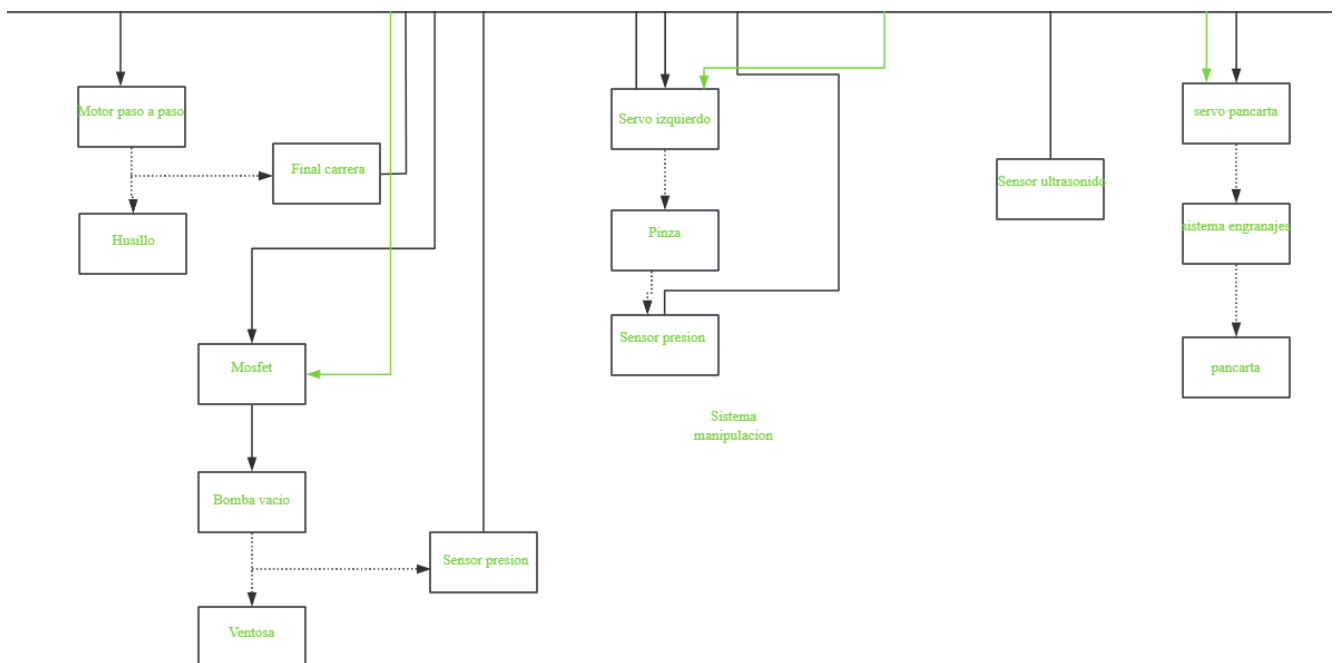


Ilustración 32: Sistema de manipulación

**Trabajo fin de grado**

Podemos ver cómo se distribuye la lista de conexiones, por ejemplo, las conexiones que son por vía cable se representan con una línea negra recta una flecha, y los que son meramente conexión física se representan con una línea con puntos; por ejemplo, el sensor de presión de la pinza está incrustado en la pinza pero no es una conexión eléctrica, o el final de carrera que no va directamente conectado en el husillo pero tiene una conexión física. Y luego la Línea Verde representa lo que viene a ser la alimentación.

### 4.1.2.3. Sistema de locomoción

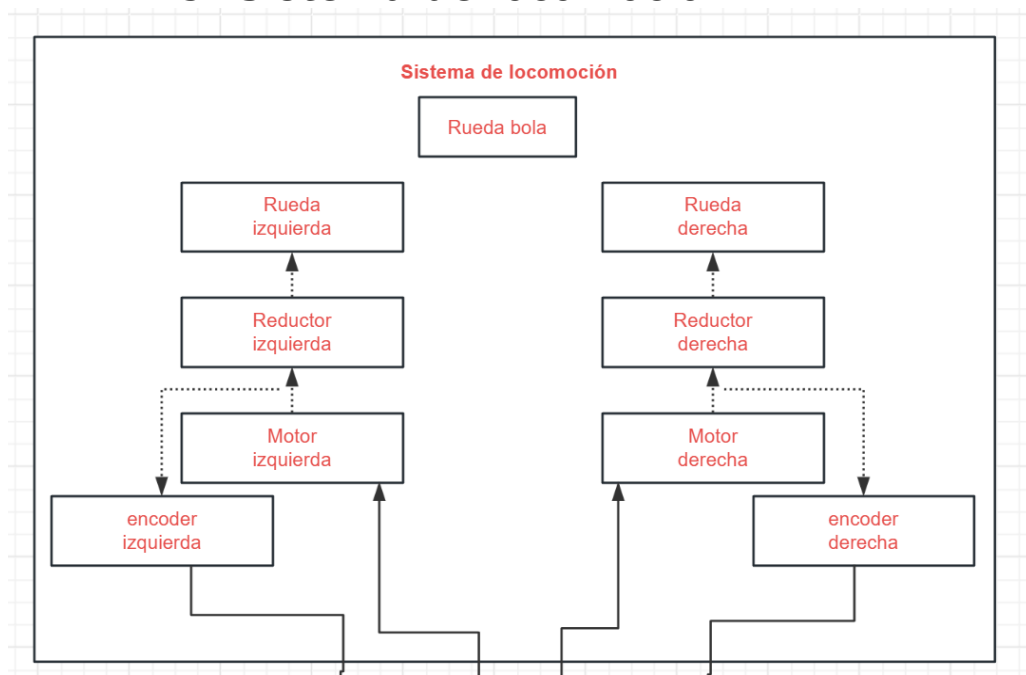


Ilustración 33: Sistema locomoción

### 4.1.2.4. Sistema navegación

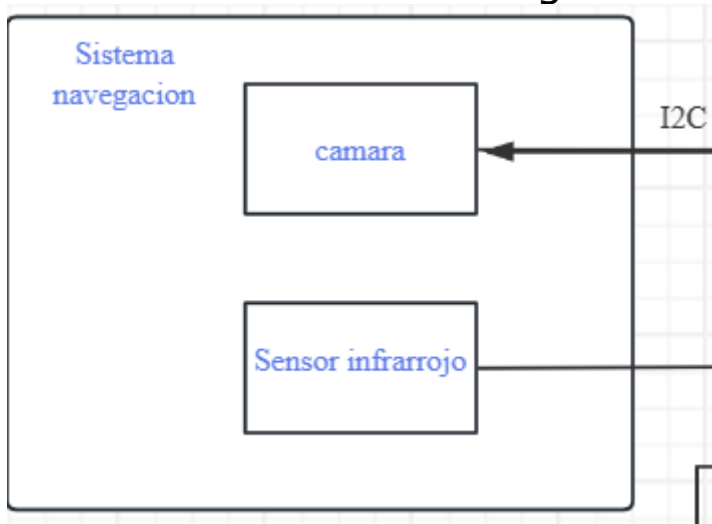


Ilustración 34: Sistema navegación

### 4.1.2.5. Sistema alimentación

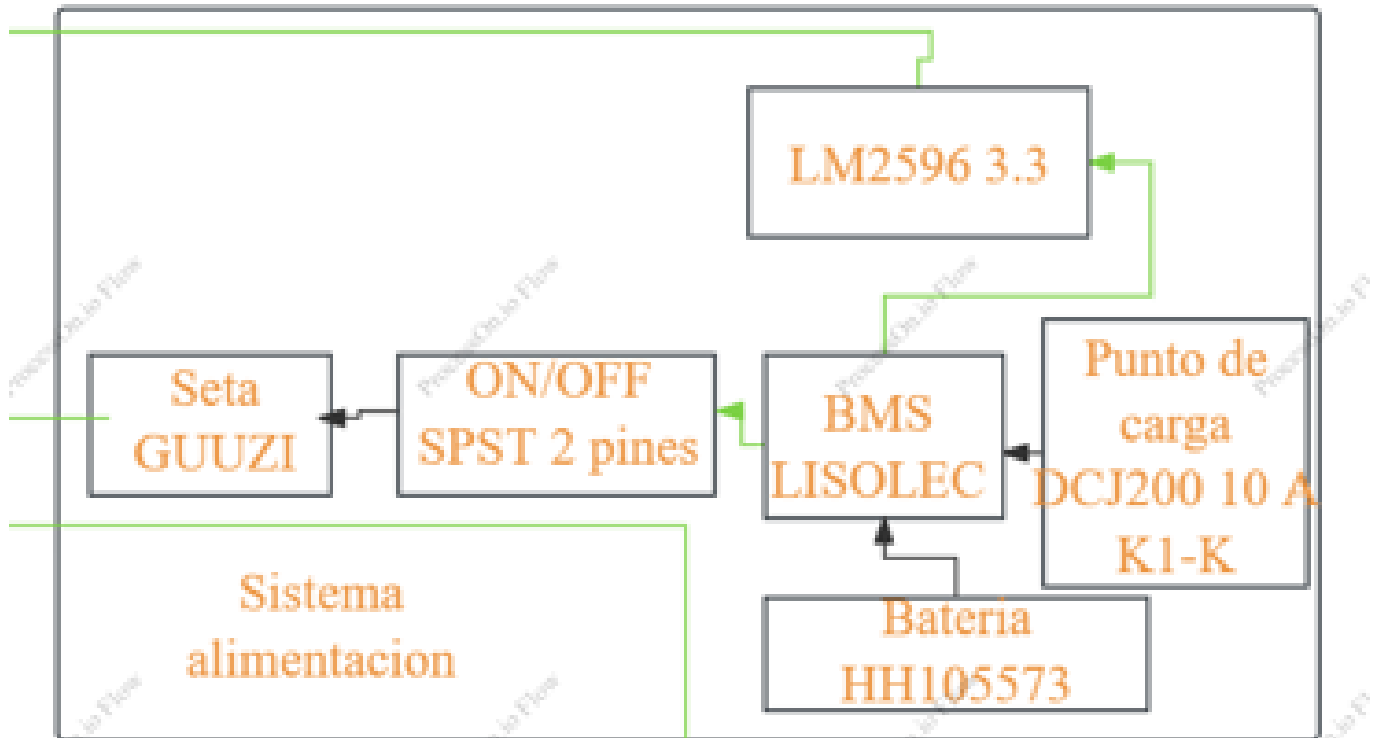


Ilustración 35: Sistema alimentación

### 4.1.3. Diagrama de bajo nivel con referencias comerciales

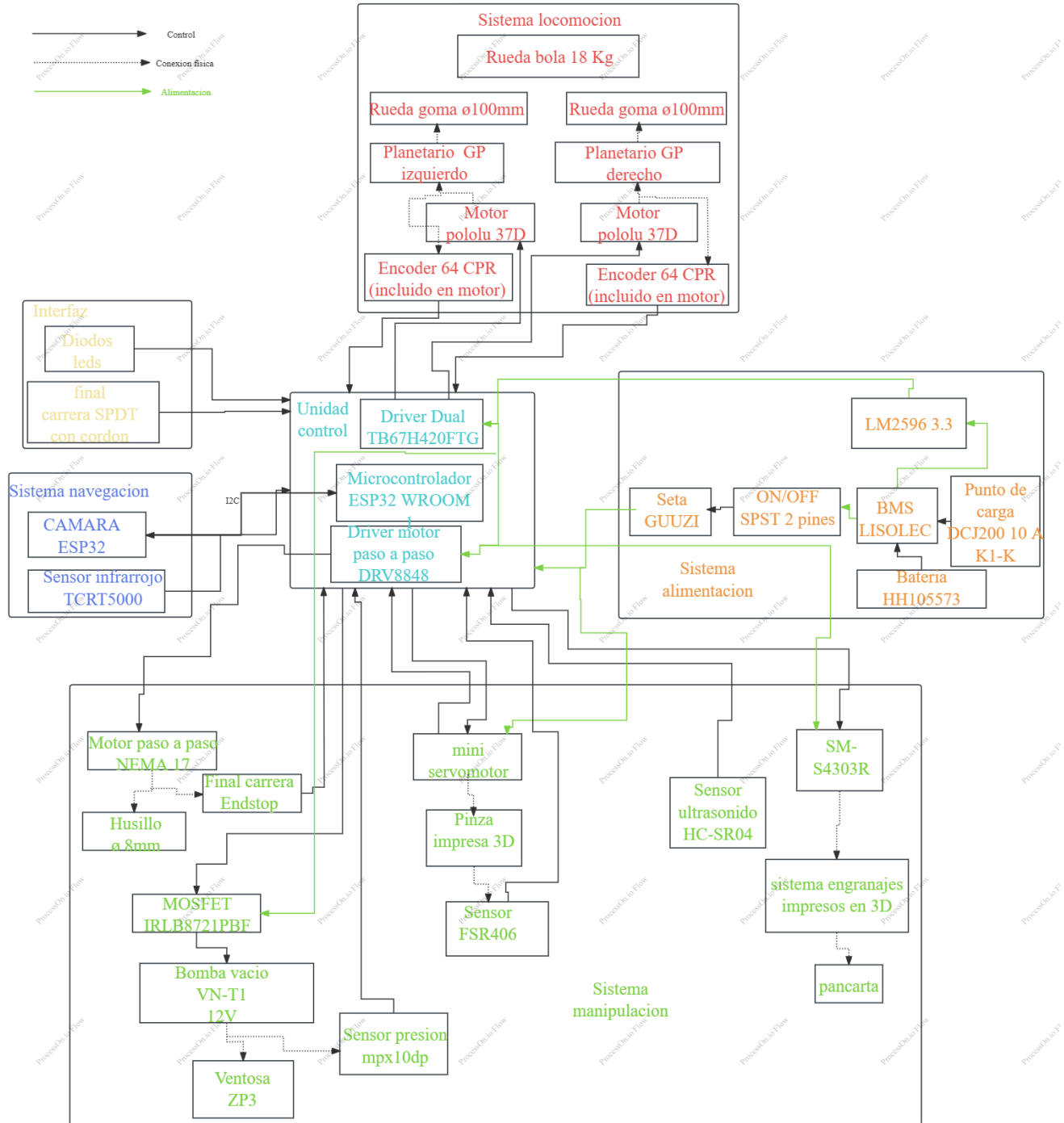
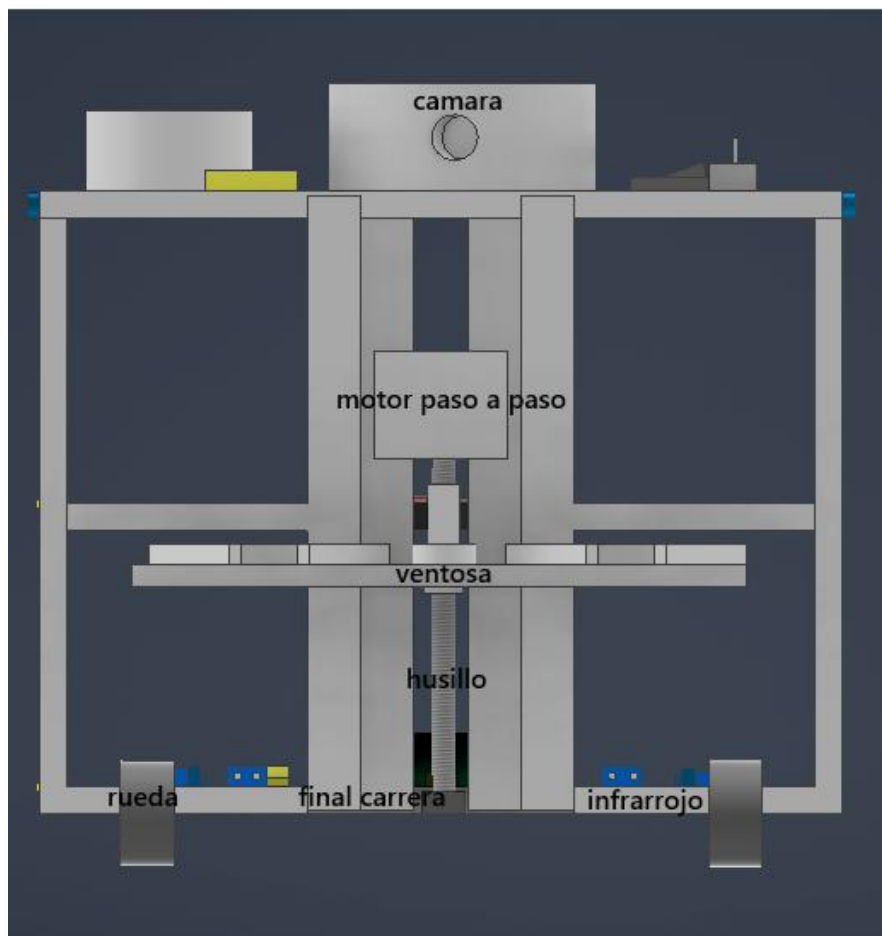


Ilustración 36: Diagrama bajo nivel con referencias comerciales

## 4.1.4. Layout

A la hora de hacer este robo me he decantado por en el sistema de locomoción coger un sistema de tracción diferencial con dos ruedas y una rueda de apoyo. Para la elevación de las pinzas y de la ventosa he decidido usar un motor paso a paso, y un final de carrera, para cuando la plataforma llegue a ese límite no siga bajando más. Cuando la pinza agarra un objeto y mientras que cuando el sensor de presión de la pinza mande información al microcontrolador, el microcontrolador le manda información de vuelta al servomotor para que no actúa sobre la pinza y esta no haga más presión sobre el objeto que está manipulando.

Como se puede ver en las imágenes mi robot se compone de 3 plantas en la tercera planta decidí optar por poner elementos como La Cámara ya que hace analiza los objetos del entorno, el botón de emergencia también lo puse arriba ya que debe estar asequible por reglamento para parar el robot en caso de emergencia, hem puesto también el interruptor para poder encender el robot de manera adecuada. También he colocado LED señalizadores ya que debe estar de manera visible el estado del robot, y por último el cordón de arranque del robot arriba del todo ya que debe ser fácil ponerlo en marcha.



*Ilustración 37: Vista delantera*

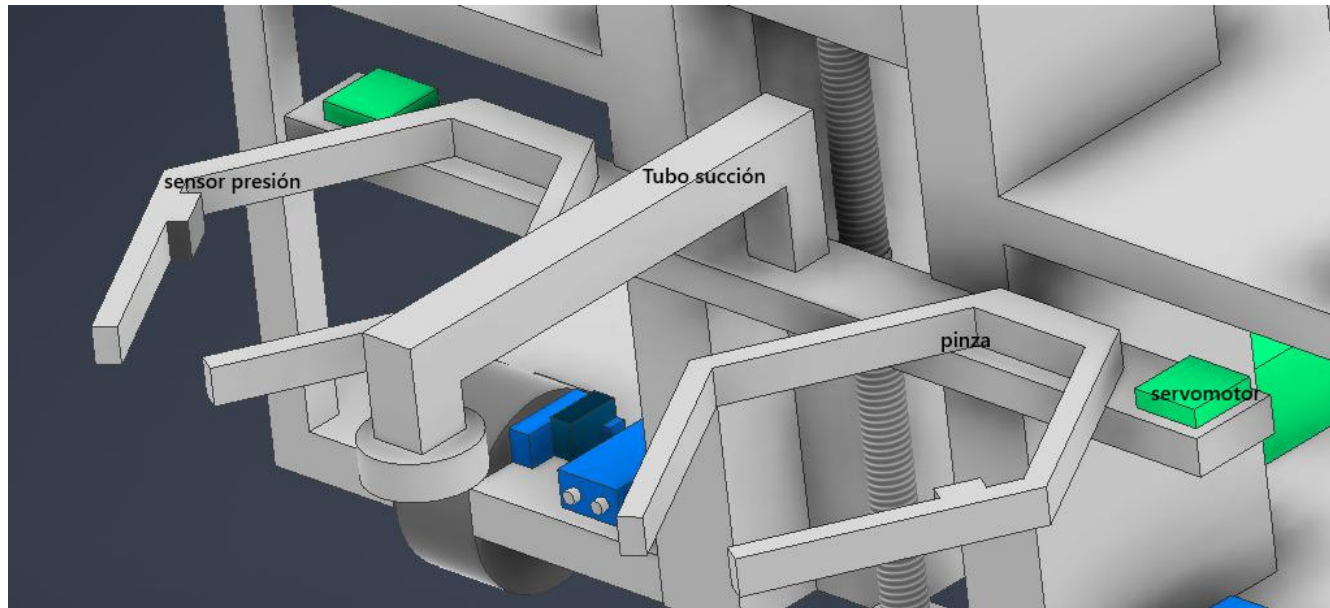


Ilustración 38: Elementos de plataforma

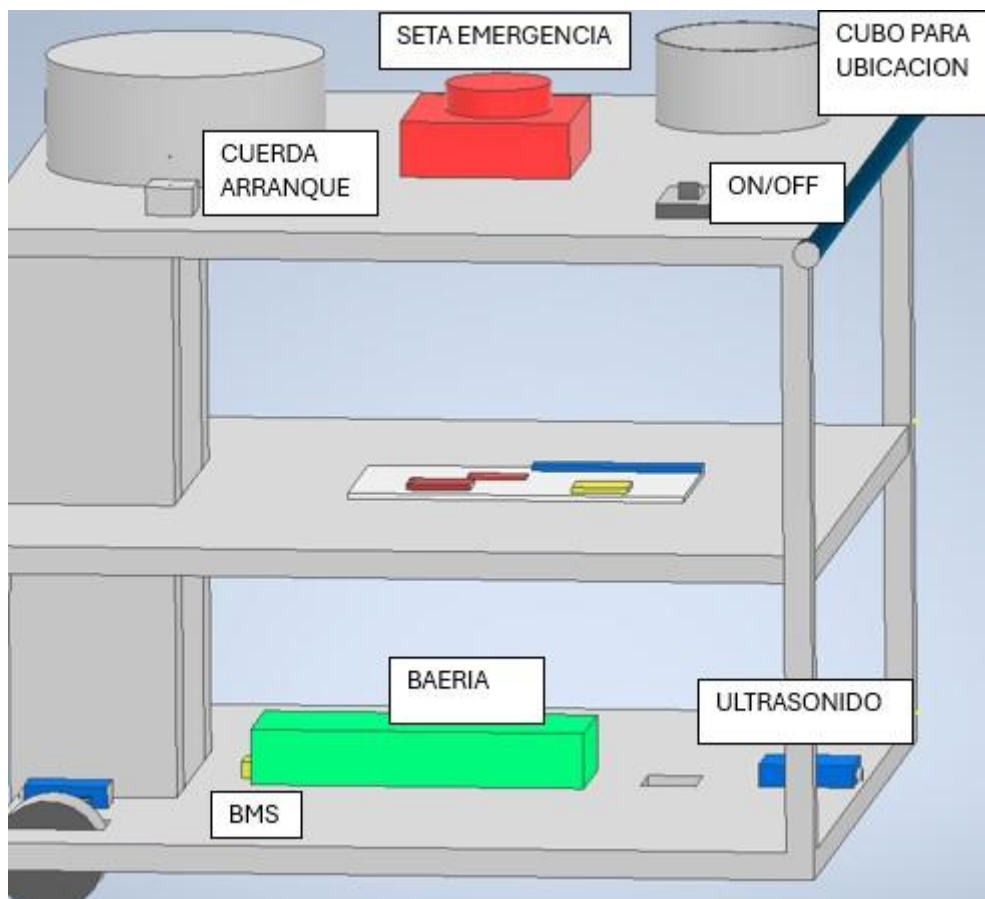
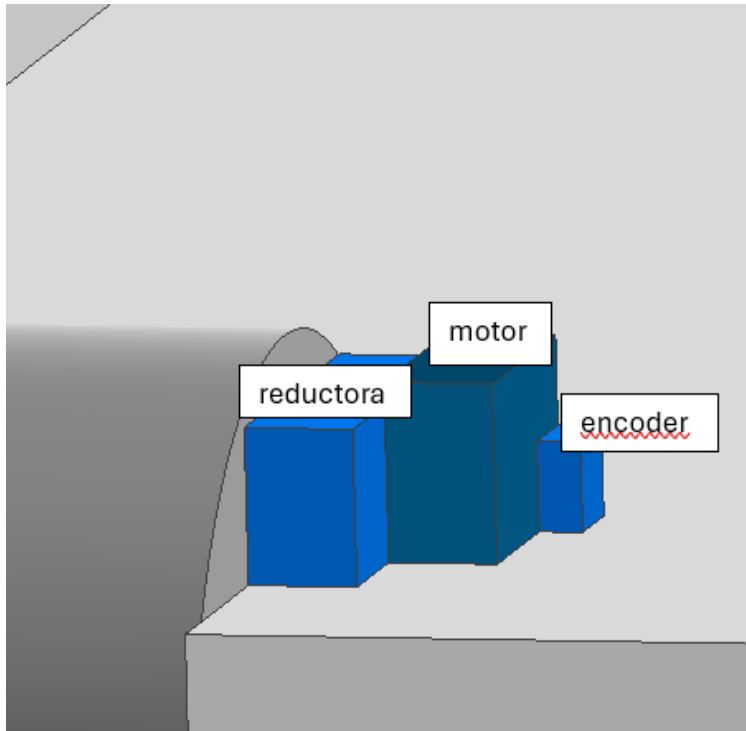
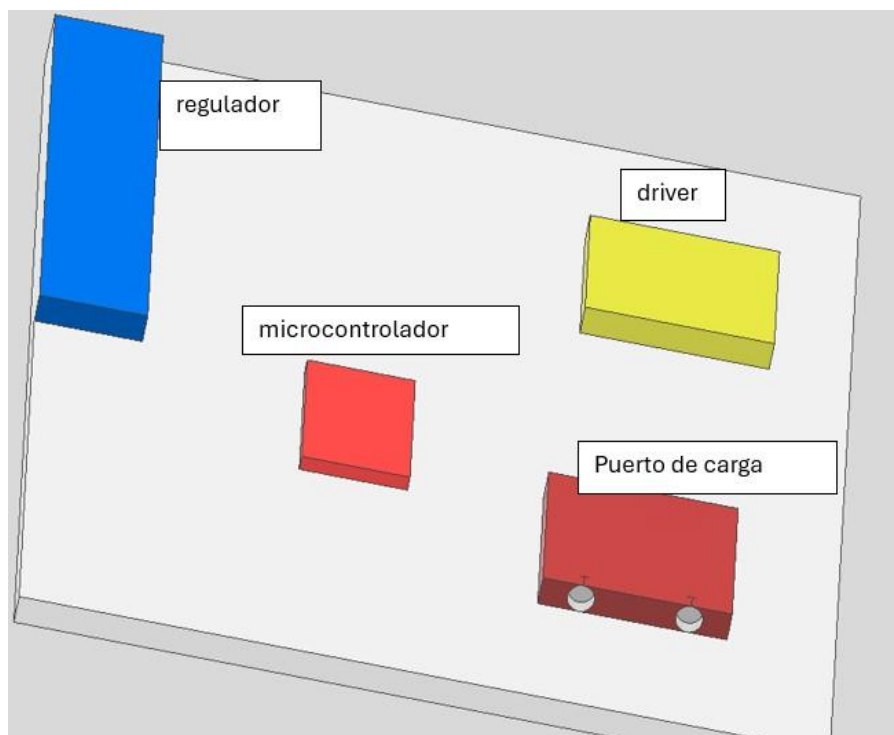


Ilustración 39: Vista frontal

La batería la he colocado en la primera planta en la parte trasera para que nos dé un poco de estabilidad en el robot y equilibrar el peso.



*Ilustración 40: Sistema locomoción*



*Ilustración 41: Unidad de control*

## 4.2. Diagramas de funcionamiento

### 4.2.1. UML casos de uso

#### 4.2.1.1. Diagrama general

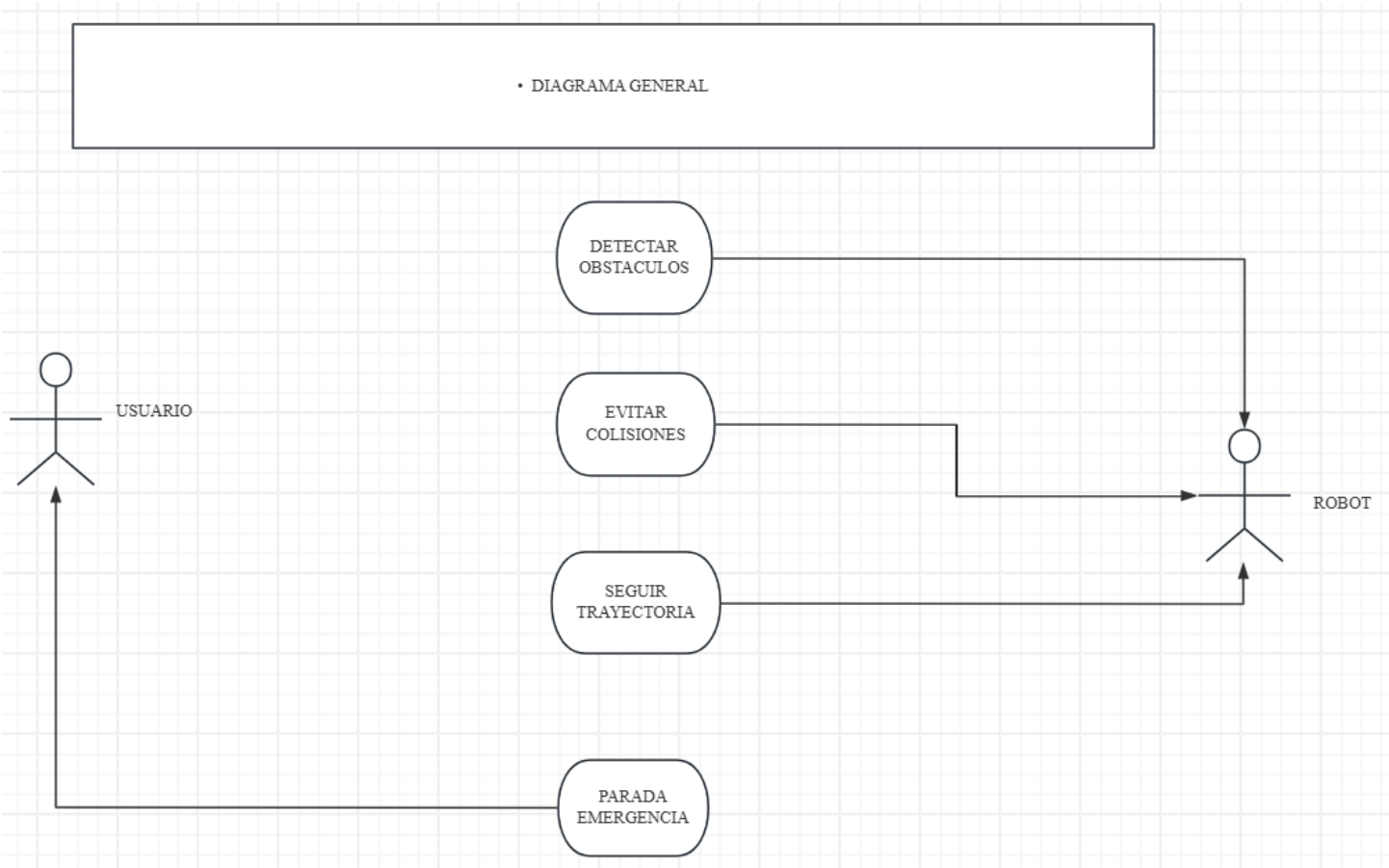


Ilustración 42: Diagrama general

#### 4.2.1.2 Diagrama de celebración

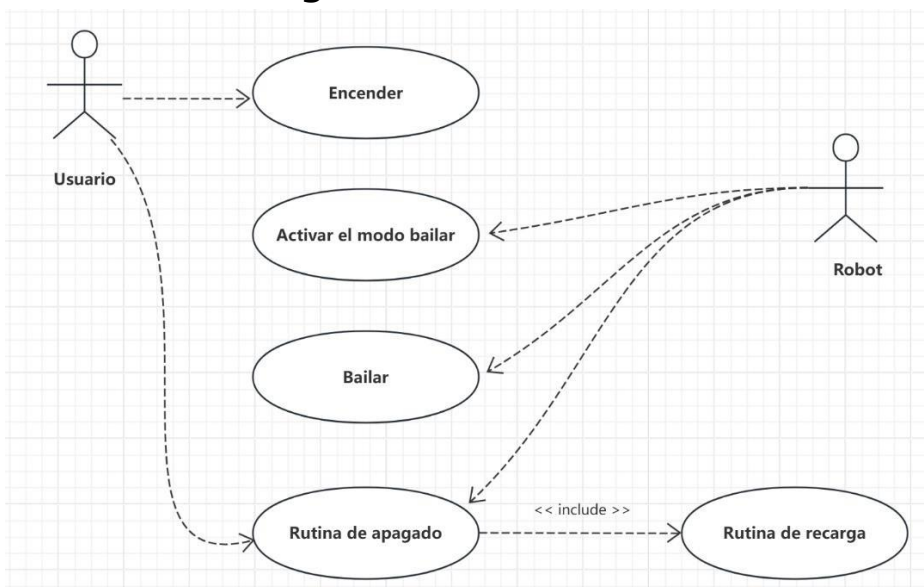


Ilustración 43: Diagrama de celebración

### 4.2.1.3. diagrama de navegación

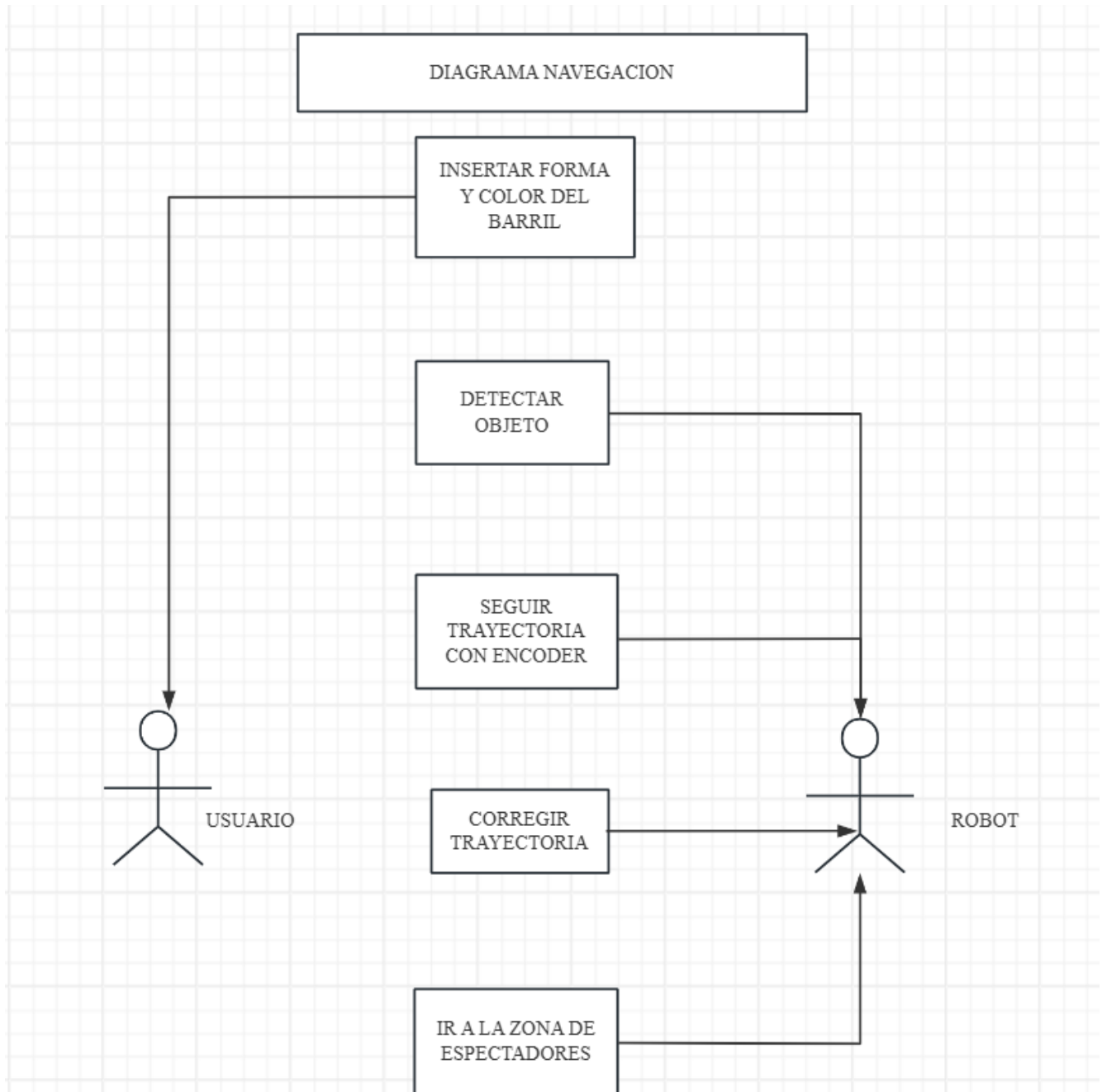


Ilustración 44: Diagrama de navegación

### 4.2.1.4. Diagrama de manipulación

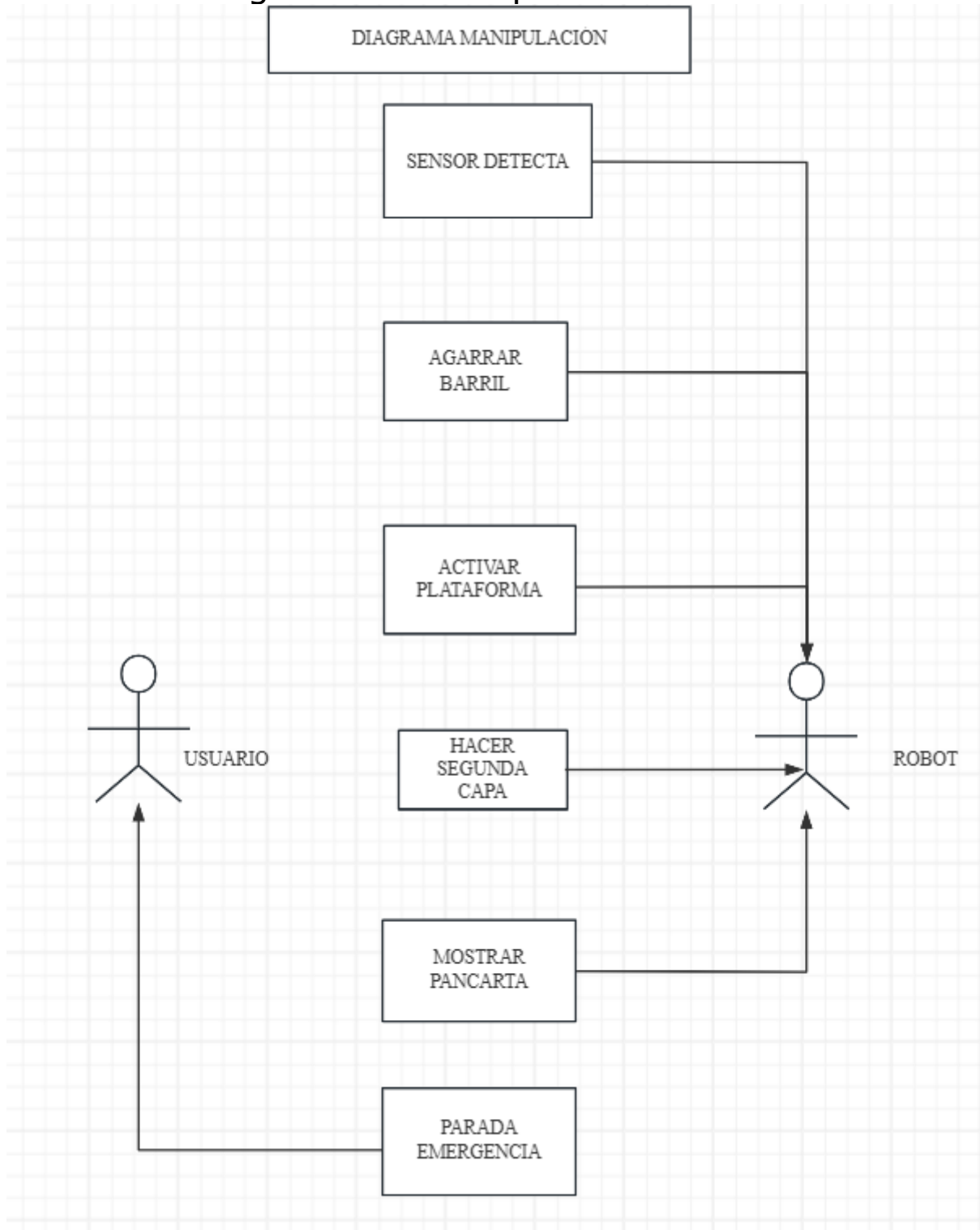


Ilustración 45: Diagrama de manipulación

## 4.2.2. UML actividad

### 4.2.2.1. Diagrama general

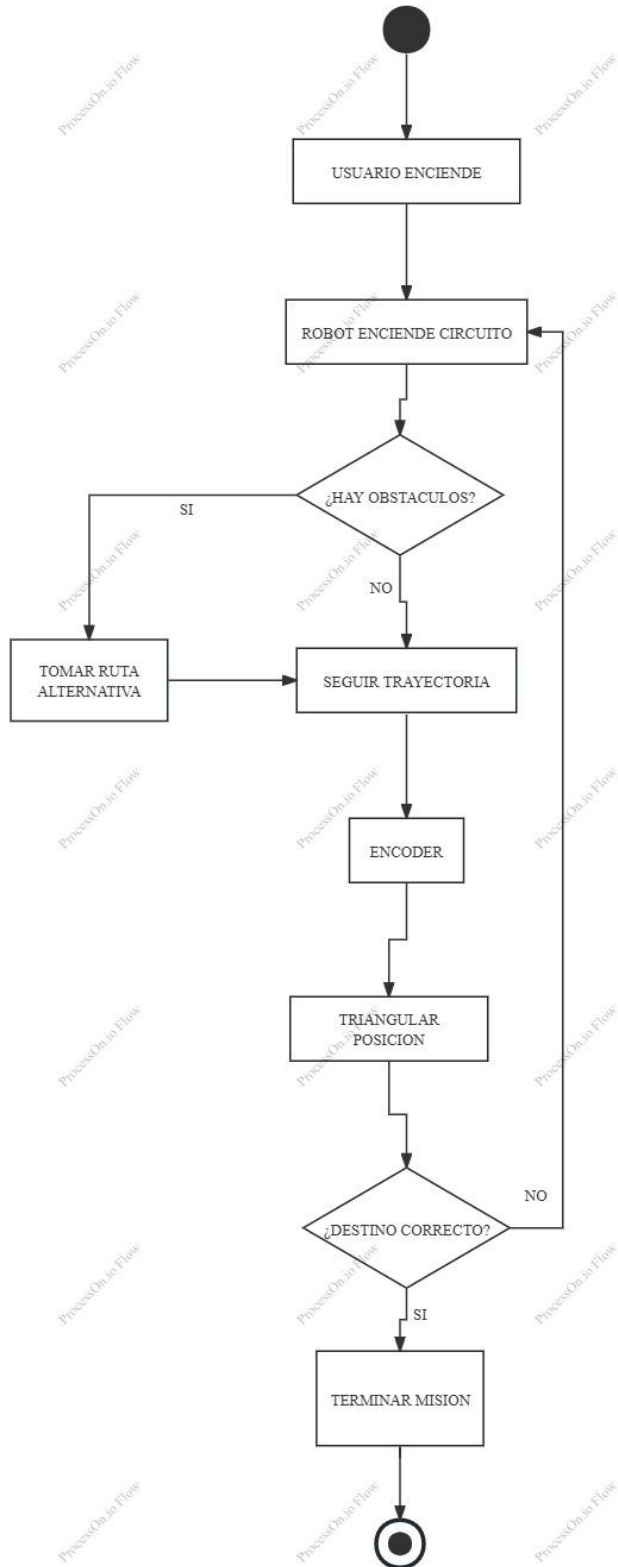


Ilustración 46: Diagrama general

## 4.2.2.2. Diagrama de celebración

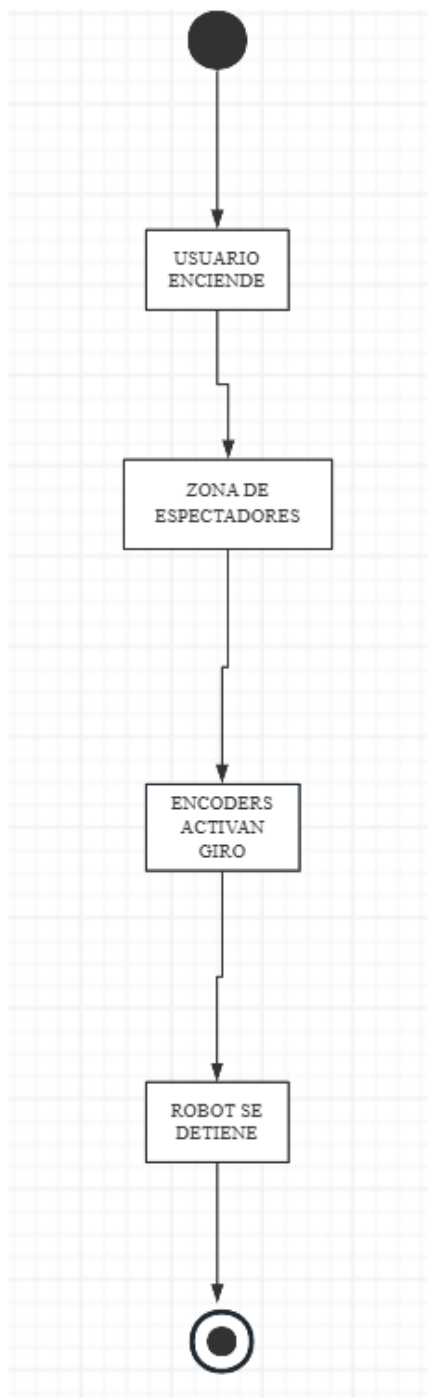


Ilustración 47: Diagrama celebración

### 4.2.2.3. Diagrama de navegación

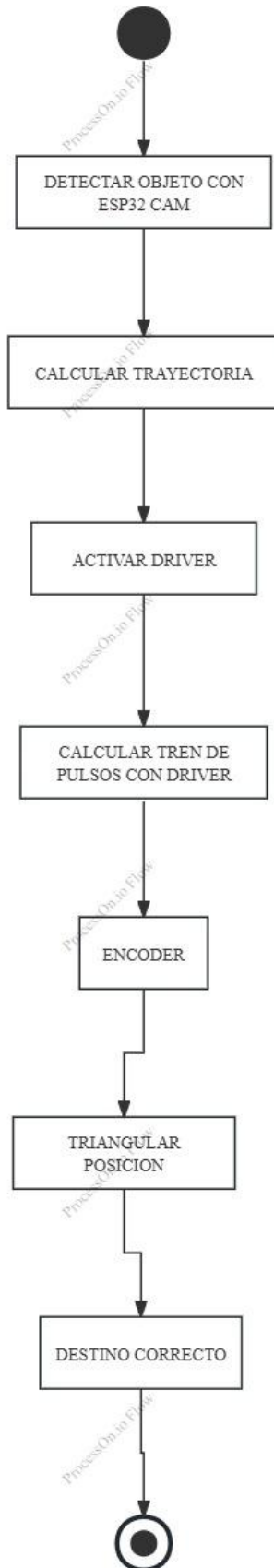


Ilustración 48: Diagrama de navegación

### 4.2.2.4. Diagrama de manipulación

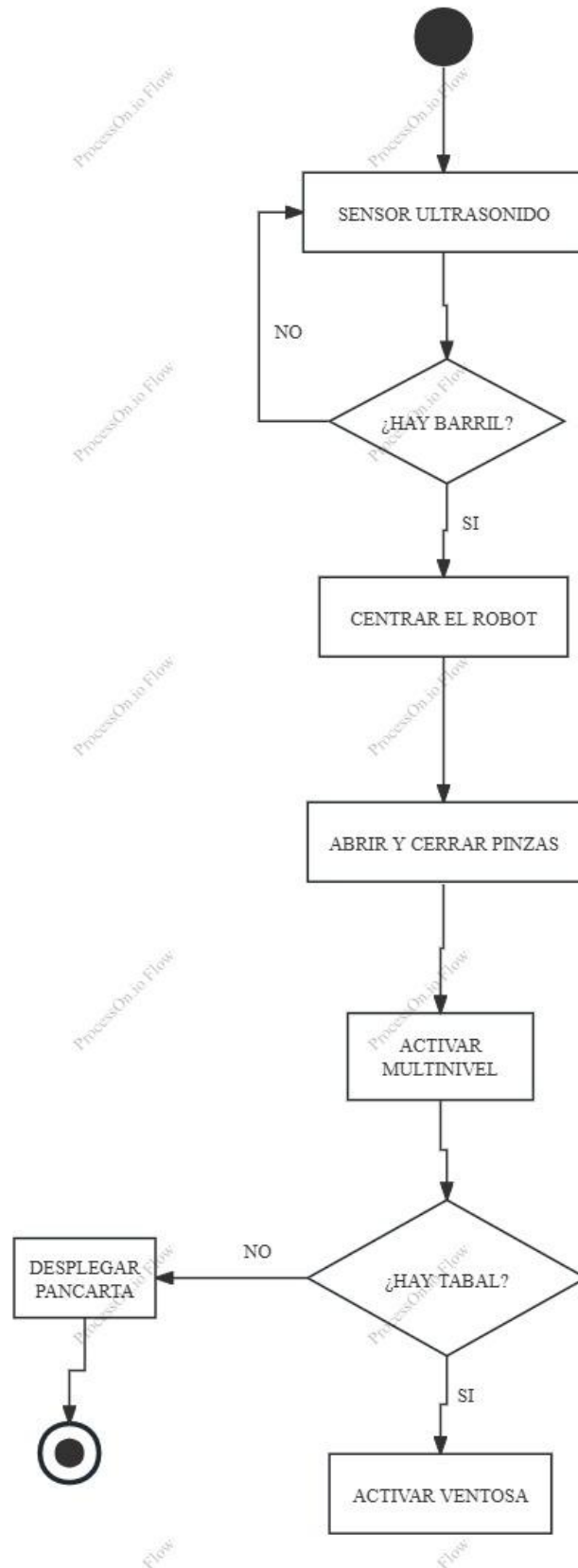


Ilustración 49: Diagrama de manipulación

## 4.2.3. Diagrama de arquitectura (clases)

### 4.2.3.1. Diagrama general

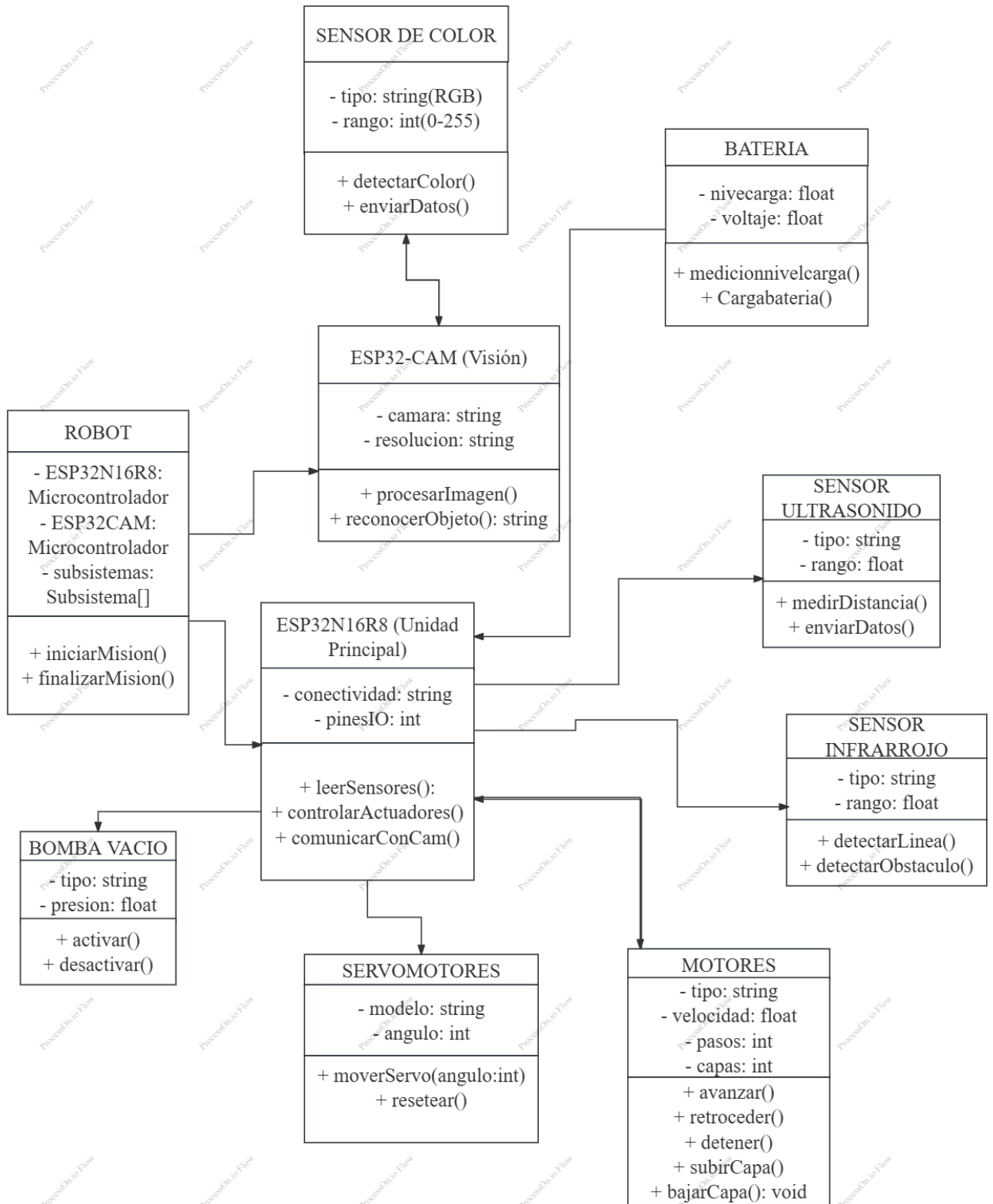


Ilustración 50: Diagrama general clases

### 4.2.3.2. Funcionamiento robot

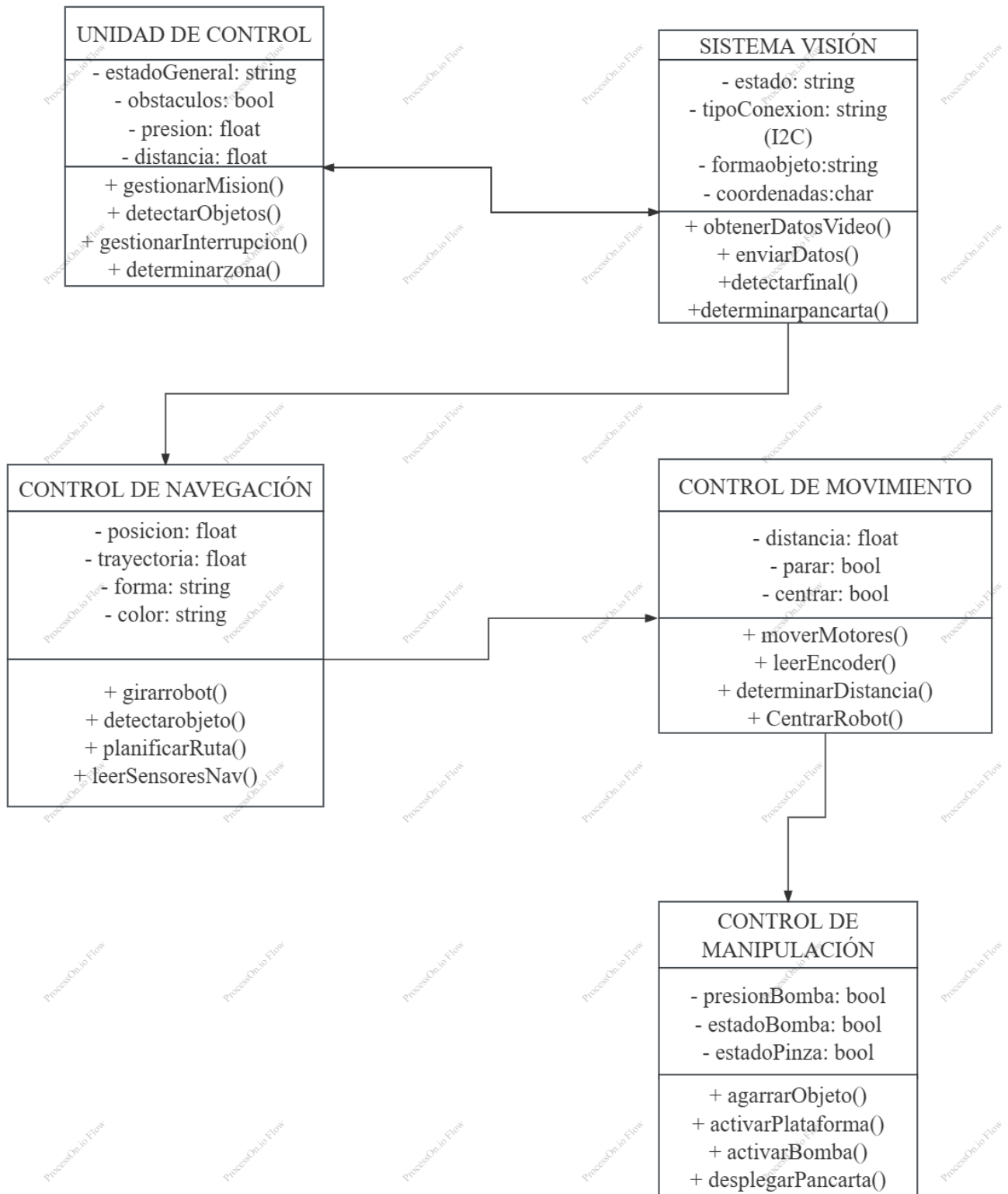


Ilustración 51: Funcionamiento general

## 4.3. CALCULO DE MOTORES, ACTUADORES Y BATERIA

### 4.3.1. CALCULO DEL SISTEMA DE LOCOMOCIÓN

<b>Aspecto importante del motor</b>			
$\omega$	Velocidad máxima	rad/s	$\omega = v/r$
M	Par motor	Nm	$M = T \cdot r$
P	Potencia máxima	W	$P = M \cdot \omega$

<b>Aspectos importantes del robot</b>	
m	Masa del robot
Q	Peso en las rueda
n	Número de ruedas total
$n_m$	Número de motrices

<b>Fuerza o mementos externos</b>	
$M_{RR}$	Momento resistente a la rodadura
$M_i$	Momento de inercia
$M_R$	Rozamientos del sistema

$$M = M_{RR} + M_i$$

$$M_{RR} = N' \cdot C_{RR}$$

$$N' = Q' = m \cdot g / n_m$$

$$M_i = F_i \cdot r$$

$$F_i = m \cdot a = (m/n_m) \cdot (v/t)$$

$$T \cdot r = (m \cdot g/n_m) \cdot C_{RR} + (m/n_m) \cdot (v/t) \cdot r$$

$$F_R = \mu \cdot N$$

$$N = m \cdot g/n$$

Si:

$$T < F_R \rightarrow \text{Rodadura}$$

$$T > F_R \rightarrow \text{Deslizamiento}$$

### CALCULO LOCOMOCIÓN

A la hora de elegir algún tipo de material debemos tener en cuenta lo que sería el tipo de movimiento, en mi caso me he decidido elegir un sistema de tracción diferencial, porque puedo realizar movimientos cerrados, y solo hacen falta dos ruedas motrices y una de apoyo. Es por ello que he considerado escoger ruedas convencionales. Debido a que quiero moverme en una superficie plana que no tiene desniveles y encima hay un espacio muy pequeño para moverme.

Entonces para empezar a realizar los cálculos lo primero que debemos ver son los elementos que componen nuestro sistema de locomoción.

Ruedas: dos motrices y una de apoyo

Motores: una para cada rueda

Reductora: una para cada rueda

Cálculo del motor:

Para saber qué tipo de motor puedo coger y que se adapte perfectamente, debo averiguar el par y la potencia que se va a requerir para mover el robot

Entonces debo definir los parámetros que va a tener mi robot:

2 ruedas motrices  $n_m = 2$

1 rueda de apoyo  $n = 1$

masa estimada = 10 kg,  $N = 49N$

radio ruedas motrices = 30 mm = 0.03m

$V_{max} = 1 \text{ m/seg}$

$t = 2 \text{ s}$

$a_{deseada} = 0.8 \text{ m/s}^2$

una vez sabidos estos datos ya puedo empezar a calcular. Primero la velocidad angular:

$$W = \frac{V_{max}}{r} = \frac{1}{0.03} = 33.3 \text{ rad/s}$$

A continuación se comprueba el peso de cada rueda motriz:

$$N' = Q = \frac{m}{mm} * g = \frac{33.3}{2} * 9.81 = 163.5 \text{ N}$$

Para el momento tendré en cuenta que CMM=0.005 debido a las características del suelo y de la propia rueda:

$$M = N' * cmm + \frac{m}{nm} * a * r = 163.5 * 0.005 + \frac{10}{2} * 0.03 * 0.8 = 0.9375 \text{ Nm}$$

$$T = \frac{M}{r} = \frac{0.9375}{0.03} = 31.25 \text{ N}$$

Ahora se debe comprobar si hay deslizamiento, al ser una rueda de goma sobre una superficie lisa se escoge un rozamiento de 0.6

$$Fr = \mu * \frac{m * g}{n} = 0.6 * \frac{10 * 9.81}{1} = 58.86 \text{ N}$$

$$Fr > T \rightarrow 58.86 > 31.25$$

Con esto tenemos que no hay deslizamiento.

Ahora procedo a calcular la potencia del motor:

$$p_{motor} = \frac{m * a * v}{2\pi * \eta} = \frac{10 * 0.8 * 1}{2 * \pi * 0.8} = 1.59 \text{ W}$$

Con estos datos tengo que elegir un motor con 1.59W y un par de 0.8925 Nm

En donde me he decidido por el modelo Pololu 37D Metal Gearmotor 24 V

Que tiene una potencia de eficiencia de 4W y una potencia máxima de 12W La potencia de eficiencia del motor ya supera con creces la potencia necesaria, y la potencia máxima del motor no haría falta utilizarla con lo cual este motor nos vale más que de sobra.

$$P_{motor} > P_{necesaria} \rightarrow 4.9 \text{ W} > 1.59 \text{ W}$$

Entonces con esto se comprueba que la potencia que suministra el motor es más que suficiente que la que necesito.

El motor es un [pololu](#) 37D 50:1 24 V.



*Ilustración 52: Pololu 37D*

A continuación, calcularé el par de la reductora:  $M(\text{reductora}) = 4.3 * M(\text{motor}) * 4.3 * 0.48 = 2.06\text{Nm}$

Que sería más que suficiente, ya que sería mayor al par total y el par de las columnas a mover

### 4.3.2. Calculo motor paso a paso

Cálculo para subir y bajar la plataforma con las pinzas y el sistema de succión

Datos:

- Peso Pinza: 55 gr
- Peso plataforma: 700 gr
- Peso barriles: 215 gr
- Peso tabla: 300 gr
- Peso sistema de succión: 700 gr
- peso servomotores: 120 g
- Paso del Husillo trapezoidal: 2 mm
- Diámetro primitivo del husillo: 8mm
- Coeficiente de rozamiento del husillo: 0.5
- Coeficiente de rozamiento del collarín: 0.4
- Diámetro primitivo del collarin: 8mm

$$\text{peso} = 700 + 55 * 2 + 215 + 300 + 700 = 2145 \text{gr}$$

$$T_s = \frac{\text{Peso} * d_{\text{phusillo}}}{2} * \frac{\text{phusillo} * \pi * u_{\text{husillo}} * d_{\text{phusillo}}}{\pi * d_{\text{phusillo}} - u_{\text{husillo}} * \text{phusillo}} + \frac{u_{\text{collarin}} * d_{\text{pcollarin}} * \text{peso}}{2}$$

$$T_s = \frac{2.145 * 0.008}{2} * \frac{0.002 * \pi * 0.5 * 0.008}{\pi * 0.008 - 0.5 * 0.002} + \frac{0.4 * 0.008 * 2.145}{2}$$

$$T_s = 3.248 * 10^{-3} \text{Nm}$$

$$T_I = \frac{\text{Peso} * d_{\text{phusillo}}}{2} * \frac{-\text{phusillo} * \pi * u_{\text{husillo}} * d_{\text{phusillo}}}{\pi * d_{\text{phusillo}} + u_{\text{husillo}} * \text{phusillo}} + \frac{u_{\text{collarin}} * d_{\text{pcollarin}} * \text{peso}}{2}$$

$$T_I = \frac{2.145 * 0.008}{2} * \frac{-0.002 * \pi * 0.5 * 0.008}{\pi * 0.008 + 0.5 * 0.002} + \frac{0.4 * 0.008 * 2.145}{2}$$

$$T_I = 3.23 * 10^{-3} \text{Nm}$$

He optado por un motor paso a paso [NEMA 17](#), ya que tiene 0.14 Nm, suficiente para subir y bajar la plataforma, con una tensión de 12V y corriente de 1.5 A



Ilustración 53: Nema 17

### 4.3.3. Calculo ventosa de succión

Para la fuerza de sujeción de la ventosa me basare en la siguiente ecuación, proporcionada por el fabricante

#### Cálculo

$$W = P \times S \times 0.1 \times \frac{1}{t}$$

W : Fuerza de elevación [N]

P : Presión de vacío [kPa]

S : Área de ventosa [cm<sup>2</sup>]

t : Factor de seguridad Elevación h  
Elevación

Ilustración 54: Calculo ventosa

Diámetro de ventosa [mm]	ø1.5	ø2	ø3.5	ø4	ø6	ø8	ø10	ø13	ø16	
Área de ventosa S [cm <sup>2</sup> ]	0.02	0.03	0.10	0.13	0.28	0.50	0.79	1.33	2.01	
Presión de vacío [kPa]	-85	0.15	0.27	0.82	1.07	2.4	4.2	6.6	11.3	17.1
	-80	0.14	0.25	0.77	1.00	2.2	4.0	6.2	10.6	16.1
	-75	0.13	0.24	0.72	0.94	2.1	3.7	5.8	10.0	15.1
	-70	0.12	0.22	0.67	0.88	1.9	3.5	5.5	9.3	14.1
	-65	0.11	0.20	0.63	0.82	1.8	3.2	5.1	8.6	13.1
	-60	0.11	0.19	0.58	0.75	1.7	3.0	4.7	8.0	12.1
	-55	0.10	0.17	0.53	0.69	1.5	2.7	4.3	7.3	11.1
	-50	0.09	0.16	0.48	0.63	1.4	2.5	3.9	6.7	10.0
	-45	0.08	0.14	0.43	0.57	1.2	2.2	3.5	6.0	9.0
-40	0.07	0.13	0.38	0.50	1.1	2.0	3.1	5.3	8.0	

Ilustración 55: Eleccion ventosa

Se definirá una serie de parámetros:

- Tabla de capas: 0.5 kg =W
- Presión vacío: presión: 80 kpa
- factor seguridad (t): 1

despejando s=6.25

entonces elijo una ventosa ZP3 de diámetro 5.



Ilustración 56: Ventosa de succión

## 4.3.4. Calculo alimentación

### Consumo actuadores:

- motores:  $3.6 * 2 = 7.2$
- servomotores:  $1.2 * 4 = 4.8$
- motor paso a paso con encoder: 1A
- bomba vacio: 1A

### consumo sensores:

- ESP32 CAM: 0.5 A
- sensor ultrasonidos: 0.015A
- sensores infrarrojos: 0.015A
- sensor pelicula fina:  $0.01 * 2 = 0.002A$
- sensor de color: 0.015A
- Sensor presión ventosa: 0.02A

TOTAL CONSUMO: 14.485 A

Para el cálculo de la batería debo tener en cuenta la tensión máxima a suministrar que serían unos 24 voltios y una corriente, que por estimación serían un 14.485 A. necesitamos que la batería tenga duración de por lo menos 10 minutos (margen para la competición)

$$\frac{1}{6} * 14.485 = 2.41 \text{ ah}$$

Necesito una batería de 24 voltios y 2.41 ah

No todos los componentes funcionarían a la vez y mucho menos en su corriente stall siempre, pero así hay un margen grande

seleccióno baterías de lipo, y la batería que más se ajusta a mis cálculos es una [LI-ION](#)

Modelo: Li-ion 18650

Voltaje nominal de cada celda: 3.7 V

Voltaje máximo de carga: 4.2V

$$N^{\circ}CELDA = \frac{V_{TOTAL}}{V_{CELDA}} = \frac{24}{3.7} = 6.5 \approx 6 \text{ CELDAS}$$

Cada celda tiene 2 ah, por lo tanto necesitamos 2 celdas en paralelo, y con eso tendríamos 4 ah y sería más que suficiente

Configuración final = 6s2p = 12 celdas en total

Voltaje nominal de carga:  $3.7 * 6 = 22.2 \text{ V}$

Voltaje máximo de carga:  $4.2 * 6 = 25.2 \text{ V}$

Por lo tanto la batería puede suministrar al robot toda la competición

## 4.4. ELECCIÓN DE MATERIALES

### 4.4.1. Elección servomotor

En primer lugar, tengo que elegir el servomotor que va a desplegar la pancarta que tiene que ser un motor 360°, como lo que tiene que mover es una cosa liviana prácticamente cualquier servo mini podría hacer sin problema esta tarea. Personalmente me he decantado por un servomotor tipo [SM-24303R](#), lo cogí sobre todo por el diseño, ya que tiene dos agujeros entonces se puede anclar con facilidad a la tercera planta mediante unos tornillos y unos soportes en L. Aunque no haría falta un servo con tanta fuerza, pero bueno por comodidad he cogido este.



Ilustración 57: SM-14303R

En cuanto al servomotor de las pinzas qué va a coger los barriles, inicialmente yo tenía pensado coger y de hecho cogí un servo motor llamado el [FETEECH SCS225](#). Este servomotor básicamente tiene 3 pines en donde uno es la masa otro el voltaje, y el último esa es la señal, este servomotor manda una señal tipo TTL half dúplex, es decir, que por un mismo cable manda una señal de transmisión (TX) y de recepción (RX)  
<https://www.feetechrc.com/Data/feetechrc/upload/file/20211126/6377353179891949627084800.pdf>



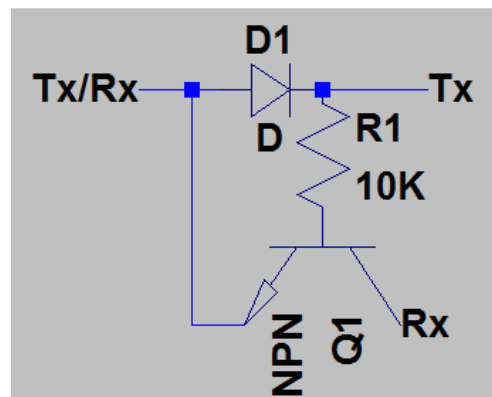
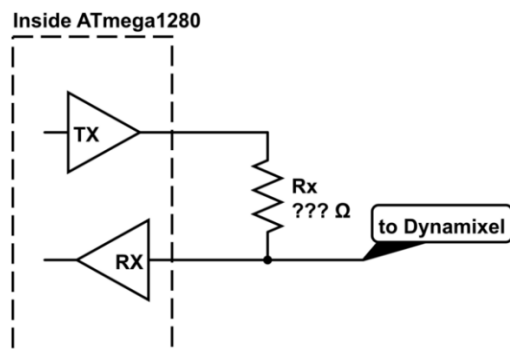
Ilustración 58: FETEECH SCS225

### Trabajo fin grado

Este servomotor tiene una cosa buena y es que tiene retroalimentación, es decir, tal cual como dice el propio datasheet este servomotor te devuelve valores, como la corriente, el voltaje, la temperatura... y mediante esos valores tú puedes determinar si la pinza agarra un objeto o no lo agarra.

Claro inicialmente esa era la idea lo que pasa es que hubieron complicaciones, en primer lugar había que coger un módulo llamado TTLinker para la correcta programación, como quería hacerlo de elaboración propia intenté hacer una conexión, al microcontrolador para que me leyera tanto los datos de transmisión como los datos de recepción sin que hubiera eco.

A la hora de probarlo use una librería "FTServo", y en los ejemplos busqué el del "feedback" de la clase SCSCCL que es de la familia SCS.



*Ilustración 59: conexiones intentadas*

Inicialmente el servomotor solo se me movía, pero no me mandaba ningún dato sobre la corriente o sobre el voltaje, me mandaba tantos vacíos.

Debido a las complicaciones que tuve, me he decidido por usar un servomotor PWM, y para saber la presión le he puesto un sensor resistivo de película fina. Al final use un [microservo](#), lo que hay que levantar no pesa mucho así que con este servo ya es más que suficiente.



*Ilustración 60: Micro servomotor*

### 4.4.2. Sensor presión pinza

Aquí me decantado por un sensor de presión resistivo que una vez agarra el objeto le envía la señal al micro, es de resistencia variable entonces si varía la resistencia va a variar el voltaje, ya que tenemos que realizar una conexión con una resistencia pull-up haciendo un tipo divisor de tensión, en donde la tensión entre las dos resistencias va a variar. He cogido este sensor de presión porque es amplio, ancho así pude sentir el objeto, [FSR406](#).

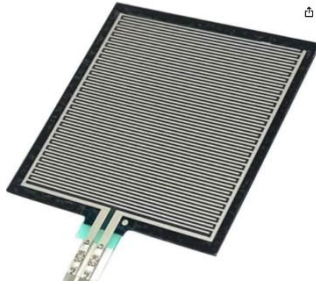


Ilustración 61: Sensor resistivo

### 4.4.3. Elección ruedas

para la elección de las ruedas debemos tener en cuenta el diámetro del robot no puede ser ni muy grande porque lo elevaría mucho, ni muy pequeño porque podría tener golpes con la base de la chapa. Las ruedas Deben estar hechas de goma Para que cuando entre en contacto con las superficies de rodadura sea un material lo suficientemente resistente para soportar el peso del robot. En mi caso he elegido una rueda de 60 mm de diámetro, la cual tiene un agujero de 6 mm. Las especificaciones técnicas indican que puede soportar peso elevado: [rueda](#)



Ilustración 62: Rueda goma

Para la elección de la rueda de apoyo necesito que sea una rueda que se pueda mover en cualquier dirección, entonces he escogido una rueda en forma de bola Metálica, que es capaz soportar una carga total de 14 kg, con lo cual sirve para mi robot: [Rueda bola](#)



*Ilustración 63: Rueda bola*

#### 4.4.4. Elección del botón de encendido

este elemento permite que se encienda el robot pero no da la puesta en marcha, ya que de eso se encarga la cuerda de arranque. Este interruptor on/off soporta corrientes bastante altas hasta 16 A, sin ningún tipo de problema puede aguantar mi robot y todo lo que consume, ya que en el peor de los caso mi robot alcanzaría cerca de lo 14.5<sup>a</sup>, este interruptor puede soportar sin problema el consumo:  
[interruptor](#)



*Ilustración 64: Interruptor*

## 4.4.5. Final carrera cordón arranque

Es el encargado de poner en marcha mi robot he seleccionado un cordón de arranque con un final de carrera, en donde al tirar de la misma el circuito se pone en marcha: [cordón arranque](#)



Ilustración 65: Cordón arranque

## 4.4.6. Sensor de ultrasonidos

En mi caso yo elegí un sensor de ultrasonidos unicá y exclusivamente para el despliegue de la pancarta. He elegido el sensor de ultrasonidos ya que detecta el suelo como un objeto, Y mide con exactitud la distancia a la que se halla un objeto. El sensor que he escogido es él [HC-SR04](#), que posee un rango de detección de 2 cm a 400 cm. Que es más que de sobra para lo que necesito.



Ilustración 66: HC-SR04

## 4.4.7. Sensor de color

He elegido el sensor de color ya que la competición tiene un tapete que tiene diferentes zonas que se dividen en colores, entonces puedo saber en qué zona del tapete estoy para así saber dónde dejar la pieza y cuánto me queda para llegar a la pieza. He cogido este o sensor en concreto ya que tiene un rango de detección de 25 mm, lo cual es suficiente para esta competición y para mi propio robot. El sensor es el [TCS230 TCS3200](#)



Ilustración 67: TCS230 TCS3200

#### 4.4.8. Sensor de infrarrojos

he elegido este tipo de sensor para detectar los objetos que se deben coger con las pinzas y ventosa, me he decantado por este sensor en lugar del sensor de ultrasonidos ya que este sensor no requiere que se le ponga en un tipos de posiciones específica ya que el suelo no resulta un obstáculo. Este sensor tiene un rango de detección lo suficientemente largo para nuestras necesidades, su voltaje de funcionamiento está entre 3.3 a 5 V, lo cual hace que se pueda conectar de manera directa al microcontrolador, el modelo elegido es [TCRT5000](#)

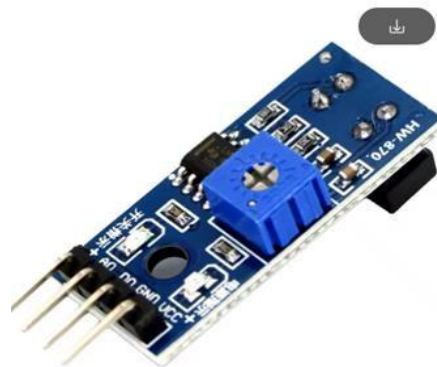


Ilustración 68: Sensor infrarrojo

#### 4.4.9. Elección de pinza

A la hora de elegir la pinza me decante por una pinza hecha en 3D de material ABS, me he basado en los antecedentes que cogí, qué es el de eurobot 2016, yo necesito que las pinzas sean rígidas y ofrezca un elevado coeficiente de rozamiento para poder agarrar los objetos con más facilidad. En el mercado me resulto difícil encontrar una pinza

**Trabajo fin de grado**

óptima, es complicado y pensé en mandarlas en 3D, pero para este proyecto lo vi alargar más el tiempo así que para hacer una demostración elegí esta pinza



Ilustración 69: Pinzas de ABS

#### 4.4.10. Elección bomba de vacío

Es la que se va a encargar de succionar aire para la ventosa, para que realice la succión de manera correcta en mi caso he elegido una bomba [VNT1](#). Qué tiene una alimentación de 12 V y una potencia de 12W, y potencia de aire es 10L/min.



Ilustración 70: Bomba vacío VN-T1

#### 4.4.11. Elección sensor de presión

Este es el sensor de presión que se incorpora a la ventosa, compara un canal que es la presión ambiente con la presión de la pinza. Envía una señal en forma analógica, la cual se le debe enviar al

microcontrolador para determinar cuándo debe dejar de succionar. He tomado un [MPX10DP](#)



*Ilustración 71: sensor de presión ventosa*

#### 4.4.12. Plataforma de elevación

Se ha desarrollado una plataforma de elevación que sirve como soporte para las pinzas y para la ventosa de succión, esta plataforma se sube y se baja mediante un motor paso a paso y husillo. Esta plataforma tiene un espesor de unos 2 mm. Para fabricar esta plataforma se ha usado madera



*Ilustración 72: Plataforma elevación*

### 4.4.13. Elección de cámara

He elegido esta cámara porque puede captar los objetos su forma y su color de una manera muy precisa, nos da su eje de coordenadas para que podamos localizarlo, aunque no tiene la precisión del LIDAR que te hace un escaneo en una zona en tres dimensiones y te dice la distancia exacta en la que están los objetos (distancia de cualquier objeto no sirve para detectar objetos solo para hacerte una representación en 3D por ejemplo una sala), pero tiene la ventaja de que sí que te ayuda a detectar el objeto de manera exacta su forma y su color, y aparte también es un microcontrolador con lo cual se puede usar para controlar sensores y actuadores. Es un microcontrolador bastante completo y es bastante económico. Por eso me he decantado por coger la ESP32 CAM



Ilustración 73: ESP32 CAM

### 4.4.14. Elección BMS

Cuando calculamos la batería, elegí un tipo de batería de litio pero para realizar el balanceo de esa batería se debe coger un BMS, en mi caso he cogido 1 BMS [Lisolec](#) de 24 voltios

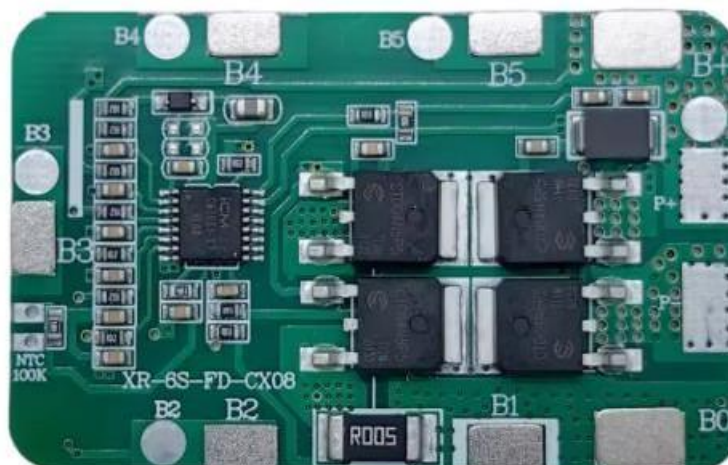


Ilustración 74: BMS Lisolec

## 4.4.15. Elección microcontrolador

Este microcontrolador es bastante completo, en primer lugar es muy fácil de programar este se programa con el mismo arduino, luego a la hora de programarlo tienes de todo, tienen muchas librerías tienes muchos ejemplos o sea es un microcontrolador muy moderno y también tiene conexión wifi. Por lo tanto me decantado por este microcontrolador ya que es bastante completo es bastante potente y para este proyecto me es más que suficiente.



ESP32-S3-WROOM-1

*Ilustración 75: ESP32 S3 WROOM 1*

## 4.4.16. Driver motor paso a paso

EL [DRV8848](#) es un controlador diseñado para proporcionar un control preciso y eficiente en sistemas que utilizan los motores de paso a paso. Ajusta la corriente de fase para optimizar la eficiencia energética y los modos de operación silenciosos. Se utilizan para un control preciso del motor paso a paso.



*Ilustración 76: DRV8848*

## 4.4.17. Final de carrera

Autor: Julio Cesar Balanta Narvaez

- 93 -

Trabajo de fin de grado

**Trabajo fin de grado**

Para este elemento vi que me servía este micro interruptor, tiene un tamaño reducido y es fácil de incrustar en la planta baja, al ser pequeño no me estorba a ningún otro componente. Este [componente](#) se pone en la base de la plataforma para que se detenga.



*Ilustración 77: Final carrera plataforma*

#### 4.4.18. Puerto de carga

Para seleccionar el tipo de conector con el que voy a alimentar mi circuito, me he decantado por un conector de Barrel Jack, ya que nos aguanta una corriente de hasta 10 amperios, con lo cual me da un margen para soportar la corriente que hay en la batería. También me resultó cómodo ya que se asemeja mucho al conector de un PC: [DCJ200- 10-A-K1-K](#)



*Ilustración 78: Puerto de carga DCJ200*

## 4.5. ESQUEMA ELECTRICO

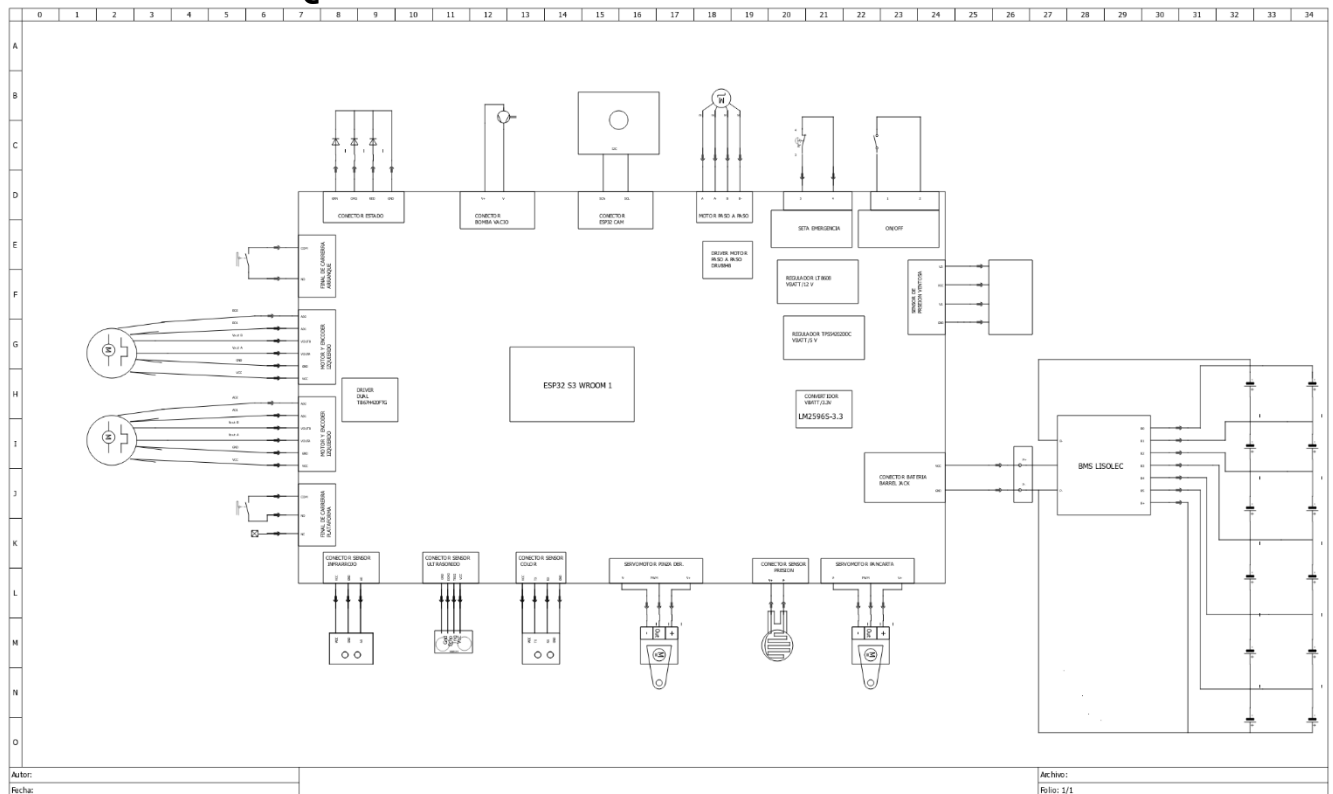


Ilustración 79: ESQUEMA ELÉCTRICO LOCOMOCIÓN Y CONTROL

### 1. Alimentación del sistema

El circuito se alimenta a través de una batería de 24 V, protegida mediante un módulo BMS, el cual gestiona la protección frente a sobrecarga, sobre descarga y equilibrio de celdas.

A partir de la fuente principal de la batería, se implementan fuentes de alimentaciones:

Una fuente de alimentación conmutada LT8608, que suministra de VBATT a 12 V y alimenta la bomba de vacío y el driver del motor paso a paso

Una fuente de alimentación TPS54202DDC que regula de VBATT a 5V para alimentar los servomotores, sensor de ultrasonido y sensor de color

Y por ultimo LM2596S-3.3, que regula de VBATT a 3.3V, se conecta antes del ON/OFF y el seta, se hace así para no dañar el micro al pulsar el botón de emergencia

Todos los módulos comparten una masa común (GND) para asegurar referencia eléctrica entre lógica, motores, servos y sensores.

### 2. Microcontrolador central

El sistema está controlado por un ESP32 S3 WROOM 1, un microcontrolador de arquitectura AVR que es muy completo en todos los sentidos

**Trabajo fin de grado**

- La lectura de sensores digitales y analógicos (infrarrojos, ultrasonido, luz, presión, etc.).
- La generación de señales PWM para el control de servomotores.
- La comunicación con encoders de motores motrices.
- El control de drivers de motores paso a paso y DC.
- La gestión de relés de encendido y emergencia.

### 3. Servomotores

Los servomotores del sistema reciben alimentación desde la línea de 5V proveniente del regulador. Los pines de control PWM están conectados a salidas del microcontrolador con capacidad de generación por hardware.

### 4. Drivers de motores motrices y paso a paso

Se utilizan dos tipos de drivers:

- Un driver para motores DC con encoder, utilizado en las ruedas motrices (tb67h420ftg).
- Un driver para motor paso a paso, para sistemas de elevación o manipulación (DRV8848).

Un driver recibe la alimentación de potencia directamente desde la fuente de 12V (VMOT). Y otro recibe los VBAT para los motores

## 4.6. DISEÑO MECANICO

### PERÍMETRO:

Según la normativa se permite un perímetro de hasta 1400 mm si la pinza está desplegada, entonces sumamos todas las dimensiones de la figura

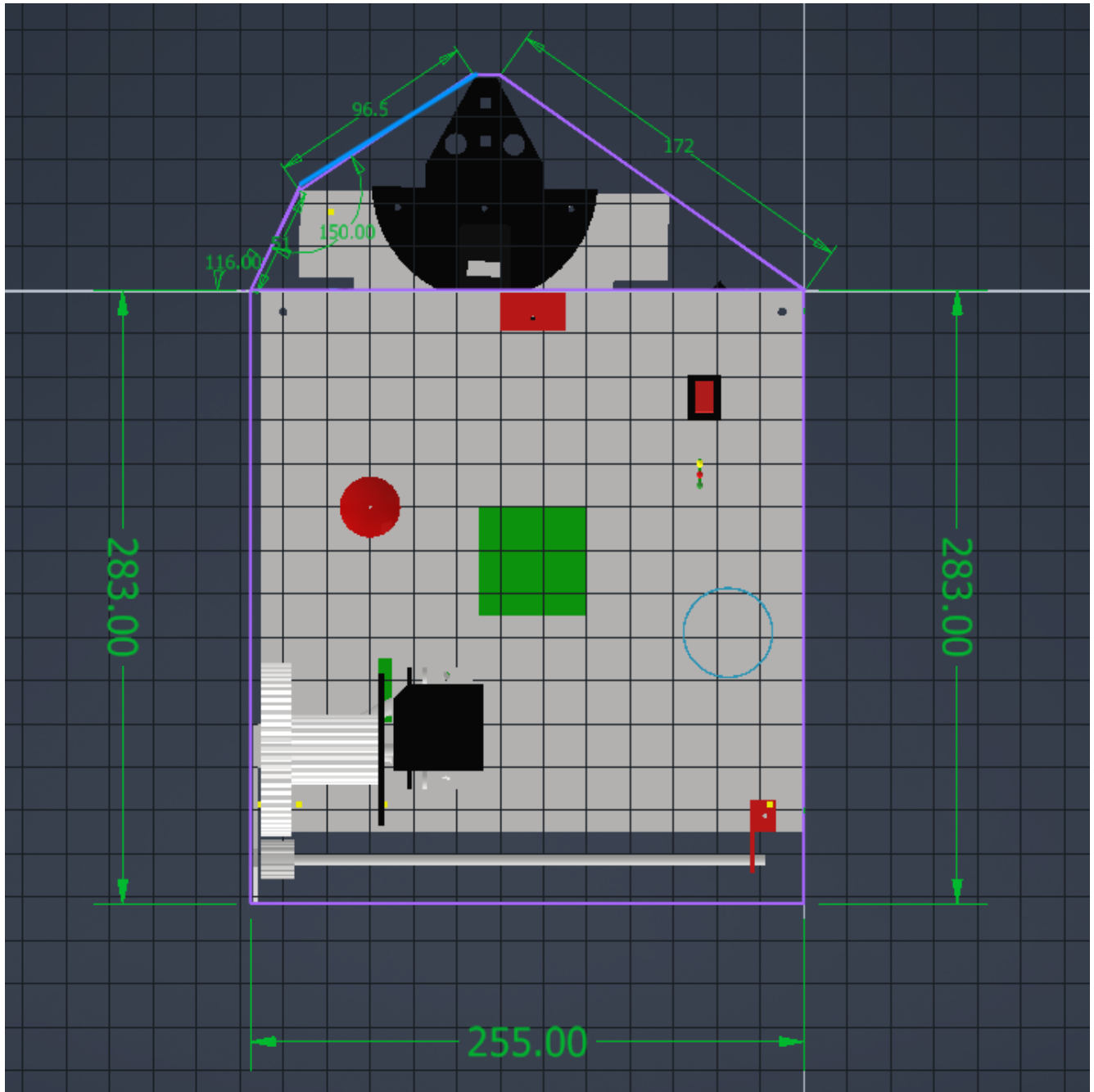


Ilustración 80: verificación perímetro

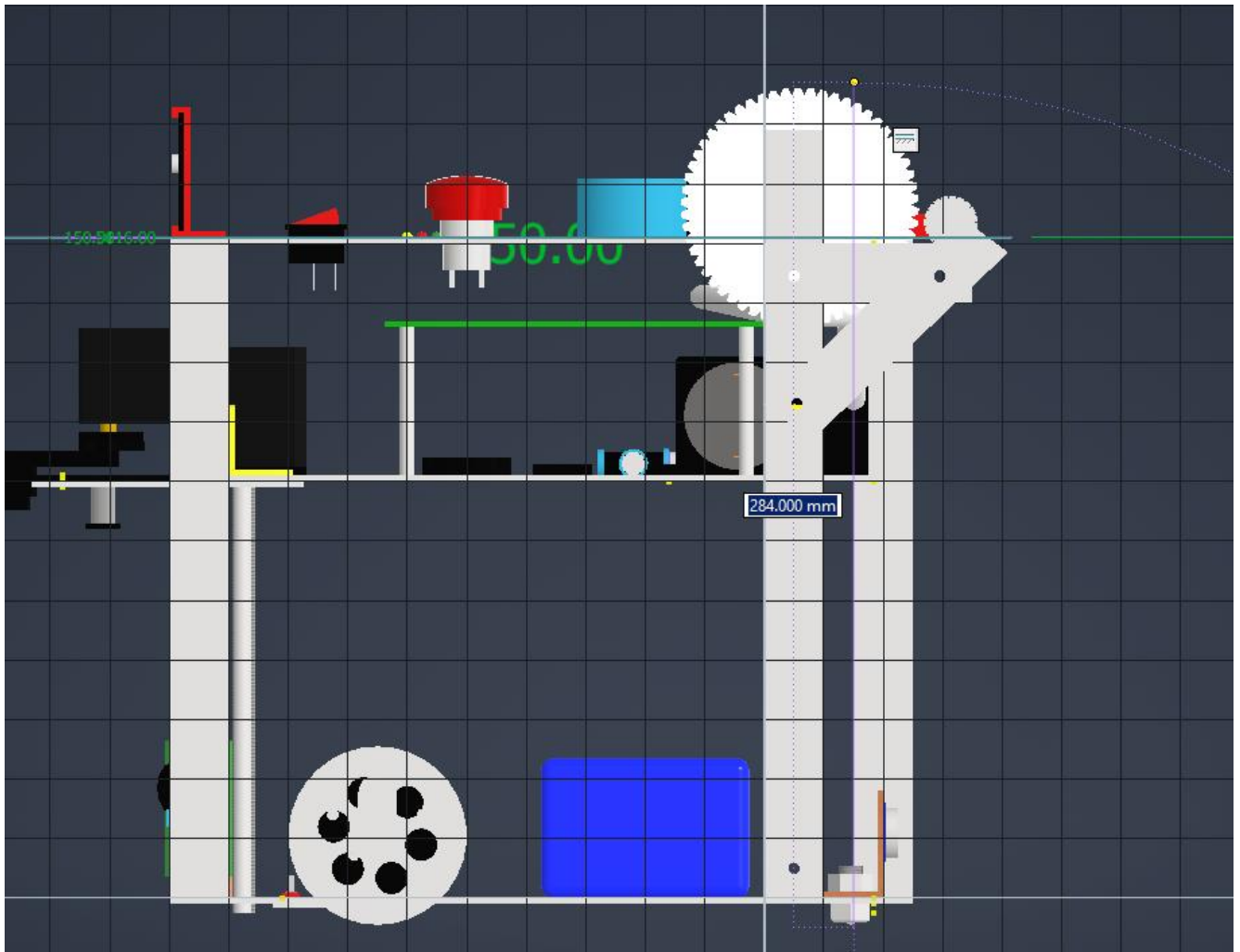
$$283+255-283+51+96.5+172=1140.5 \text{ mm}$$

Eso es con la pinza cerrada como se puede apreciar en la imagen, según las especificaciones de la pinza el robot tiene una apertura de hasta 54 mm, para poder tener un margen pongamos 60 mm con lo cual el perímetro final con la pinza desplegada que en este caso sería abierta es

PERIMETRO MAXIMO  $1140.5+60=1200.5$

$1200.5 < 1400$ , en la competición se permitía con la pinza desplegada un máximo en el perímetro de 1400 mm con lo cual esta condición la cumplo sobradamente

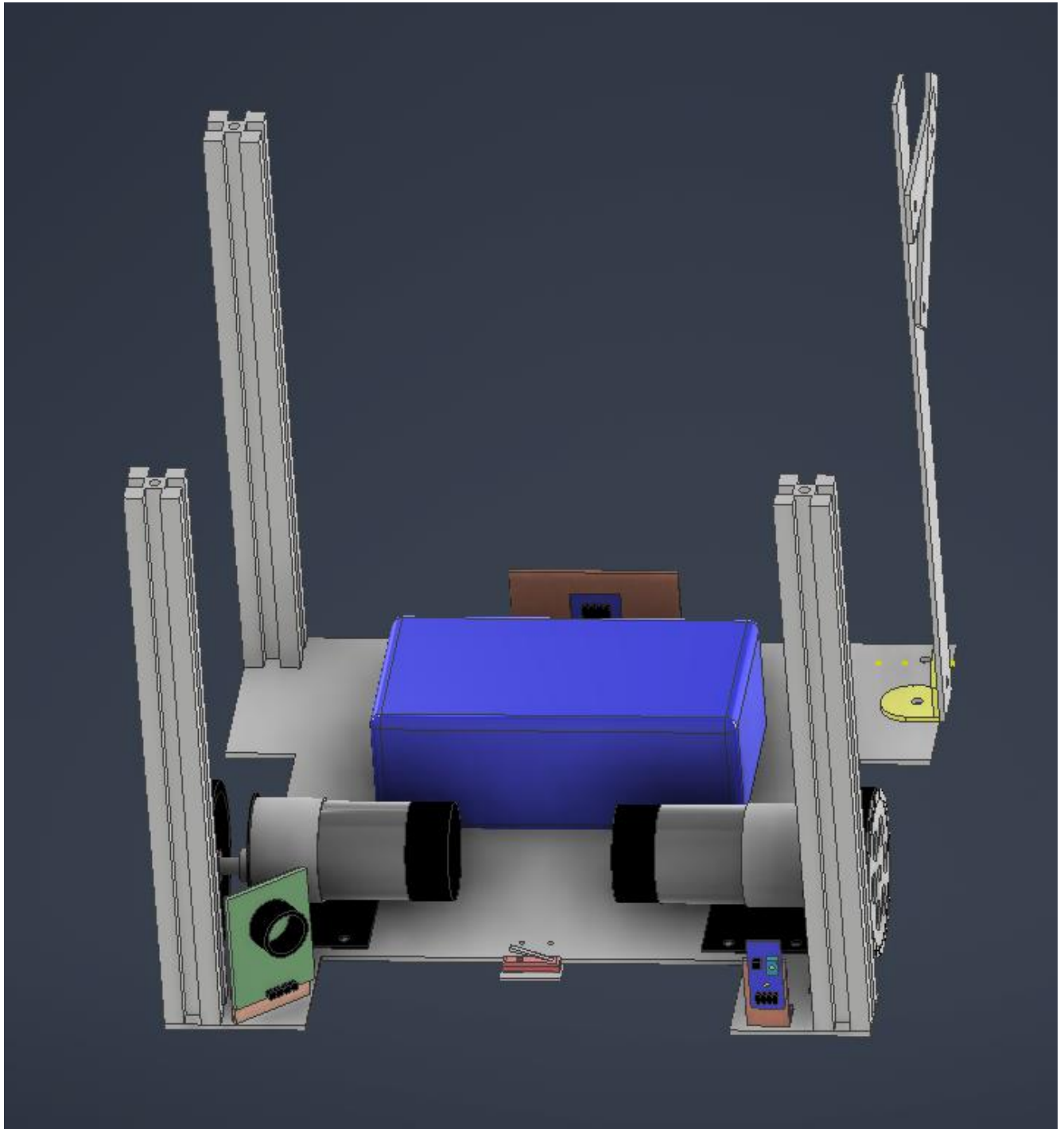
ALTURA MÁXIMA:



*Ilustración 81: verificación de altura*

Con lo que se concluye que se está dentro de la normativa, ya que la normativa decía que podía llegar hasta 375 mm con el botón de emergencia con lo cual cumplo con la normativa el robot está dentro del rango. Ni siquiera con el botón de emergencia que es lo que más espacio en altura ocupa ni siquiera estoy ni cerca de pasarme

PRIMERA PLANTA:



*Ilustración 82: Primera planta, vista 1*

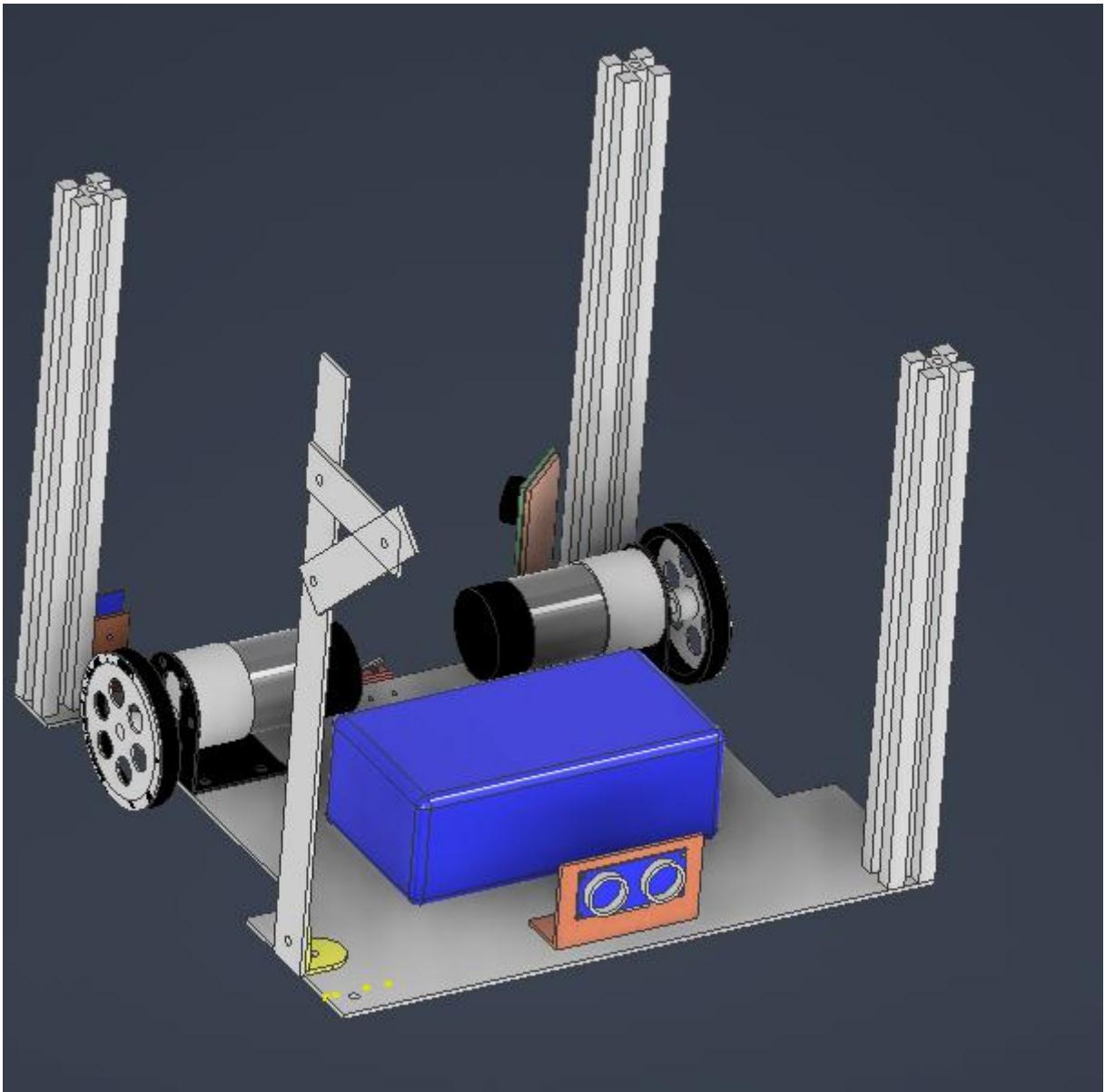
En esta primera planta he colocado como se puede apreciar la batería, que la he sujetado con bridas, luego he colocado los dos motores motrices junto con un soporte de acero inoxidable, que lo vendían el mismo fabricante que vendían los motores. Después he colocado un sensor de color con un soporte hecho de madera de 2 mm de grosor, realmente este sensor y en general los sensores no suelen pesar mucho, también en el lado derecho colocado un sensor de infrarrojo con el objetivo de poder detectar los barriles que hay que coger con las pinzas este es el apoyo para detectar los objetos

**Trabajo fin de grado**

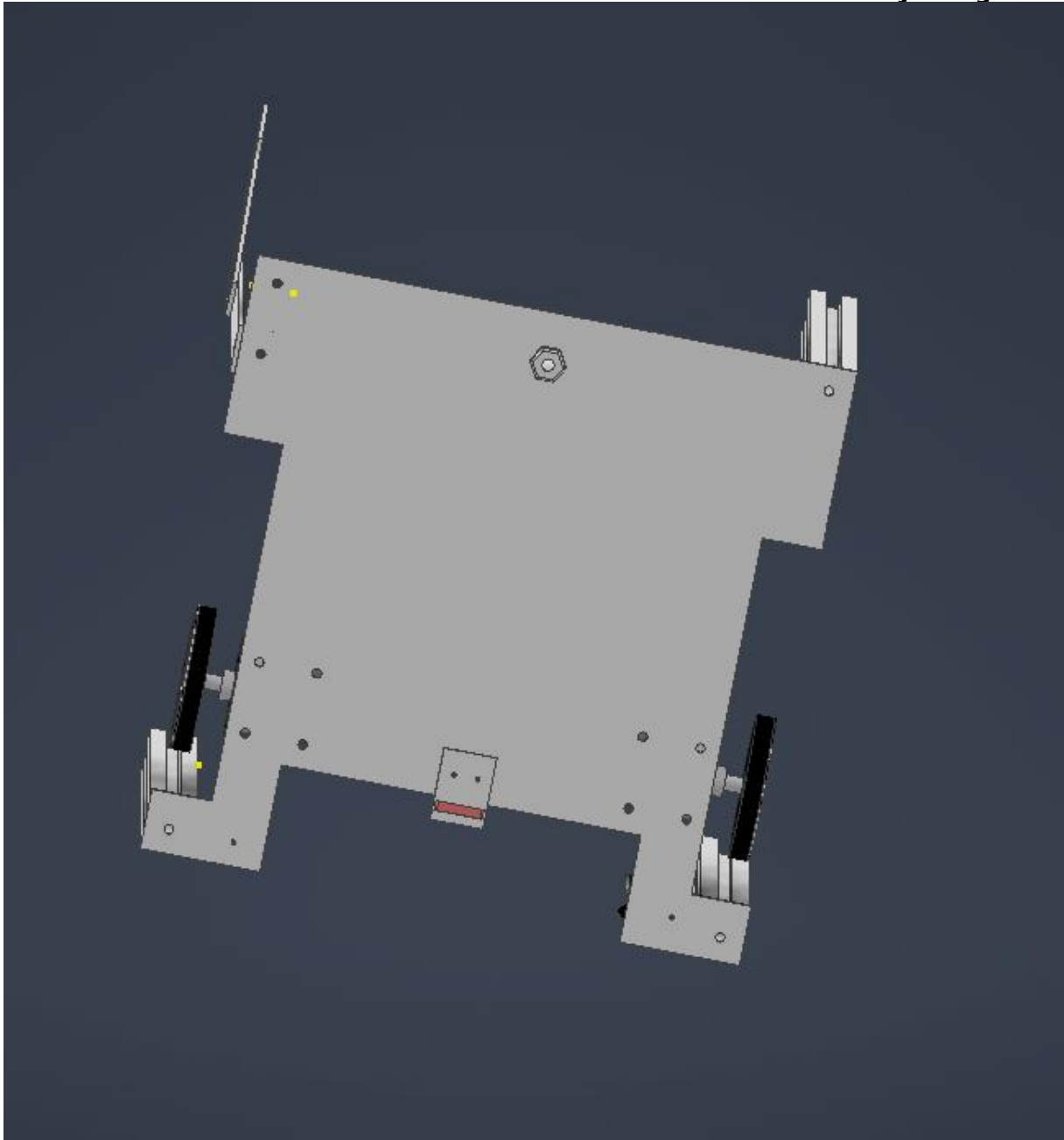
aparte de la Cámara. Luego en la parte trasera tenemos un sensor de ultrasonidos que lo que va a hacer es detectar la pancarta es decir si ve la pancarta es que el mensaje ya ha llegado al suelo entonces ya que le mandé un mensaje al servomotor de que se detenga, eso lo haría el microcontrolador.

Luego también se puede apreciar que hay 3 perfiles hechos de aluminio que son los que van a soportar las 3 planta, hay otro perfil que es una lámina de aluminio de 2 mm que ese tiene como objetivo soportara los ejes de plástico de la tercera planta es decir los engranajes, para que cuando se haga el giro tenga en donde apoyarse.

Los 3 perfiles de aluminio se van a sostener con unos tornillos que están ancladas por la parte de abajo a los 3 perfiles, y la lámina de aluminio va a sujetar con un soporte en forma de L de acero inoxidable

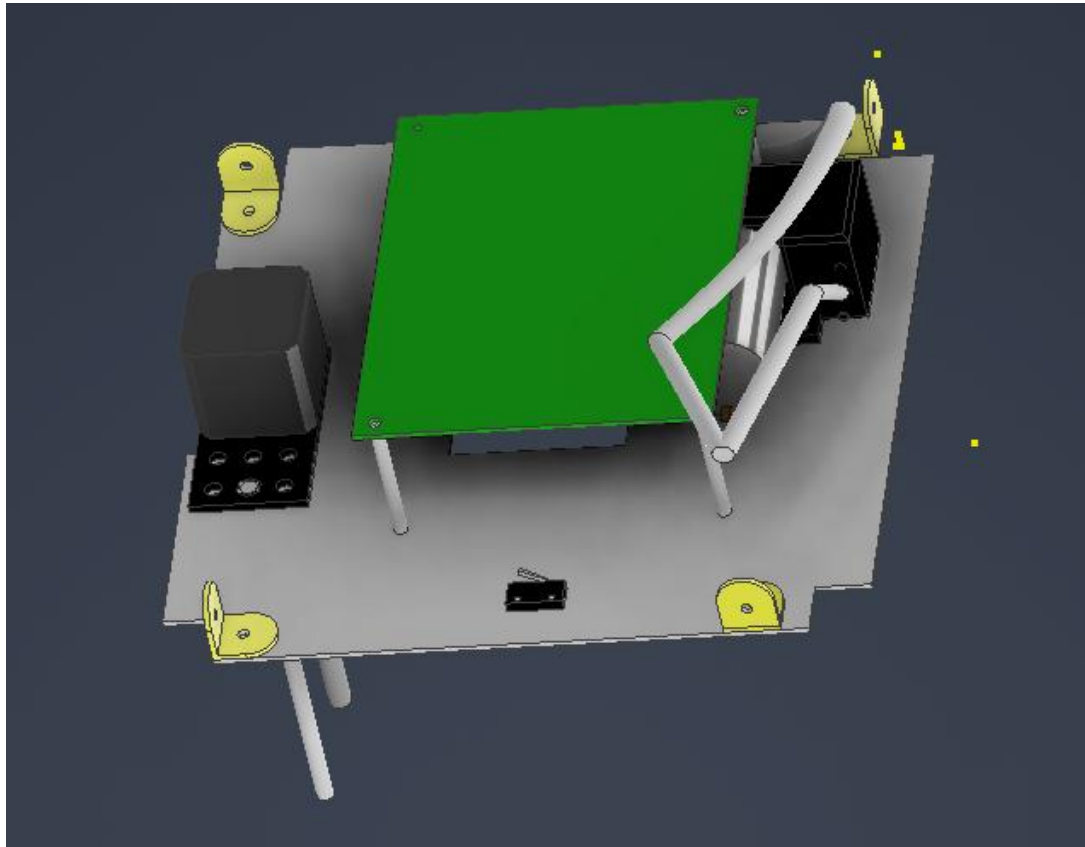


*Ilustración 83: Primera planta, Vista 2*

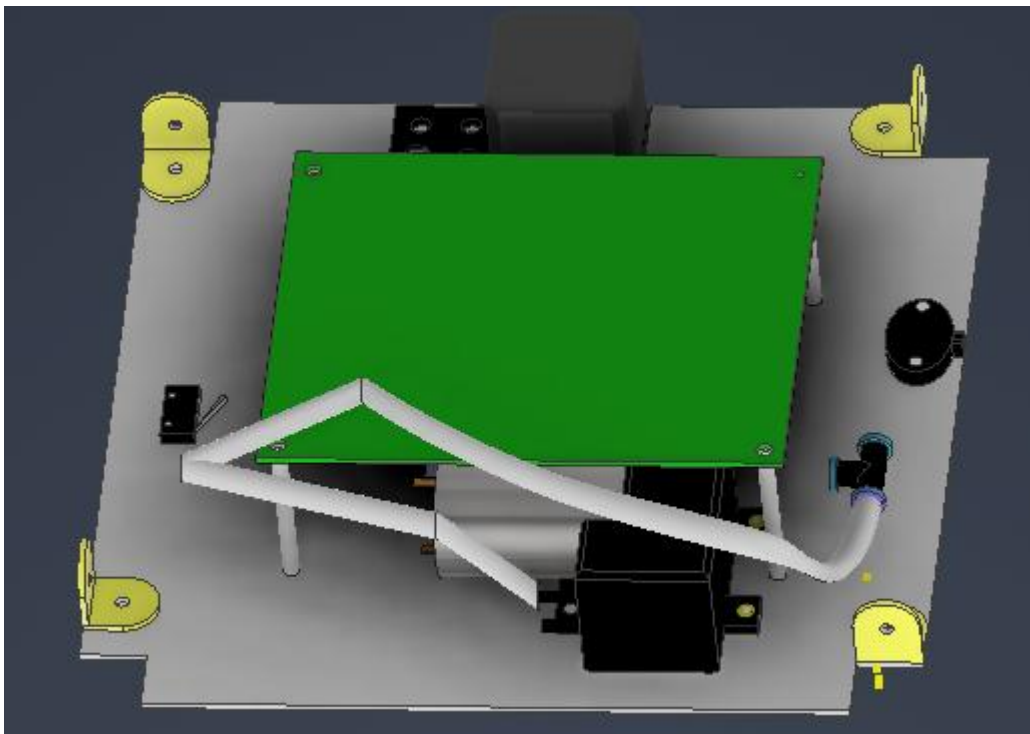


*Ilustración 84: Primera planta, vista inferior*

Las rueda de apoyo va atornillada a la base con un tornillo con tuerca y por último el final de carrera del sistema de la plataforma de la pinza, esta enganchado con unos tornillos, que se introducen por unos orificios que vienen incluidos en la placa base del final de carrera; he dejado un hueco para poder realizar la conexión y que no estorbe si se intentan pasar los cables por otro sitio



*Ilustración 85: Segunda planta, vista 1*



*Ilustración 86: Segunda planta, vista 2*

**Trabajo fin grado**

En primer lugar he decidido diseñar un soporte para la PCB y la puse de manera horizontal, Decidí ponerla con cuatro perfiles de madera en cada uno de sus extremos para sostenerla de manera horizontal, cogí madera ya que no puedo coger ningún tipo de metal ni nada parecido que soporte porque debido a la corriente muy peligroso y aparte de que la madera es fácil de agujerear.

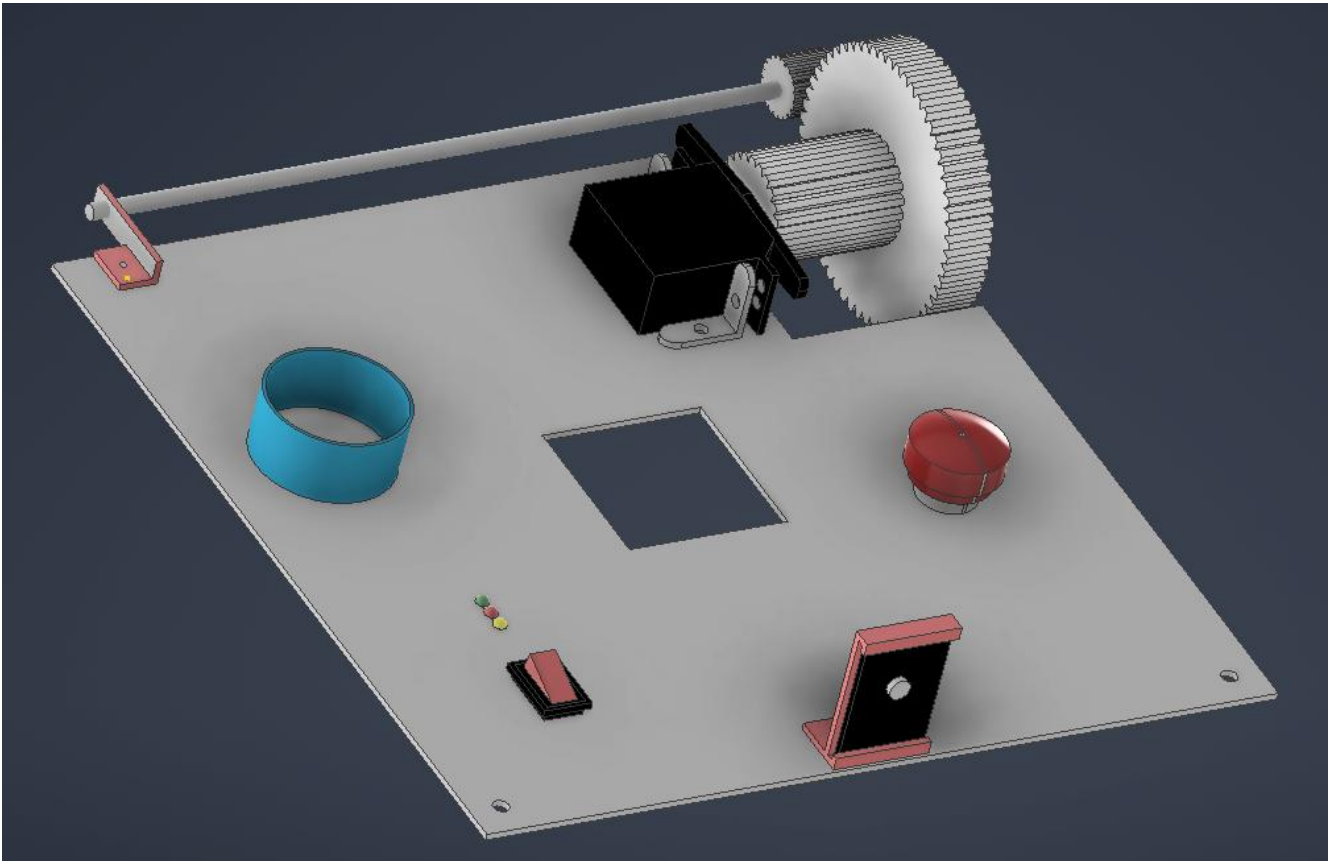
En segundo lugar he puesto una bomba de vacío y la sujetado con cuatro tornillos, la bomba vacío tiene un sistema de roscado ya de fabricación entonces solo tuve que atornillar la bomba a la segunda planta y ya está, anclarla con un tornillo de la métrica del roscado de la bomba. La bomba de vacío tiene un empalme en forma de T en donde va conectado un sensor de presión, que es el que le manda la información al micro para que active la bomba de succión.

Y por último tenemos un final de carrera del cordón de arranque, que nos permitirá enganchar el cordón para utilizarlo como cordón de arranque en mi robot.



*Ilustración 87: Segunda planta, parte inferior*

## TERCERA PLANTA:

*Ilustración 88: Tercera planta*

En la tercera planta tenemos a la cámara de visión artificial que está conectado en la esquina, para que pueda captar todos los objetos.

Luego tenemos el cubo que es donde los adversarios dejan los localizadores para saber la ubicación de los demás robot, este cubo viene anclado a la base con un tornillo y tuerca.

Además tenemos el seta de emergencia que vendrán colocado con un ajuste mecánico que viene incluido con el seta de emergencia y tiene los contactos normalmente abierto y normalmente cerrado, pero solo se utilizará el último.

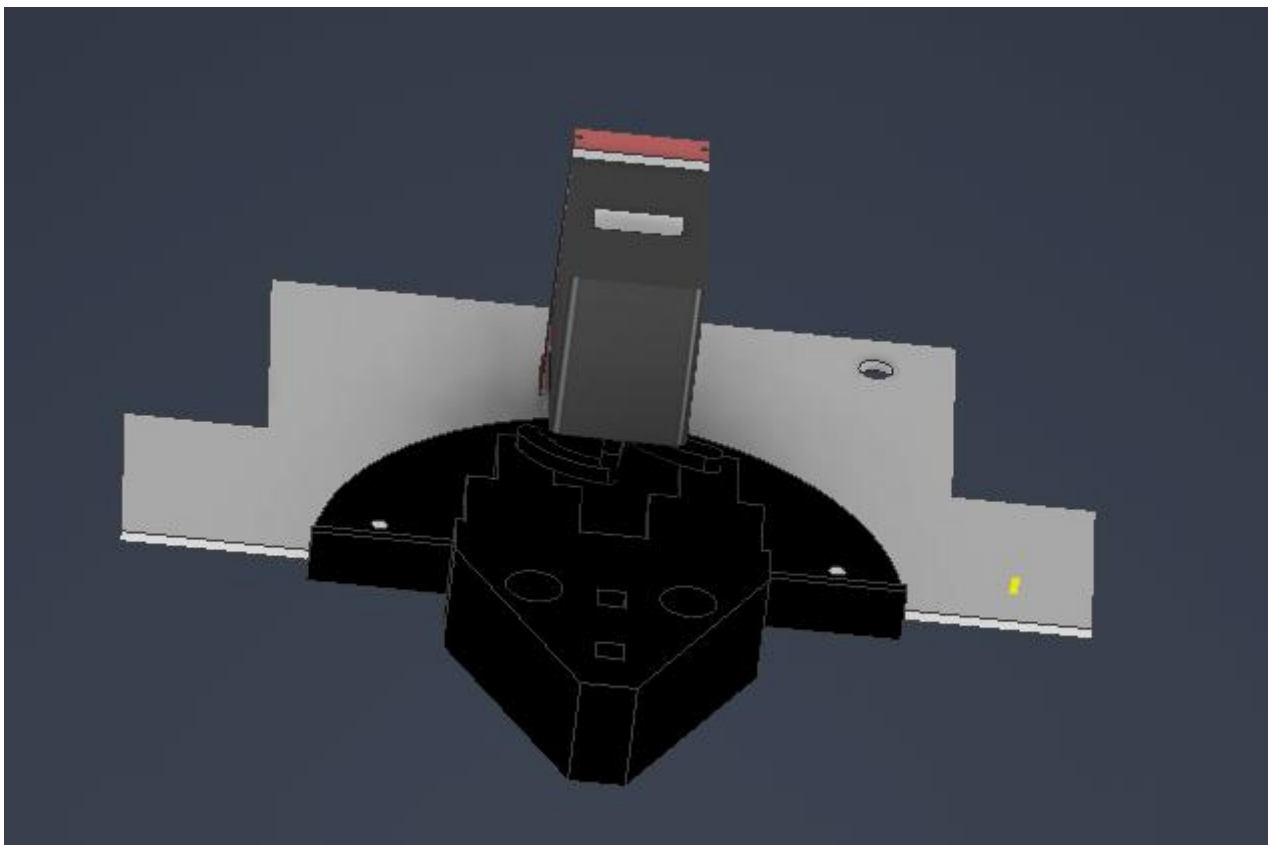
Para la indicación he colocado 3 indicadores de LED en forma de semáforo de esta manera se indicará los estados, estos LED está enganchado a un soporte e irán conectados a la base con cinta adhesiva de doble cara.

### Trabajo fin grado

He puesto un botón on/off incrustado a la placa de la planta, este botón cuenta con unas pequeñas lengüetas que sujetarán el botón en el sitio una vez puesto.

Por último tenemos un servomotor que está ensamblado con unos soportes en L y unos tornillos ya que el propio servo tiene unos orificios, en el extremo del servomotor hay un engranaje, en donde ese engranaje mueve otro engranaje el cual estás sujeto a la lámina de aluminio, ya que lo tenía que sujetar al extremo para que tuviera más estabilidad y así el momento no se inclinará más para un lado que para el otro y las fuerzas se distribuyan por eso decidí hacerlo de esta manera, sujetarlo por los dos extremos.

#### PLATAFORMA:

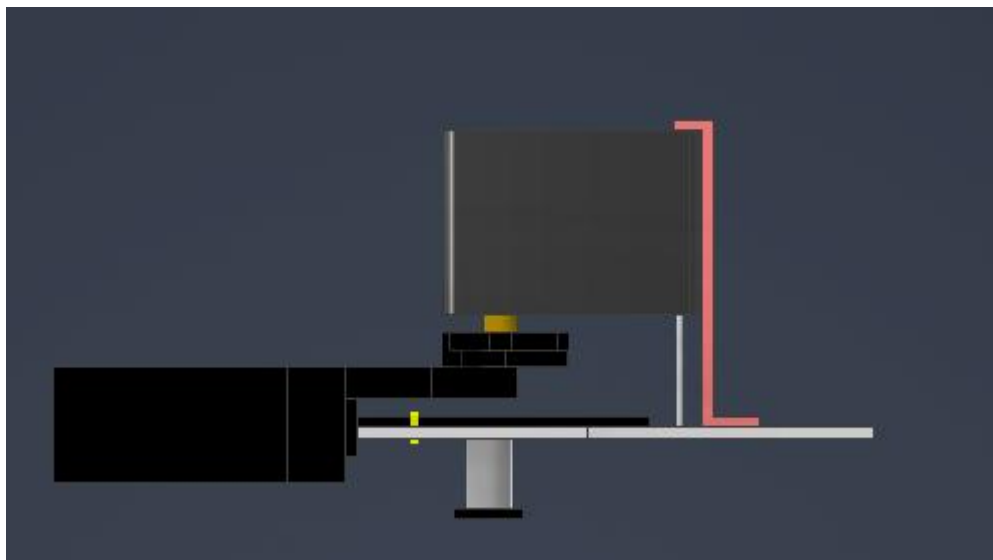


*Ilustración 89: Plataforma, vista 1*

en la plataforma tenemos una pinza que está atornillada por 5 tornillos de métrica 2, luego tenemos al servomotor que está anclado con un soporte de madera y con dos tornillos que están por la parte de abajo

**Trabajo fin de grado**

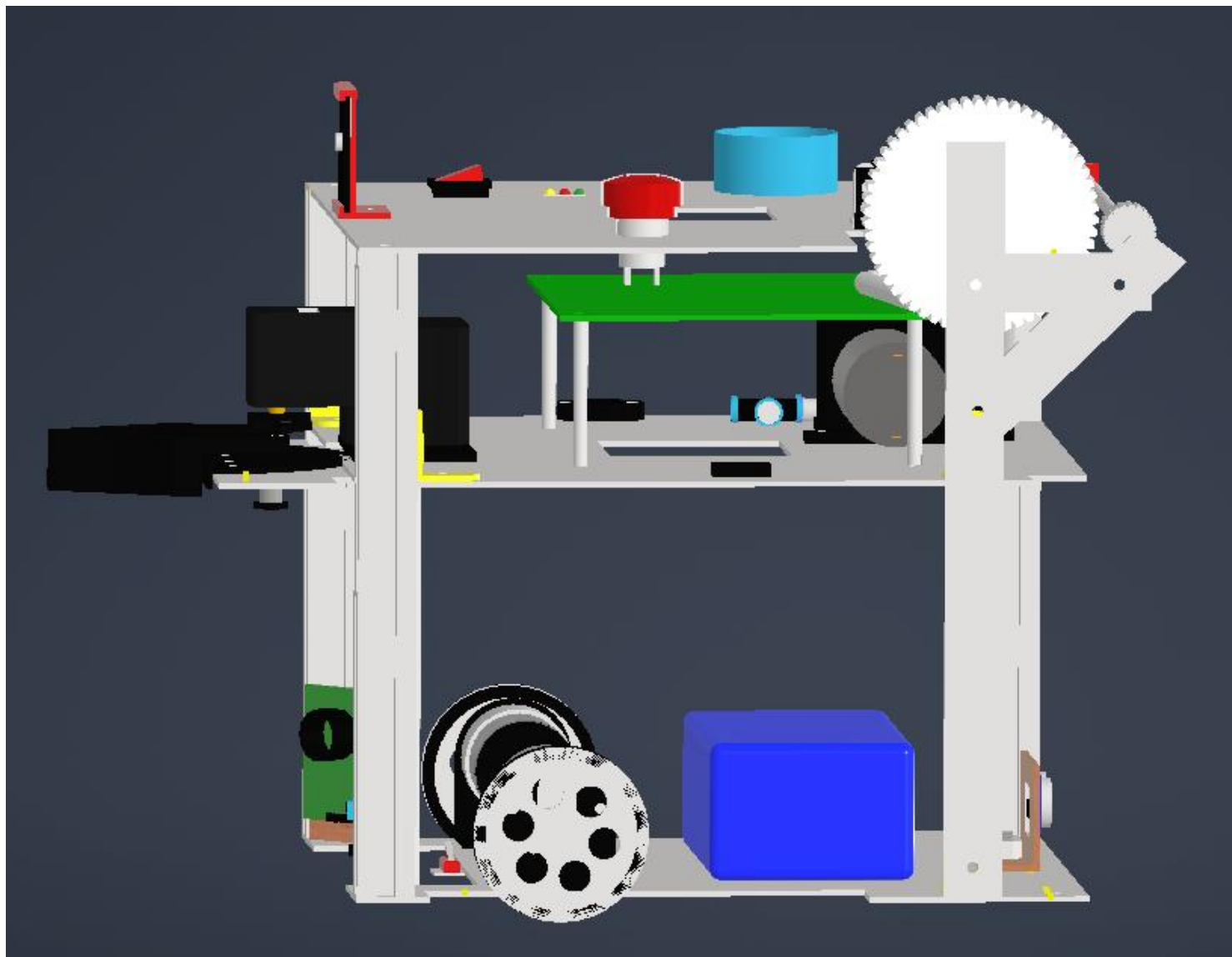
para soportar y darle más estabilidad para en caso de que haya movimientos bruscos no se mueva, y que solo se mueva la pinza.



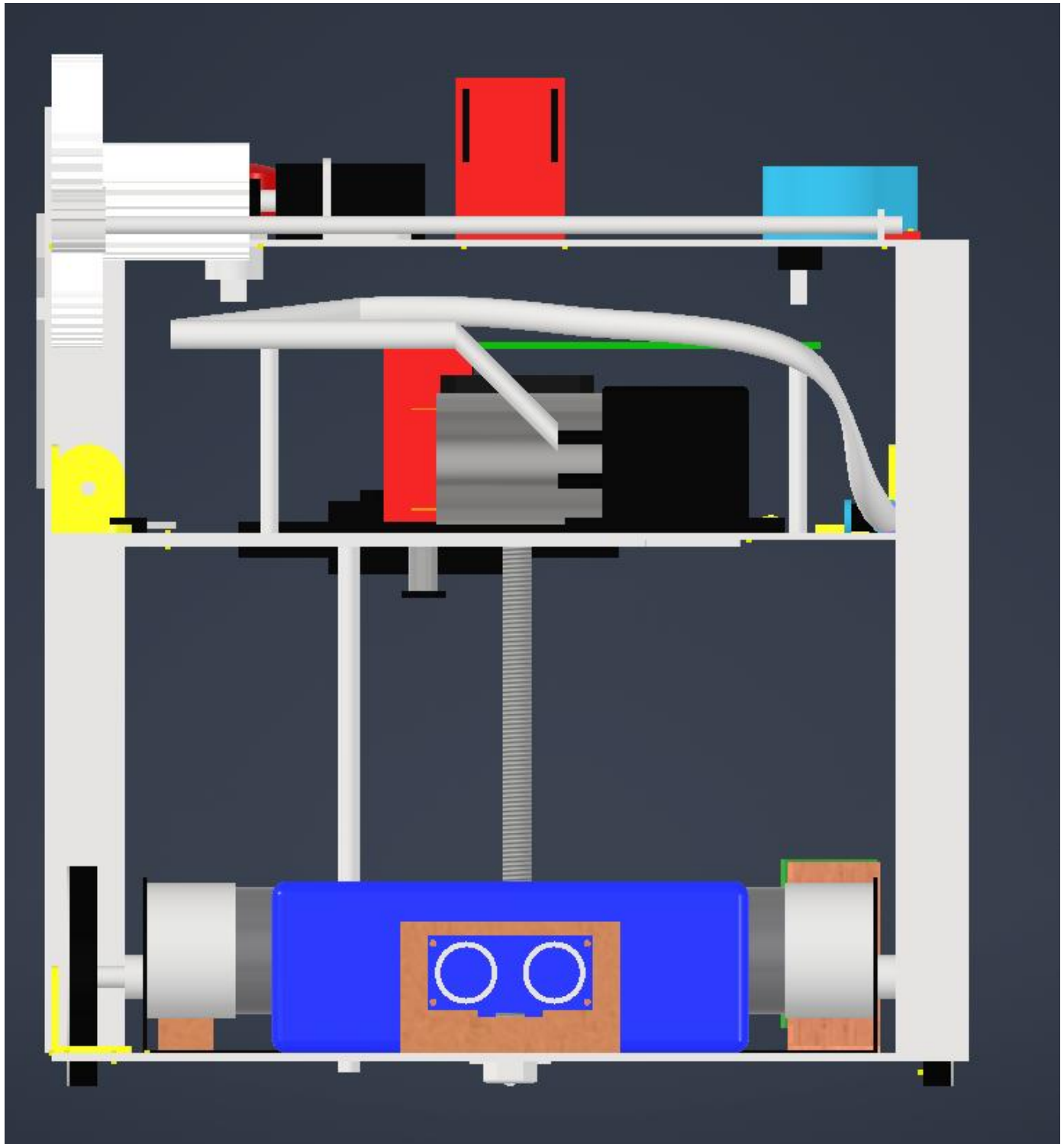
*Ilustración 90: Plataforma vista lateral*

Por otro lado tenemos la ventosa, que está puesta en la parte inferior y esta ventosa básicamente se pone a un nivel inferior de la pinza ya que cuando haya que agarrar una placa para hacer una capa o sea una segunda capa, tiene que poder agarrarla sin problema sin ningún estorbo y a la distancia a la que está puesta la pinza no va a estorbar en ningún momento a la ventosa ni la ventosa va a estorbar a la pinza. El tubo de silicona que va a la ventosa se coloca a través de un agujerito que he hecho en un sitio donde el sistema que abre y cierra las pinzas no le llegue a tocar.

ROBOT COMPLETO:



*Ilustración 91: Robot completo, vista 1*

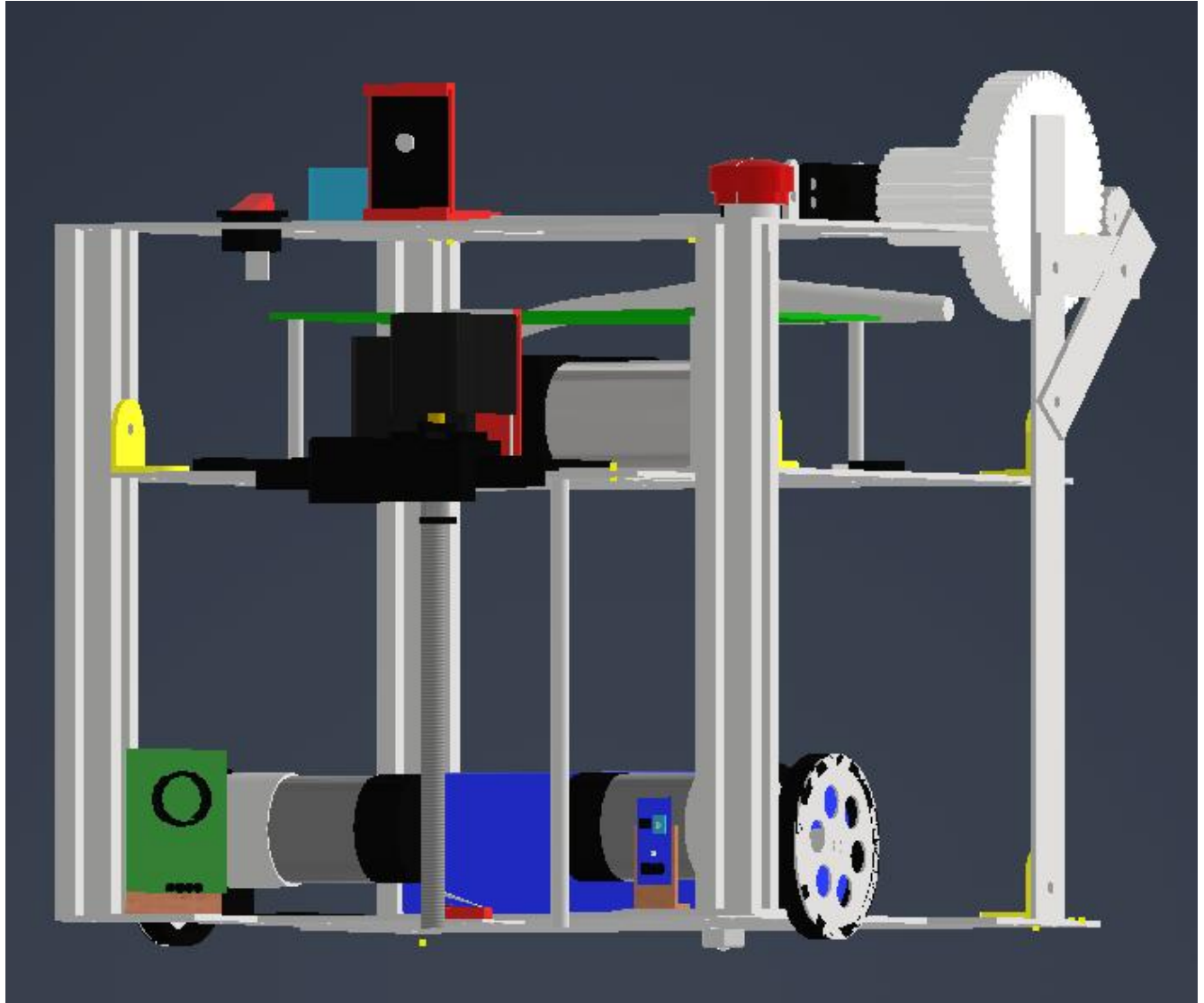


*Ilustración 92: robot completo, vista 2*

A la hora del despliegue de la pancarta la estrategia que he seguido es que el motor al inicio se acerque a la esquina del terreno de juegos y cuando él sensor de ultrasonidos no detecta nada, se inicie el despliegue de la pancarta, esto se realiza activando el servomotor que activa el sistema de engranajes. En el momento que el sensor de ultrasonido

### Trabajo fin grado

vuelva a detectar un objeto, o sea detecta la pancarta, en ese momento el servomotor ya se para, esto vendría a ser como una especie de final de carrera, pero sin serlo usando el sensor de ultrasonidos



*Ilustración 93: Tercera planta, parte delantera*

Para el sistema de manipulación lo que tenemos es un motor paso a paso que tiene anclado un husillo de diámetro 8, que lo que hace es subir y bajar la plataforma en donde se encuentra la pinza y la ventosa. La plataforma cuando baja nunca se va a salir del husillo ya que tiene un final de carrera en la parte inferior que hace que pare el motor paso a paso. Luego también tenemos una guía, eso lo que hace es para que la plataforma baje y suba de manera recta y nunca se salga de su camino. Otra cosa a recalcar es que en la segunda planta tenemos soportes en forma de L para soportar la segunda planta que se anclan a los perfiles de aluminio.

## 4.7. Esquema electrónico

Autor: Julio Cesar Balanta Narvaez

- 109 -

Trabajo de fin de grado

## 4.7.1. Fuente de alimentación

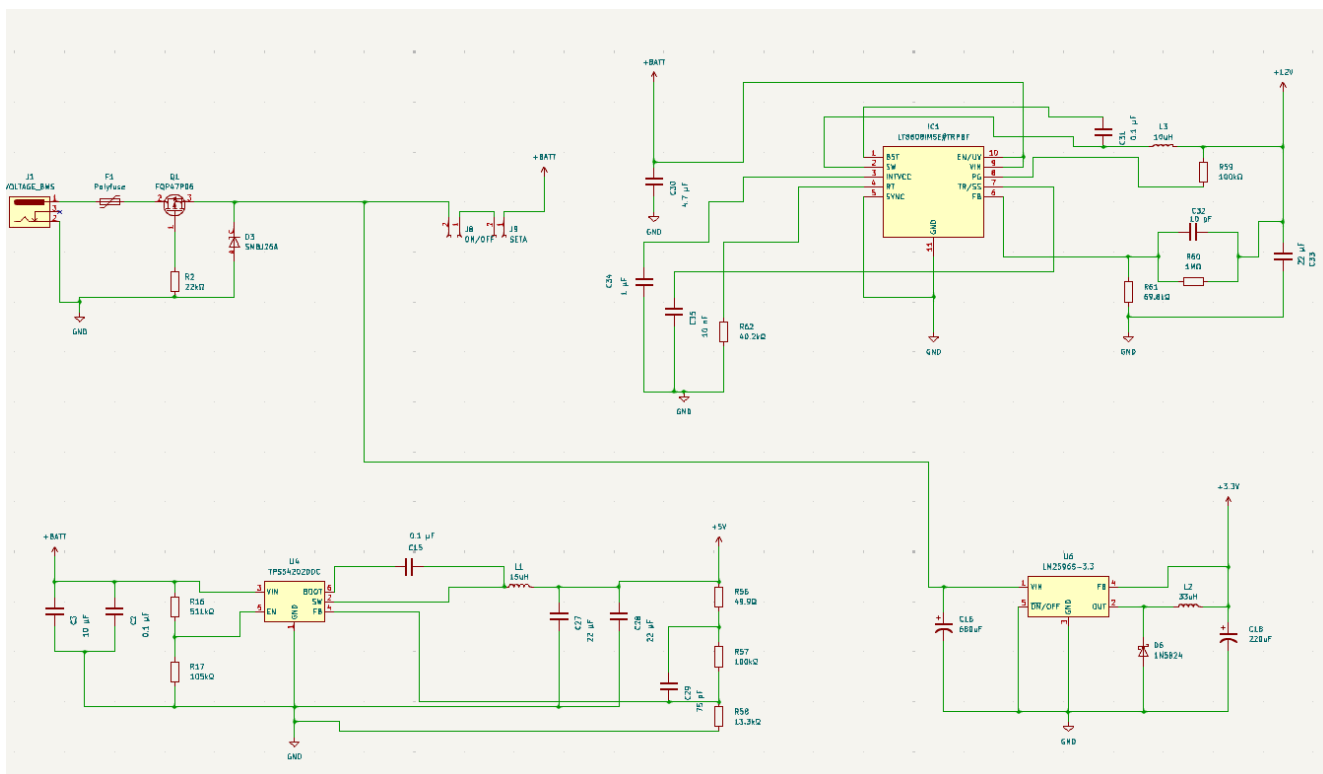


Ilustración 94: Fuente de alimentación

Para elegir la fuente de alimentación, debido a que he cogido una batería externa que tenía un conector barrel macho, me he decantado por coger un conector barrel hembra que puede aguantar una tensión nominal de hasta 24 voltios Y 5 amperios en corriente continua, corriente más que suficiente ya que no todos los componentes van a estar activos al mismo tiempo.

En primer lugar he puesto un fusible reseteable que al calentarse interrumpe la corriente por sobreintensidad, pero una vez que se enfrían se regeneran solos y vuelven a conducir, también son llamados polyfuse.

Después de eso he puesto un MOSFET que lo que hace es proteger contra la polaridad inversa, es decir, en caso de que nos equivoquemos en la conexión evita un cortocircuito.

Luego de ello tenemos un diodo TVS entre alimentación y masa, que absorbe los picos de tensión que puedan dañar los componentes, he cogido un TVS de 26V, solo actúa cuando la tensión excede este valor.

Por último la resistencia lo que hace básicamente es que me mantiene el MOSFET normalmente encendido, sin la resistencia el MOSFET quedaría flotando y podría encenderse o apagarse de manera errática, también asegura que en reposo la puerta se descarga a GND adoptando un estado definitivo.

$$R = \frac{V * t}{Qg} = \frac{24 * 165 * 10^{-6}}{180 * 10^{-6}} = 22k\Omega$$

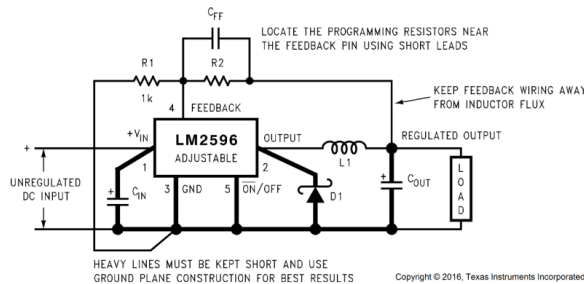
En donde el tiempo es el que queremos que se tarde en encender y apagar el MOSFET, Qg esta en el datasheet

Q <sub>g</sub>	Total Gate Charge	—	—	180
----------------	-------------------	---	---	-----

Después conecte un interruptor ON/OFF y una seta de emergencia. Antes del ON/OFF he sacado VBATT y lo he conectado directamente en el integrado (LM2596S-3.3) debido a que el microcontrolador si se le corta la corriente mediante la seta de emergencia, se puede rayar es decir deteriorar. También sí yo llegará a cortar la corriente al microcontrolador luego cuando lo volviera a encender perdería las coordenadas en las que estaba, no se sabría si agarró un objeto o el recorrido que llevaría, por eso he decidido no cortarle la corriente al micro y solo le corté la corriente a los demás componentes

Figure 9-11. Horizontal time base: 4 ps/div Figure 9-12. Horizontal time base: 100 ps/div

### 9.2.2 LM2596 Adjustable Output Series Buck Regulator



$$V_{OUT} = V_{REF} \left( 1 + \frac{R_2}{R_1} \right) \text{ where } V_{REF} = 1.23 \text{ V} \quad R_2 = R_1 \left( \frac{V_{OUT}}{V_{REF}} - 1 \right)$$

Select R<sub>1</sub> to be approximately 1 kΩ, use a 1% resistor for best stability. C<sub>IN</sub> — 470-μF, 50-V, Aluminum Electrolytic Nichicon *PL Series* C<sub>OUT</sub> — 220-μF, 35-V Aluminum Electrolytic, Nichicon *PL Series* D1 — 5-A, 40-V Schottky Rectifier, 1N5825 L1 — 68 μH, L38 R1 — 1 kΩ, 1% C<sub>FF</sub> — See [Section 9.1.2](#)

manufacturers data sheet for this information.

#### 9.2.2.2.4 Feedforward Capacitor (C<sub>FF</sub>)

See [Table 9-6](#).

For output voltages greater than approximately 10 V, an additional capacitor is required. The compensation capacitor is typically between 100 pF and 33 nF, and is wired in parallel with the output voltage setting resistor, R<sub>2</sub>. It provides additional stability for high output voltages, low input or output voltages, or very low ESR output capacitors, such as solid tantalum capacitors. Calculate the value for C<sub>FF</sub> with [Equation 10](#):

$$C_{FF} = \frac{1}{(31)(10^3)(R_2)} \quad (10)$$

This capacitor type can be ceramic, plastic, silver mica, and so forth. Because of the unstable characteristics of ceramic capacitors made with Z5U material, TI does not recommend them.

[Table 9-6](#) contains feedforward capacitor values for various output voltages. In this example, a 560-pF capacitor is required.

Para el cálculo de este integrado el convertidor de VBATT a 3.3 V, vemos que el fabricante nos especifica de que cuando el voltaje de salida es mayor a 10 voltios se requiere este condensador adicional, como en nuestro caso queremos 3.3 voltios no nos haría falta este condensador.

Trabajo fin de grado

$$R2 = R1 * \left( \frac{VOUT}{VREF} - 1 \right) = 1K\Omega * \left( \frac{3.3V}{1.23V} - 1 \right) = 1.69K\Omega$$

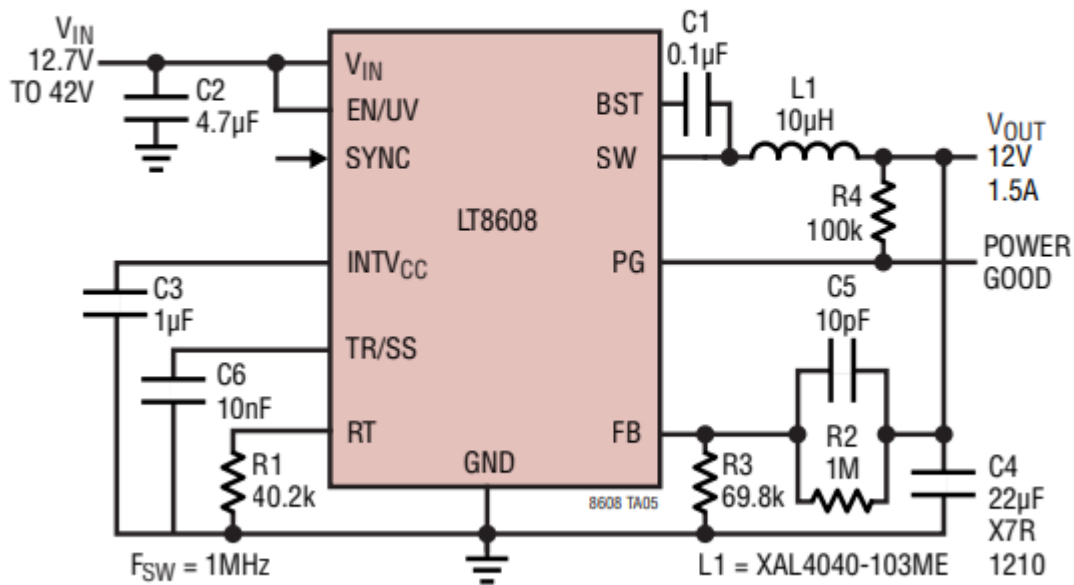
Y este sería nuestro convertidor y con el que alimentaríamos el microcontrolador y algún que otro sensor adicional.

La cosa de esto es que hay un modelo LM2596S-3.3, que ya incluye la R1 y R2 internamente, que fue el que tome y puse en el esquemático.

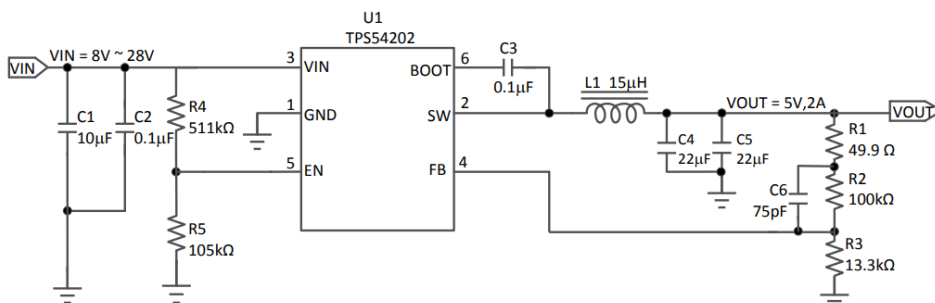
El convertidor LT8608IMSE#TRPBF, Qué es el convertidor que convierte de VBATT a 12 V, que tienen una corriente de salida de hasta 1.5 amperios más que suficientes en mi caso, solo lo usare para la bomba de vacío y el paso a paso que en condiciones normales no lo superan.

si conecto SYNC a 0V según el fabricante de hasta el 88%.

12V Step Down



El convertidor TPS54202DDC te transforma de manera directa VBATT voltios en 5 voltios, ala salida proporciona 5V y da 2A mas que suficiente, ya que los servomotores y sensores no superan esto.



## 4.7.2. Driver motor paso a paso

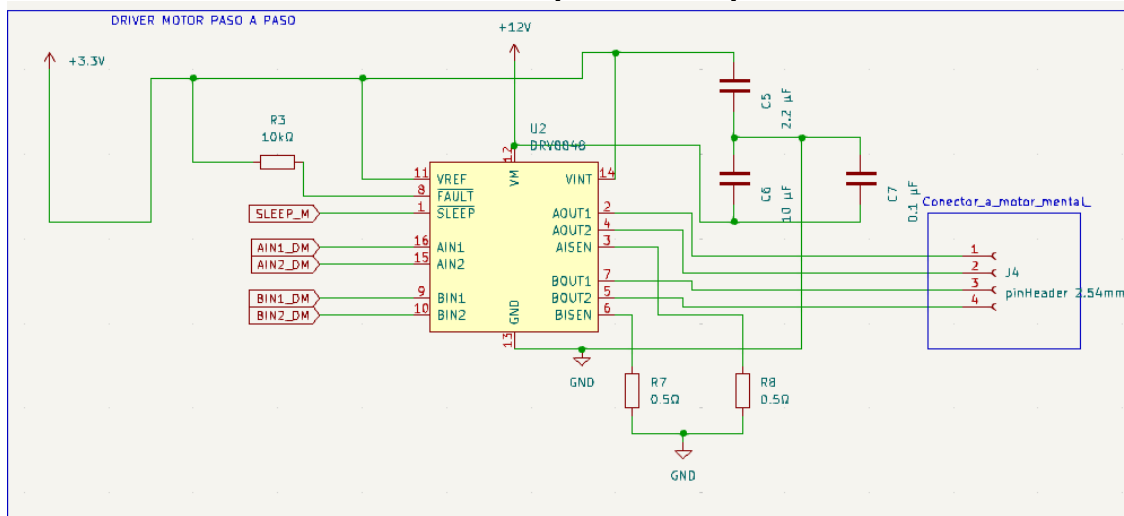


Ilustración 95: DRV8848 Kicad

Esta conexión fue sacada directamente del fabricante, aquí lo que hay que tener en cuenta en primer lugar sería la corriente que nos puede suministrar este driver

### 5.3 Recommended Operating Conditions

		MIN	MAX	UNIT
$V_{VM}$	Power supply voltage range <sup>(1)</sup>	4	18	V
$V_{VREF}$	Reference rms voltage range <sup>(2)</sup>	1	3.3	V
$f_{PWM}$	Applied INPUT Signal	0	250	kHz
$I_{VINT}$	VINT external load current		1	mA
$I_{rms}$	Motor rms current per H-bridge <sup>(3)</sup>	0	1	A
$T_A$	Operating ambient temperature	-40	85	°C

(1) Note that  $R_{DS(ON)}$  increases and maximum output current is reduced at VM supply voltages below 5V.

(2) Operational at VREF between 0 and 1V, but accuracy is degraded.

(3) Power dissipation and thermal limits must be observed.

es decir este motor se alimenta en mi caso lo he puesto a 12 voltios y proporciona una corriente de hasta un amperio

Electrical Characteristics	
1. Number of Phase	2
2. Step Angle	1.8 °
3. Rated Voltage	3.3V DC
4. Rated Current	1.5 A
5. Holding Torque	400mN.m Min(2 Phases)
6. Phase Resistance	2.2 Ω ± 10% (20° C)
7. Phase Inductance	3.8mH±20%(1kHz 1V rms)
8. Motor Inertia	57 g.cm <sup>2</sup>
9. Motor Weight	280g Ref.
10. Insulation Resistance	100MΩ Min.(DC 500V)
11. Insulation Class	B(130° C)

Si lo comparamos con el motor paso a paso que he cogido el nema 17, tenemos que el driver nos proporciona 1 amperio y este motor puede trabajar hasta con 1.5 A. Como en este proyecto realmente lo que tenemos que levantar no es mucho peso (no hay mucho esfuerzo) este driver sirve, aunque no se utilizará todo el par máximo del motor pero no es necesario.

Trabajo fin de grado

Luego tenemos el voltaje de alimentación, según dice en el datasheet del nema 17 el motor se alimenta a 3.3 V. Este driver según las especificaciones técnicas se puede alimentar de 4 a 18 V, este driver es un driver del tipo chopper, es decir, que permite usar una tensión de alimentación mayor a la nominal del motor, y eso lo que hace es que ayuda a que la corriente suba más rápido en las bobinas. Al usar una tensión de alimentación de 12 voltios el motor va a responder mejor a grandes velocidades. Por eso aplica una tensión mayor para forzar que la corriente suba más rápido en la bobina y cuando llega a su límite la recorta porque siempre va a estar limitado por la corriente.  $V = I * R$

5.1 Absolute Maximum Ratings

over operating free-air temperature range referenced with respect to GND (unless otherwise noted)<sup>(1)</sup>

		MIN	MAX	UNIT
	Power supply voltage (VM)	-0.3	20	V
	Power supply voltage ramp rate (VM)	0	2	V/μs
	Internal regulator voltage (VINT)	-0.3	3.6	V
	Analog input pin voltage (VREF)	-0.3	3.6	V
	Control pin voltage (AIN1, AIN2, BIN1, BIN2, nSLEEP, nFAULT)	-0.3	7	V
	Continuous phase node pin voltage (AOUT1, AOUT2, BOUT1, BOUT2)	-0.3	V <sub>VM</sub> + 0.6	V
	Continuous shunt amplifier input pin voltage (AISEN, BISEN) <sup>(2)</sup>	-0.6	0.6	V
	Peak drive current (AOUT1, AOUT2, BOUT1, BOUT2, AISEN, BISEN)	Internally limited		A
T <sub>J</sub>	Operating junction temperature	-40	150	°C
T <sub>stg</sub>	Storage temperature	-65	150	°C

Y otro dato que no se tiene en cuenta es el voltaje de entrada que puede aguantar este microcontrolador, el ESP32 envía pulsos de 3.3V con lo cual está dentro del rango permitido.

Control pin voltage (AIN1, AIN2, BIN1, BIN2, nSLEEP, nFAULT)	-0.3	7	V
--	------	---	---

<https://www.ti.com/lit/an/slva504a/slva504a.pdf>

A continuación, voy a calcular la potencia de disipación, para ver si este driver es capaz de aguantar, según este ejemplo proporcionado por Texas instruments, en donde se debe elegir la opción H-Bridge, ya que en este caso estamos trabajando con un puente de H dual.

f <sub>PWM</sub>	Applied INPUT Signal	0	250	kHz
t <sub>RISE</sub>	Output rise time	60		ns
t <sub>FALL</sub>	Output fall time	60		ns

$$PRON = RON * IL^2 * 2 \rightarrow 900M\Omega * 1^2A * 2 = 1.8W$$

$$PSW = VM * IL * (trise + tfall) * FPWM \rightarrow 12 * 1 * (60 * 10^{-9} + 60 * 10^{-9}) * 250000Hz \rightarrow 0.36W$$

$$PTOT = PRON + PSW \rightarrow 1.8W + 0.36W \rightarrow 2.16W$$

$$TJ = TA + (PTOT * R\theta JA) \rightarrow 25^{\circ}C + \left(2.16W * 40.3^{\circ}\frac{C}{W}\right) \rightarrow 112.1^{\circ}C$$

T <sub>J</sub>	Operating junction temperature	-40	150	°C
----------------	--------------------------------	-----	-----	----

112.1 < 150

Usando los valores exactos del datasheet y teniendo en cuenta el tipo de motor nos da que la temperatura de Unión se mantiene por debajo de 150°C, con lo cual este driver puede aguantar más que suficientemente a este motor paso a paso nema 17

### 4.7.3. Driver motores motrices

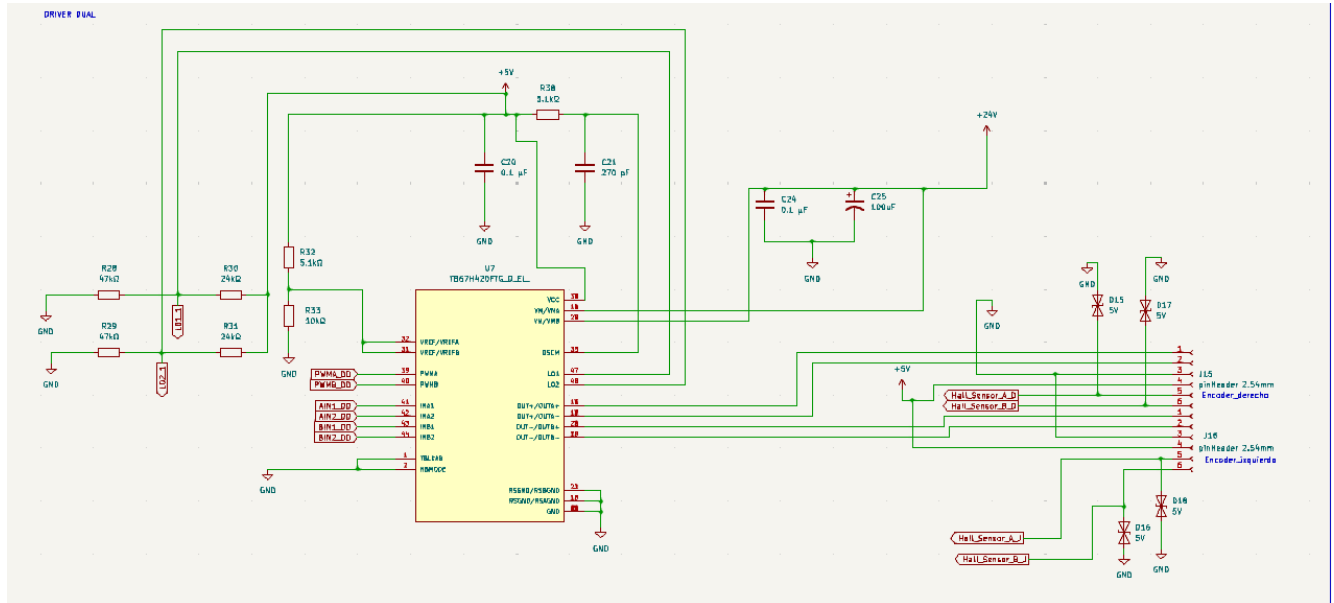


Ilustración 96: TB67H420FTG Kicad

Para ver si este driver aguanta el motor lo primero en que nos fijaremos es en la corriente

#### Operation ranges (Ta=-20 to 85°C)

Characteristics	Symbol	Min	Typ.	Max	Unit	Remarks
Motor power supply	VM	10	24	47	V	—
Motor output current	IOUT(SHB)	—	4.5	9.0	A	HBMODE=High (Note1)
	IOUT(DHB)	—	2.25	4.5	A	HBMODE=Low (Note2)
LO output pin voltage	VLO	—	3.3	VCC	V	—
Chopping frequency	fchop(range)	40	70	150	kHz	—
VREF input voltage	VREF	GND	2.0	4.0	V	—

Como podemos ver este driver en condiciones normales puede proporcionar una corriente de 2.25 A, y en condiciones de bloqueo o de esfuerzo hasta 4.5

amperios le proporcionaría a cada motor.

Rated Voltage	Pololu Item #	Gear Ratio	No Load		At Maximum Efficiency				Max Power	Stall Extrapolation <sup>(2)</sup>		Graph Pages
			Speed	Current	Speed	Torque	Current	Output		Torque	Current	
			RPM	A	RPM	kg-mm	A	W	W	kg-mm	A	
12 V	4750 <sup>(1)</sup>	1	10,000							5		
	4747, 4757	6.25	1600		1300	4.9	1.2	6.4	12	30		5, 6
	4748, 4758	10	1000		850	6.6	0.91	5.7	12	49		7, 8
	4741, 4751	18.75	530		470	10	0.76	5.0	12	85		9, 10
	4742, 4752	30	330	0.2	280	18	0.78	5.1	12	140		11, 12
	4743, 4753	50	200		180	22	0.66	4.0	10	210	5.5	13, 14
	4744, 4754	70	150		130	32	0.68	4.2	10 <sup>(3)</sup>	270		15, 16
	4745, 4755	102.08	100		87	42	0.72	3.8	8 <sup>(3)</sup>	340		17, 18
	4746, 4756	131.25	76		66	60	0.74	4.1	6 <sup>(3)</sup>	450		19, 20
	2828, 2829	150	67		58	65	0.72	3.8	6 <sup>(3)</sup>	490		21, 22
24 V	4690 <sup>(1)</sup>	1	10,000							5.5		
	4688, 4698	6.25	1600		1300	5.5	0.58	7.4	14	35		23, 24
	4689, 4699	10	1000		850	7.5	0.49	6.6	14	65		25, 26
	4681, 4691	18.75	530		450	13	0.49	6.1	13	95		27, 28
	4682, 4692	30	330	0.1	280	19	0.46	6.5	13	150	3.0	29, 30
	4683, 4693	50	200		170	27	0.41	4.9	12	230		31, 32
	4684, 4694	70	140		120	39	0.42	5.0	10 <sup>(3)</sup>	310		33, 34
	4685, 4695	102.08	100		86	51	0.42	4.5	8 <sup>(3)</sup>	390		35, 36
	4686, 4696	131.25	79		68	63	0.40	4.4	6 <sup>(3)</sup>	470		37, 38
	4687, 4697	150	68		59	73	0.41	4.4	6 <sup>(3)</sup>	560		39, 40

Como podemos ver la corriente total del motor es de 3 A, pero eso hablamos en el peor de los casos, si el motor no tiene cargas trabaja a una corriente de

Trabajo fin de grado

10 mA. Este driver a diferencia del driver del motor paso a paso no tiene una función que proporcione la corriente de manera activa, la corriente que suministra al motor depende directamente de la carga que tiene que mover. Si la carga es baja entonces la corriente es baja como se va a mover por una zona rectas sin empinaduras no habrá mucho esfuerzo.

El robot debe de pesar menos de 10 kg, pero pongamos que pesa 10 kg entonces quedaría de esta manera.

$\frac{10kg}{3} = 3.33kg$  Ya que tenemos dos ruedas y una rueda de apoyo se distribuye el peso

$$F = 3.3 * 9.81 = 32.67N$$

Par motor:

$$T = F * R \rightarrow 32.67N * 0.03m \rightarrow 0.98N * m \rightarrow 10kg * cm$$

Basándonos en los datos del datasheet de los motores vemos que esto se encuentra más cerca del par máximo de eficiencia que del par máximo de parada  $2.7kg*cm < 10kg*cm < 23kg*cm$

Tenemos que

- máxima eficiencia el motor produce  $2.7kg*cm$  y consume  $0.41A$
- máximo par de parada el motor produce  $23kg*cm$  y consume  $3A$

Corriente estimada con carga:

$$I_{estimada} = \left( \frac{10kg * cm}{23kg * cm} \right) * 3A \rightarrow 1.3A \text{ por cada motor}$$

Entonces en función del peso me sale que  $1.3A$ , y este driver proporciona corriente más que suficiente.

Ahora voy a calcular la disipación

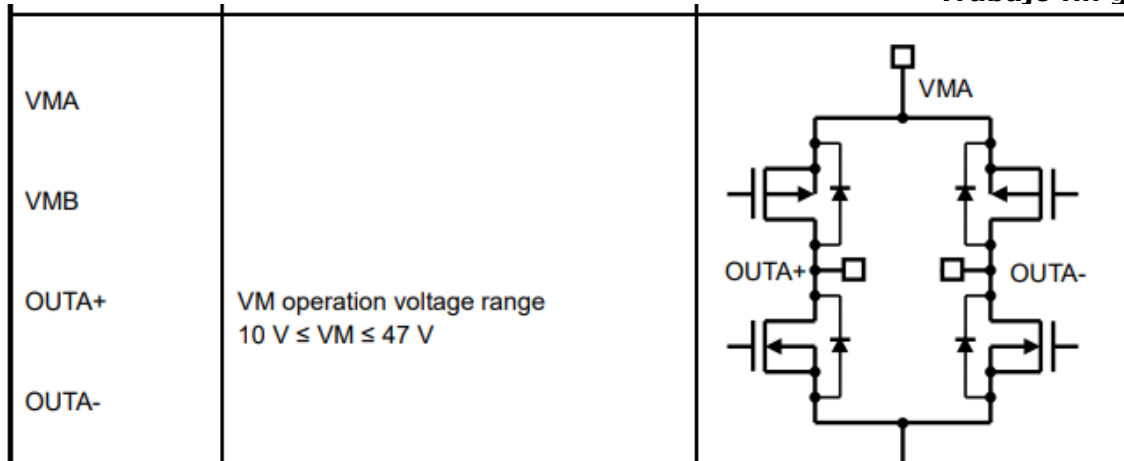
$$PRON = RON * I^2 * 2 \rightarrow (0.33\Omega * 1.3^2) * 2 \rightarrow 1.12W$$

$$PTOT \approx PRON = 1.12W$$

$$TJ = TA + (PTOT * R\theta JA) \rightarrow 25^{\circ}C + \left( 1.12W * 25^{\circ} \frac{C}{W} \right) \rightarrow 53^{\circ}C$$

Operating temperature	Topr	-20 to 85	°C	—
Storage temperature	Tstg	-55 to 150	°C	—
Junction temperature	Tj(max)	150	°C	—

$53^{\circ}C < 150$ , con lo cual en estas condiciones determinadas debido al peso que tiene que mover cada motor lo cual da una corriente específica, tenemos que la temperatura de Unión se mantiene muy por debajo, con lo cual este driver aguanta sin problemas a este motor



Este driver puede aguantar el voltaje máximo del motor para que funcione a su máximo rendimiento sin problemas ya que se puede alimentar con una tensión de entre 10 y 47V, y el motor es de máximo 25.2 V en su máxima eficiencia, este driver de corriente continua (brushed DC) y no limita la corriente, en su lugar controla la velocidad y la dirección del motor mediante la señal PWM. Es decir a diferencia del driver del motor paso a paso este consume la corriente que necesita consumir en función de la carga que debe mover, con lo cual aquí sí que se le debe de poner VBATT.

#### 4.7.4. Indicadores Leds

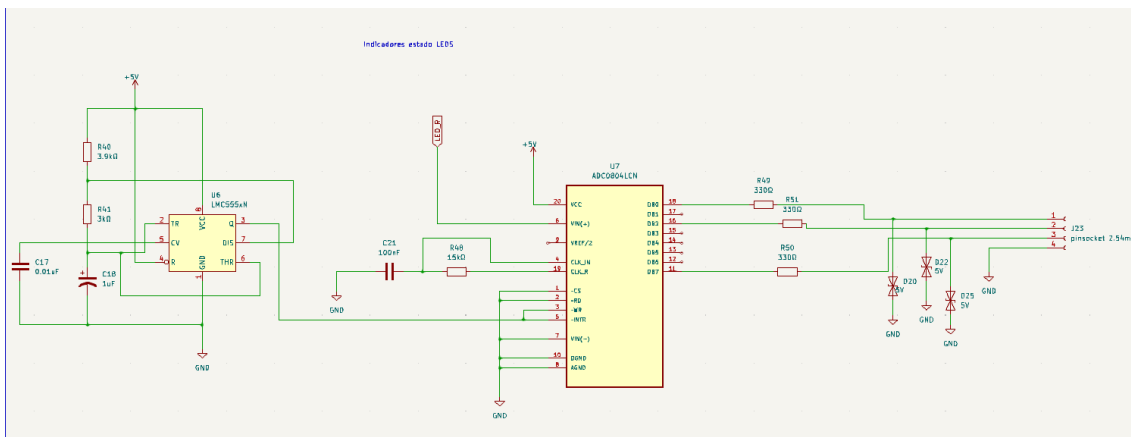
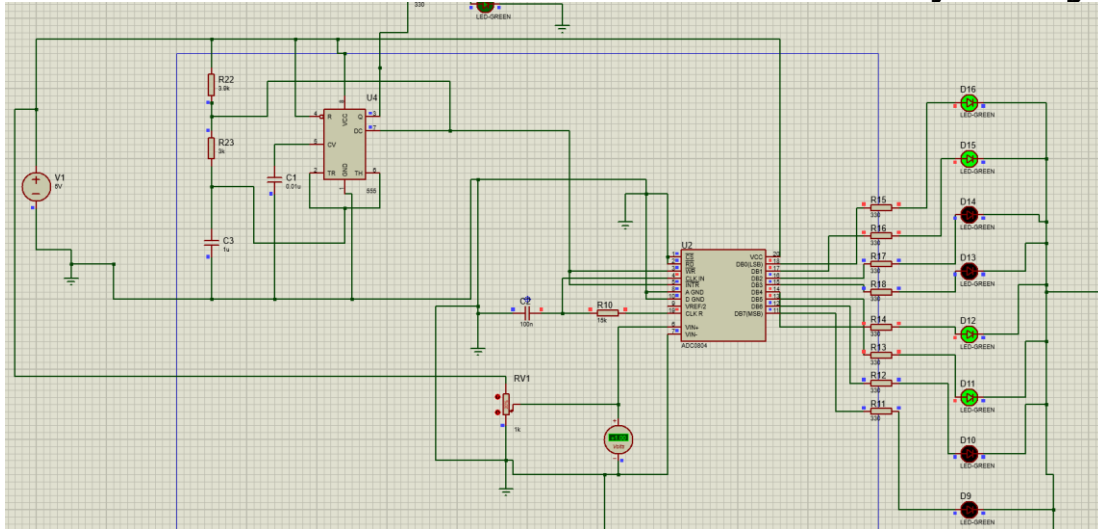


Ilustración 97: Codificador y señal de pulso

Para crear este circuito lo que he hecho es buscar un convertidor analógico digital, básicamente lo que quiero hacer es con un solo pin de entrada enviarle voltaje en función de cada estado del LED, este convertidor tiene una salida de 8 bit digitales y en este caso mi propósito sería por ejemplo enviarle un determinado voltaje y que ese voltaje a la salida me envíe una señal que me encienda un LED y que me muestre el estado del robot. Para esto he utilizado un simulador llamado proteus.



El funcionamiento es sencillo y lo tengo que mandar un valor ADC por el micro, escogí 3 valores correspondiente a los 3 estados  
 led verde → ENCENDIDO  
 led rojo → ERROR  
 Led amarillo → BATERIA BAJA

Analizando todas las salidas digitales que se encendían con estos 3 valores de voltaje decidí tomar 3 salidas que me venían bien o sea que se encendían y que no coincidían unas con otras, La lógica es más o menos así:  
 1V → enciende led verde y apaga los otros  
 2V → enciende led rojo y apaga los otros  
 3V → enciende led verde y led amarillo y apaga el rojo

Por eso elegido en concreto esas 3 salidas que se ven en el diagrama porque son las que más o menos me venían bien para lo que quería aplicar, y elegido un 555 timer, que es un temporizador que lo que hace es que genera una señal de reloj ya que este codificador necesita señales de pulsos para poder activarse.

Para el temporizador elegí los valores típicos que me daba el fabricante  
 $v_{cc} = 5v$                        $TME = 20\mu s/DIV$                        $RA = 3.9K\Omega$

$$RB = 3K\Omega$$

$$C = 0.01\mu F$$

$$t1 = 0.693 * (RA + RB) * C = 4.78 * 10^{-5}$$

$$t2 = 0.693 * (RB) * C = 2.079 * 10^{-5}$$

$$T = t1 + t2 = 6.859 * 10^{-5}$$

Es el tiempo que envía pulsos al convertidor, es decir, el convertidor prácticamente al instante actualiza el estado de los LEDs.

## 4.7.5. Sensor de ultrasonidos

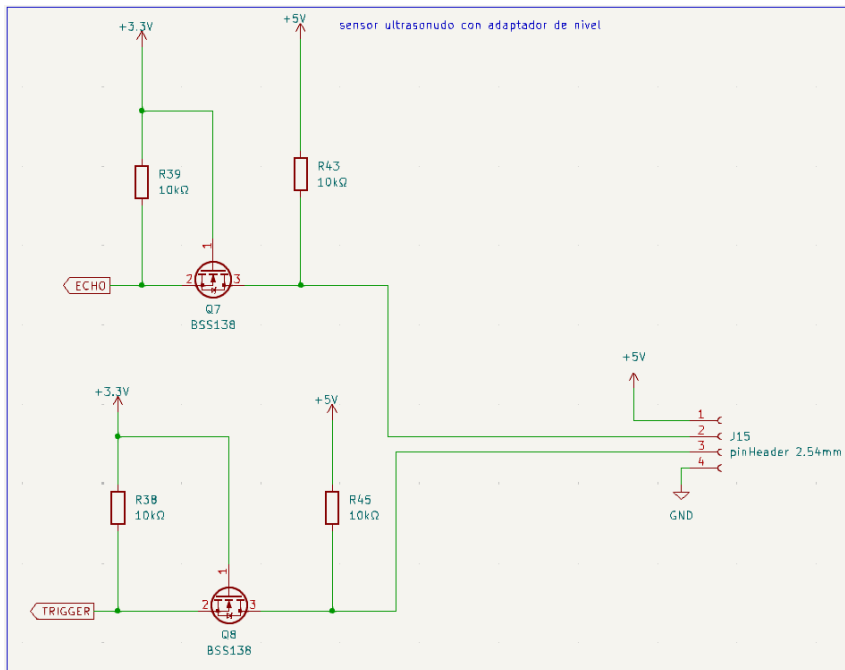


Ilustración 98: Conectores ultrasonido

Como la lectura del microcontrolador ESP32 se realiza a 3.3V se debe hacer una conversación en este caso digital de 5 voltios a 3.3V para ello se usa un adaptador de nivel lógico porque si no quemaría las patillas del microcontrolador.

## 4.7.6. Adaptador de nivel

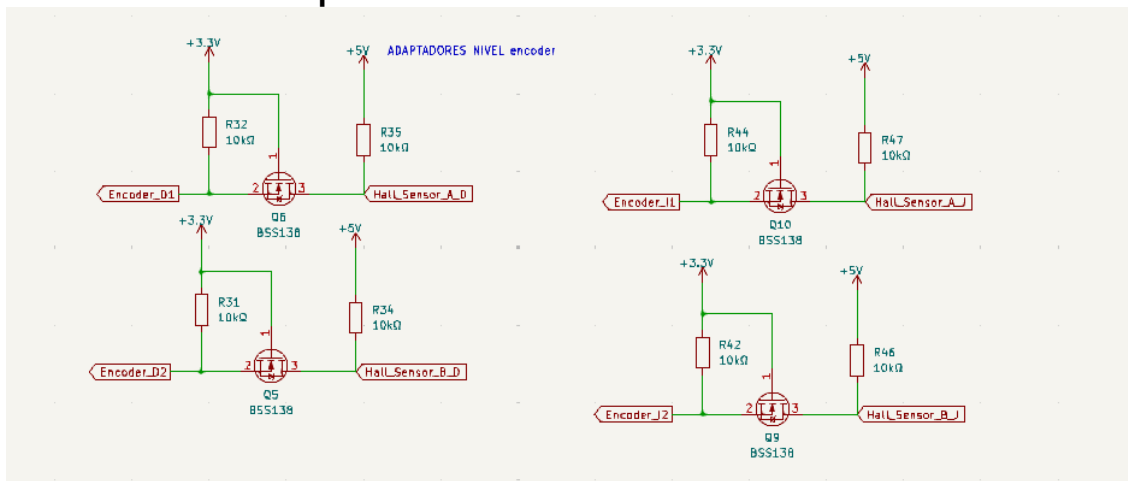


Ilustración 99: Adaptadores de nivel

Los encoders están alimentados a 5 voltios, estos son los encoders del motor, entonces se le pone una adaptador de nivel para que el micro pueda leer sin que haya ningún problema; según el fabricante se podrían alimentar a 3.3V pero lo alimenté a 5V ya que tiene mayor inmunidad al ruido y mayor amplitud de la señal.

## 4.7.7. Conectores servomotores

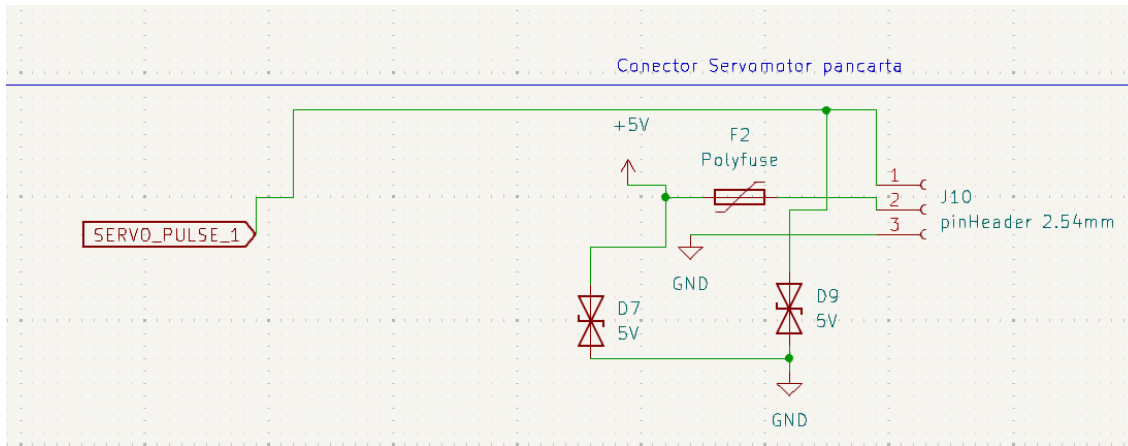


Ilustración 100: Conector servo pancarta

Es un servomotor PWM y 360°, en este caso es el micro el que le envía una señal PWM para que el servo gire, bueno y unas protecciones para proteger al servomotor y al microcontrolador en caso de que hayan un evento inesperado, unos picos de corriente muy altos o algún cortocircuito.

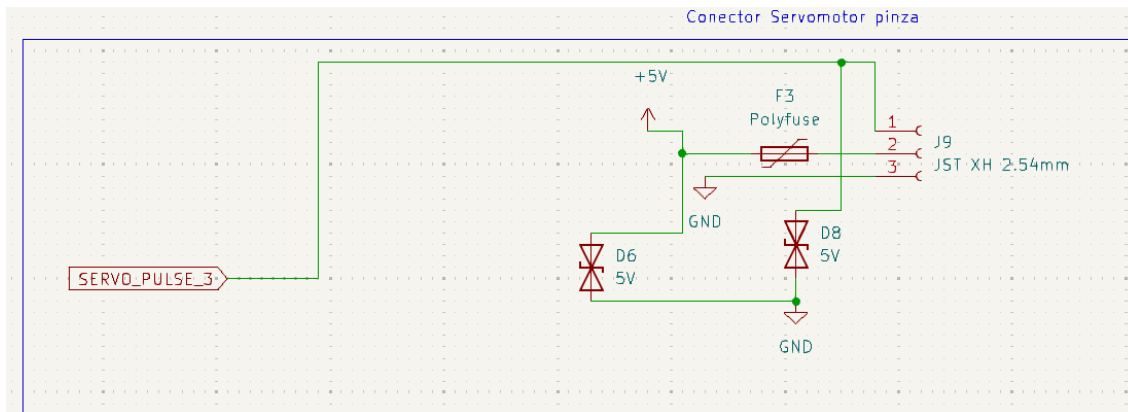


Ilustración 101: Conector servo pinza

Es otro servo PWM y este es para agarrar un objeto, este viene acompañado de un sensor de película fina, que es un sensor resistivo, que en caso de que vea de que ha aumentado su resistencia más de la cuenta pues el microcontrolador envía una señal para parar el servomotor.

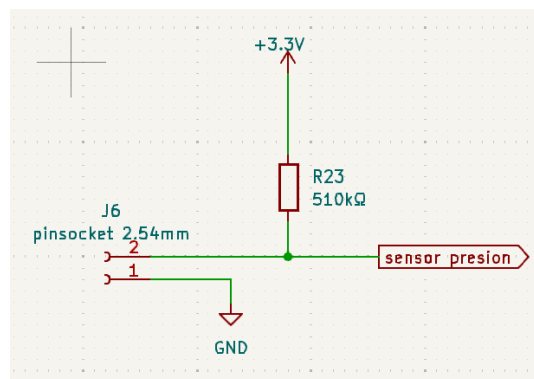


Ilustración 102: Sensor resistivo Kicad

## 4.7.8. Final carrera plataforma

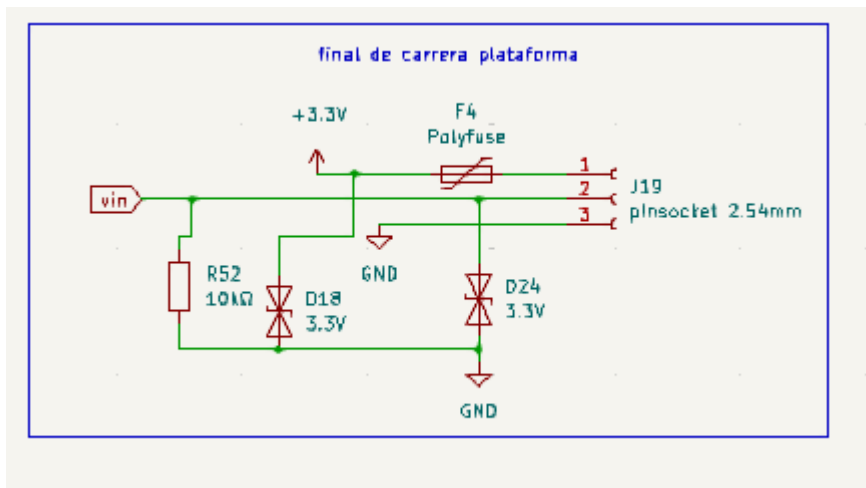


Ilustración 103: Final de carrera plataforma

## 4.7.9. Final de carrera cordón de arranque

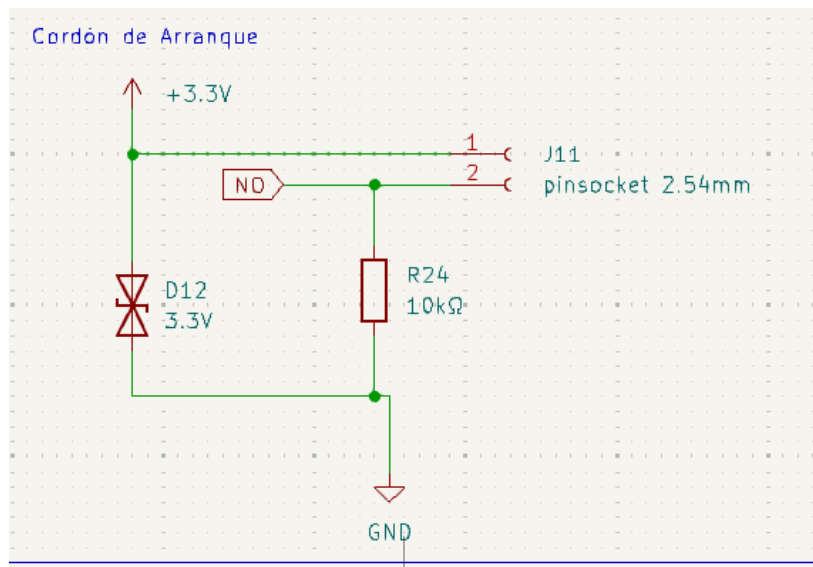


Ilustración 104: Final de carrera cordón arranque

## 4.7.10. Conector sensor infrarrojo

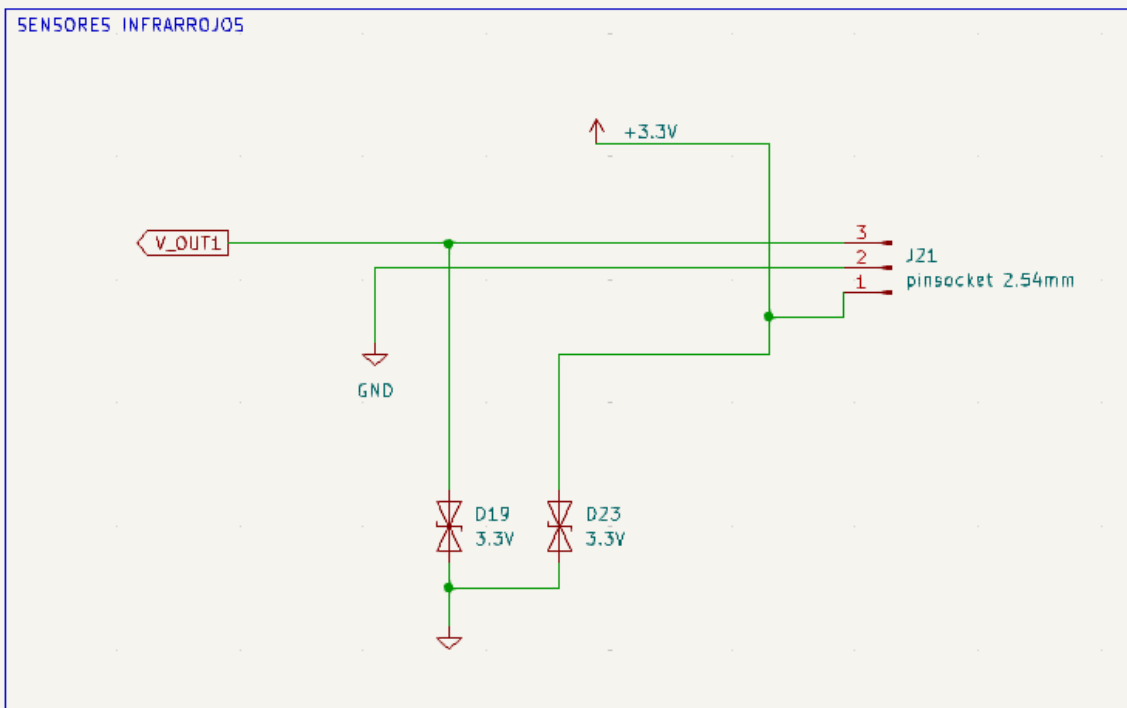


Ilustración 105: Conector sensor infrarrojo

## 4.7.11. Conector bomba de vacío con sensor de presión

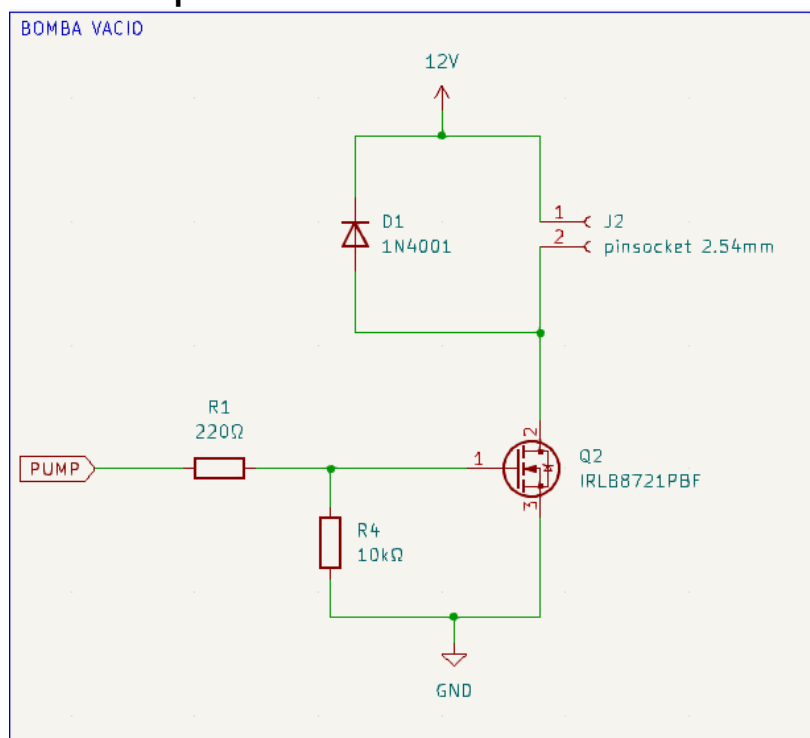


Ilustración 106: Conector bomba de vacío

Aquí el circuito que se puede ver corresponde a la bomba de vacío en el cual está controlado por el MOSFET que actúa como interruptor electrónico para encender y apagar la bomba de vacío.

**Trabajo fin grado**

El funcionamiento sería el siguiente el microcontrolador le enviaría una señal digital a la puerta (gate) del MOSFET, el cual pasaría a estar en modo saturación entonces se conectaría la fuente (source) con el drenador (drain), en ese momento ya empieza a circular corriente para el motor y se activaría. Si el micro envía una señal baja o sea un cero el MOSFET se corta. La resistencia R1 limita la corriente que circula a la puerta(gate). La resistencia R4 evita que la puerta del MOSFET se quede flotando y lo mantiene en estado 1. Y el diodo D1 lo que hace es que cuando el MOSFET se apaga proporciona un camino para que la corriente fluya y así nos genere picos de corriente que puedan dañar al MOSFET.

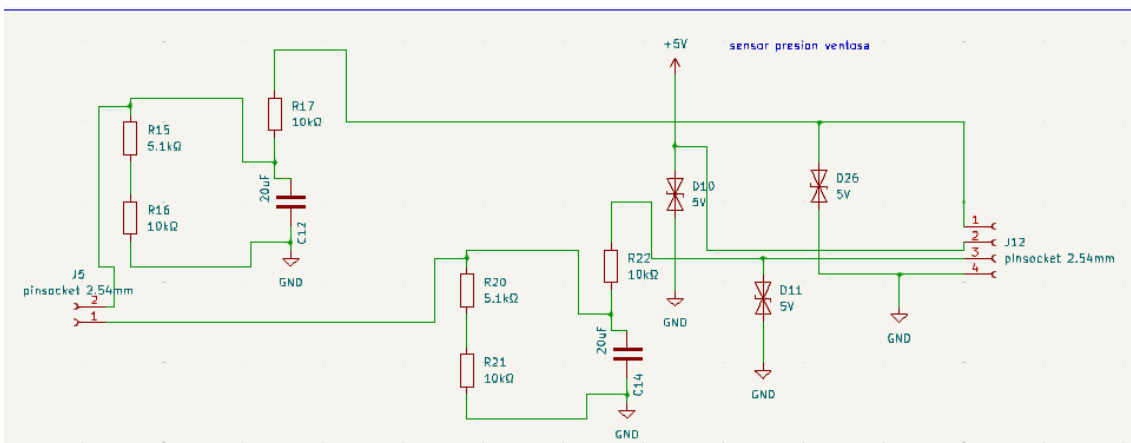


Ilustración 107: Sensor de presión ventosa

este sensor de presión mide la presión para saber cuando la ventosa agarrado un objeto y en tal caso le envía la información al microcontrolador, como no tenía muchos pines pues he decidido enviárselo a la ESP32 CAM.

### 4.7.12. Sensor de color

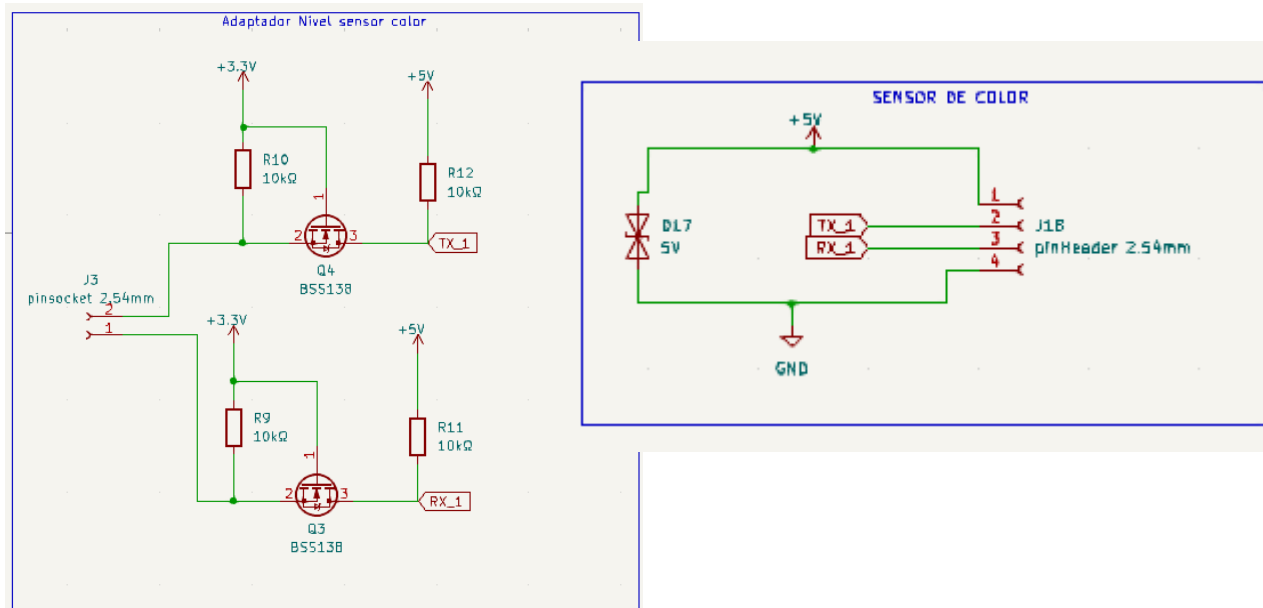


Ilustración 108: Sensor de color con adaptador

Trabajo fin de grado

Igual que en el caso anterior debido a que no tenía más pines en el micro voy a usar la cámara para controlar el sensor de color, le he puesto un adaptador de nivel aunque técnicamente este sensor acepta 3.3V pero no tiene mucho alcance con ese voltaje por eso lo he conectado a 5 V ya que tiene más alcance.

### 4.7.13. ESP32 CAM

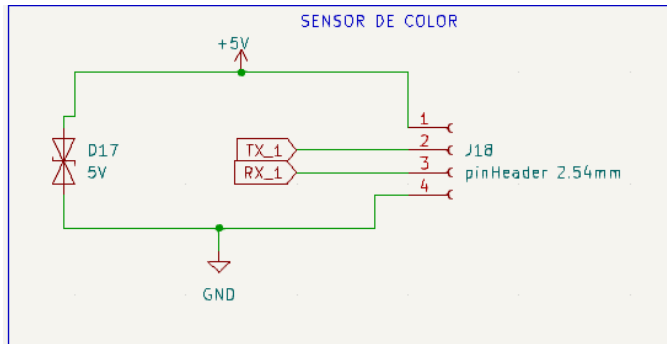


Ilustración 109: ESP32 CAM Kicad

Se conectó mediante I2C al microcontrolador con dos resistencias de 10k

### 4.7.14. ESP32 S3 WROOM 1

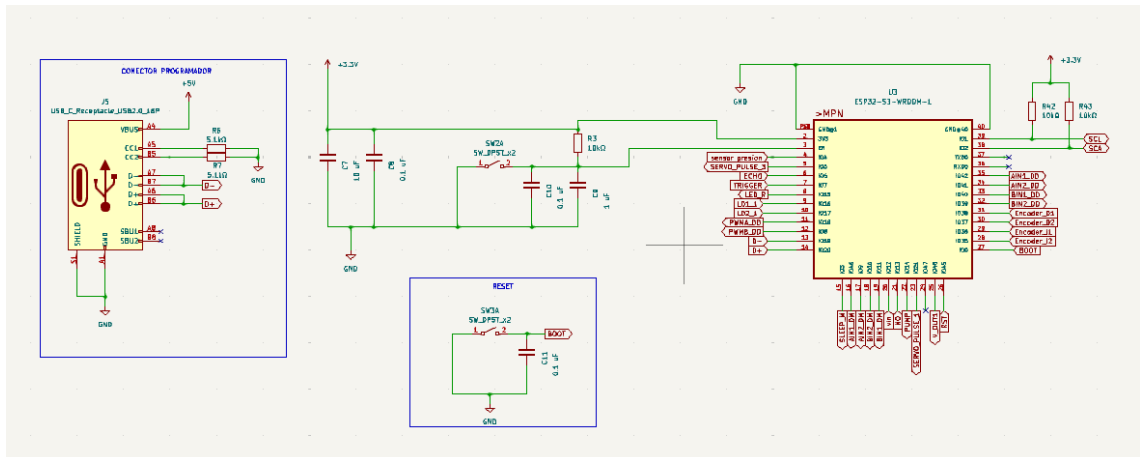


Ilustración 110: ESP32 S3 WROOM 1 Kicad

Aquí está el encargado de controlar todo, los actuadores, los sensores. Le he puesto un conector programador el cual se le introduce el lenguaje de Arduino y un reset para volver a reiniciar el programa.

## 4.8. Diseño de la PCB

A la hora de diseñar la PCB lo primero que se debe tener en cuenta es el tamaño de las pistas, obviamente el robot no va a consumir toda la potencia al mismo tiempo primero consumirá la de los motores motrices, luego la de los servomotores y la potencia de los sensores y actuadores (es muy poquita).

Kicad tiene una calculadora propia que nos permite saber el ancho de las pistas del circuito

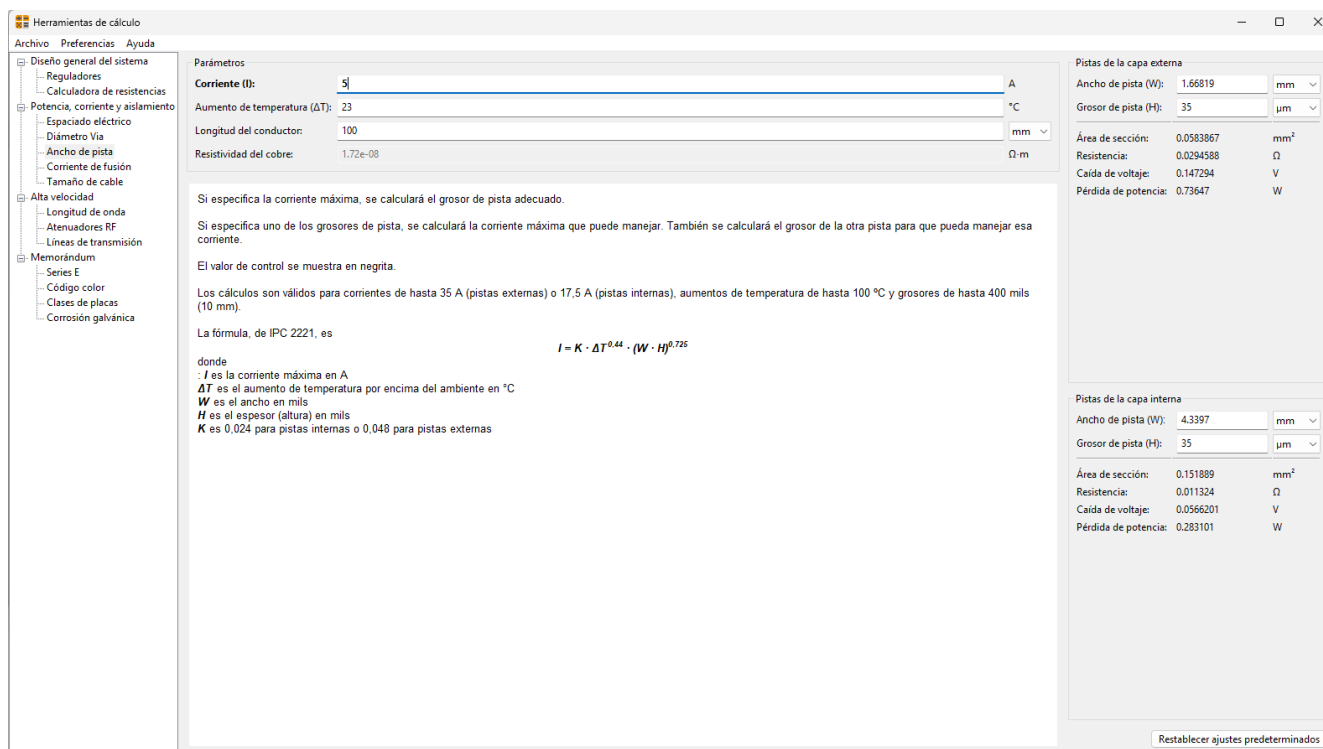
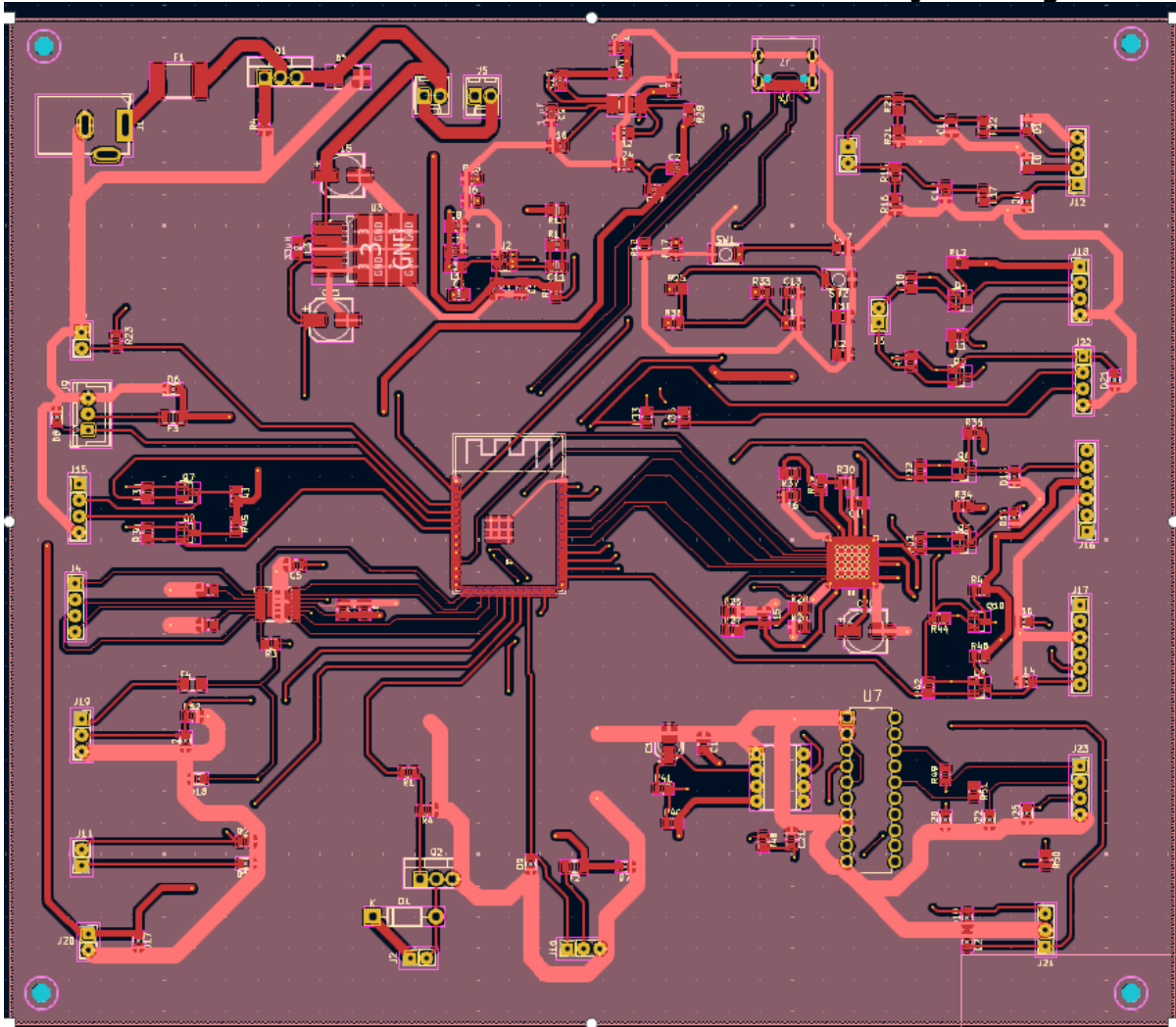


Ilustración 111: Calculo ancho de la pista

para calcular el ancho de la pista se ha de tener en cuenta la corriente que suministra los componentes, por ejemplo, lo que más consume corriente son los motores, según las especificaciones del fabricante cada motor consume de manera normal 1.7 amperios, y ya en el caso de algún inconveniente pero que no es lo habitual su corriente stall sería de aproximadamente 3 amperios.

Dejando un margen para posibles inconvenientes y todo, el ancho de la pista de la potencia que entra directamente a la PCB, la cual he llamado +BATT le he puesto un ancho de pista teniendo en cuenta el consumo de los dos motores que es lo que más consume y el consumo habitual dándole un margen, 5 amperios, y una temperatura que suele estar en la competición de unos 23°C, el programa me ha dado 1.6 mm de ancho. Para las corrientes suministradas de 5 voltios el ancho de la pista le puse aproximadamente 1 mm, y para el de 3.3 voltios el ancho de la pista le puse 0.8 milímetros. Y el ancho de la pista que iban conectados directamente al microcontrolador 0.3 mm, con todo eso queda así.



*Ilustración 112: PCB vista superior*

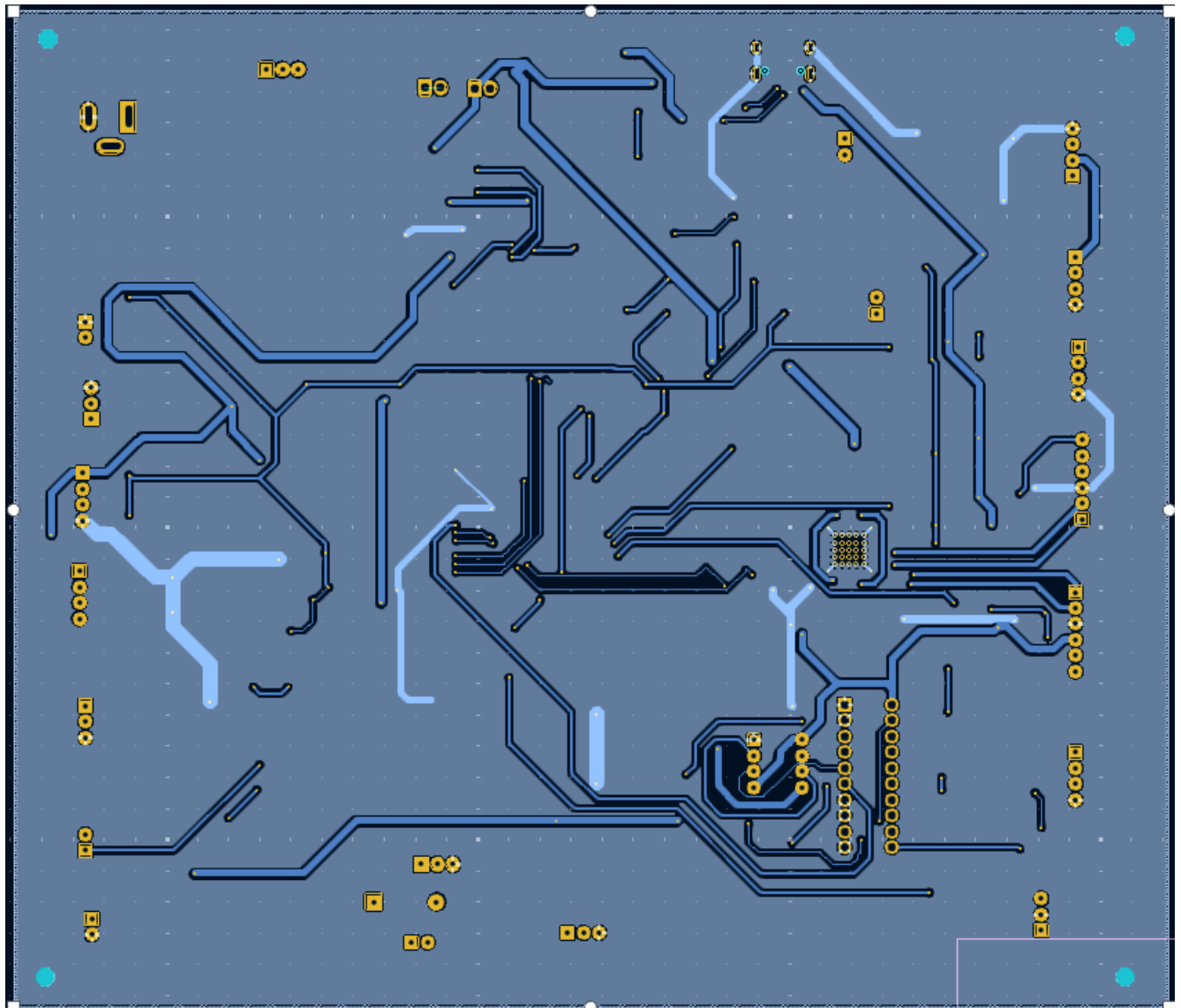
El conector para programar el microcontrolador lo he colocado en la esquina para que sea más cómodo a la hora de conectarlo y que no haya ningún obstáculo.

Los conectores están ubicados en los bordes de la placa, lo cual es una buena práctica. Esto facilita el acceso a ellos cuando se quiera conectar algo externo (como sensores, módulos o cables), y también ayuda a mantener un diseño más limpio y organizado en el centro de la placa.

En cuanto al diseño de las pistas, tiene que estar bien distribuidas. Se han usado vias para cambiar de capa donde haga falta, y también hay zonas dedicadas a la alimentación, como las de 3.3V y 12V. Todo está ordenado y parece bien pensado para evitar cruces innecesarios de pistas.

En resumen, la capa superior de esta PCB está bien estructurada, con el micro

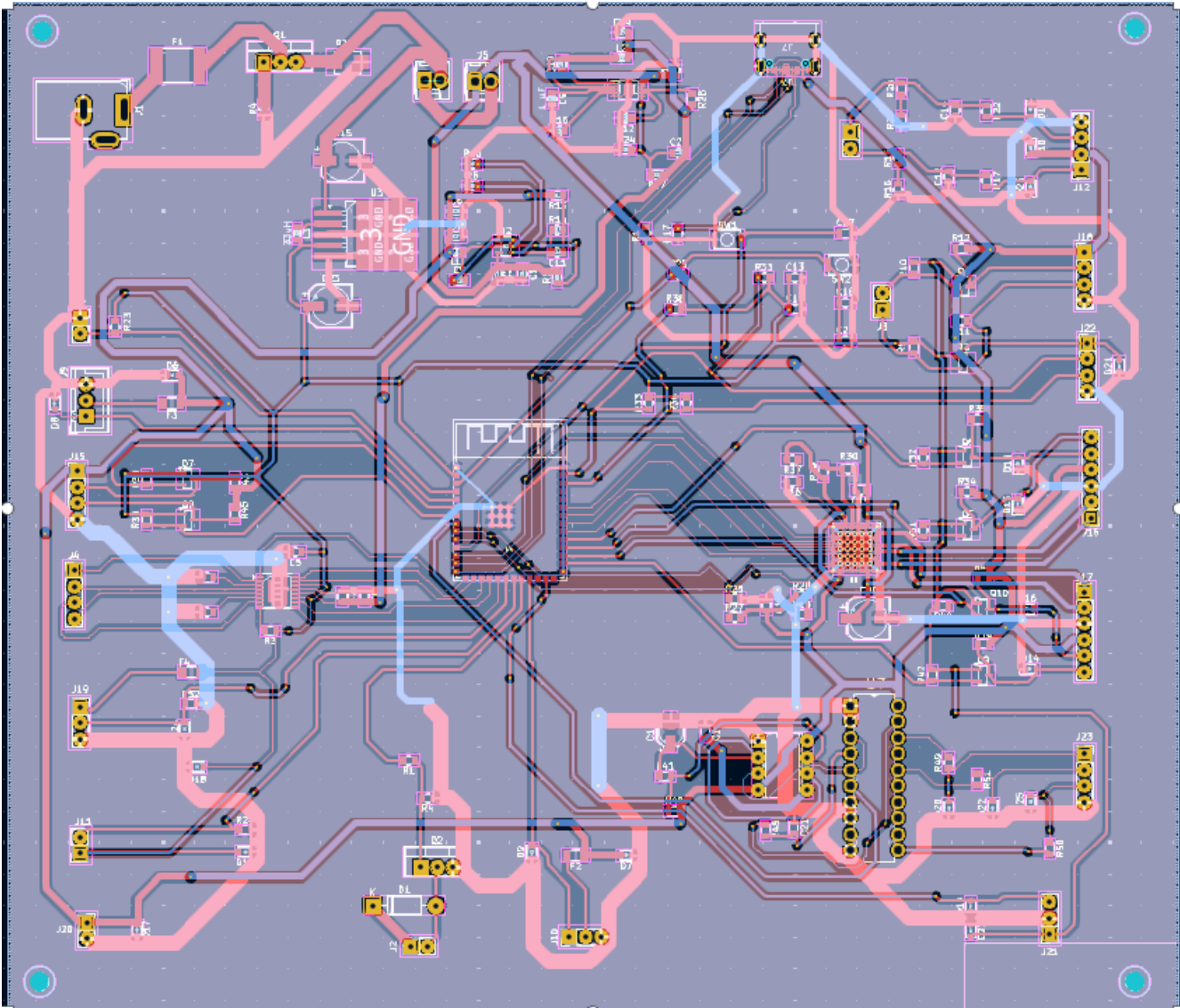
como componente central y las conexiones saliendo hacia los bordes de forma clara.



*Ilustración 113: PCB vista inferior*

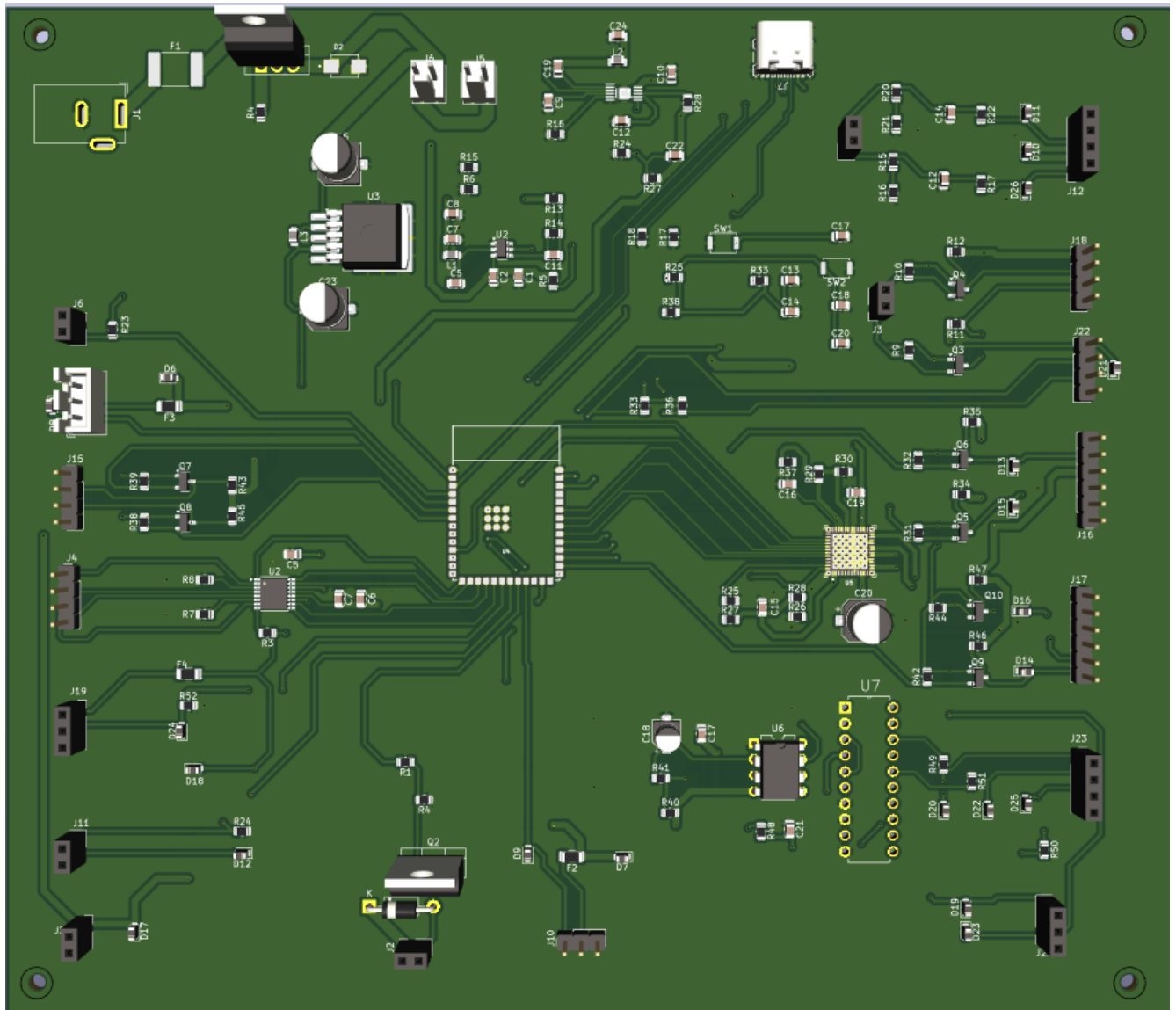
Esta capa se ha utilizado principalmente para enrutar señales que no se podían acomodar por la parte de arriba sin que se cruzaran. Gracias a las vías (los agujeros que conectan ambas capas), se logra una buena comunicación entre ambas caras del PCB, y se aprovecha mucho mejor el espacio disponible.

Las pistas están bien separadas y distribuidas, evitando cruces innecesarios. Algunas zonas también parecen estar dedicadas a la alimentación o masa, aunque en este caso no hay planos visibles. Aun así, esta capa cumple su función de ayudar a completar el circuito de forma ordenada.

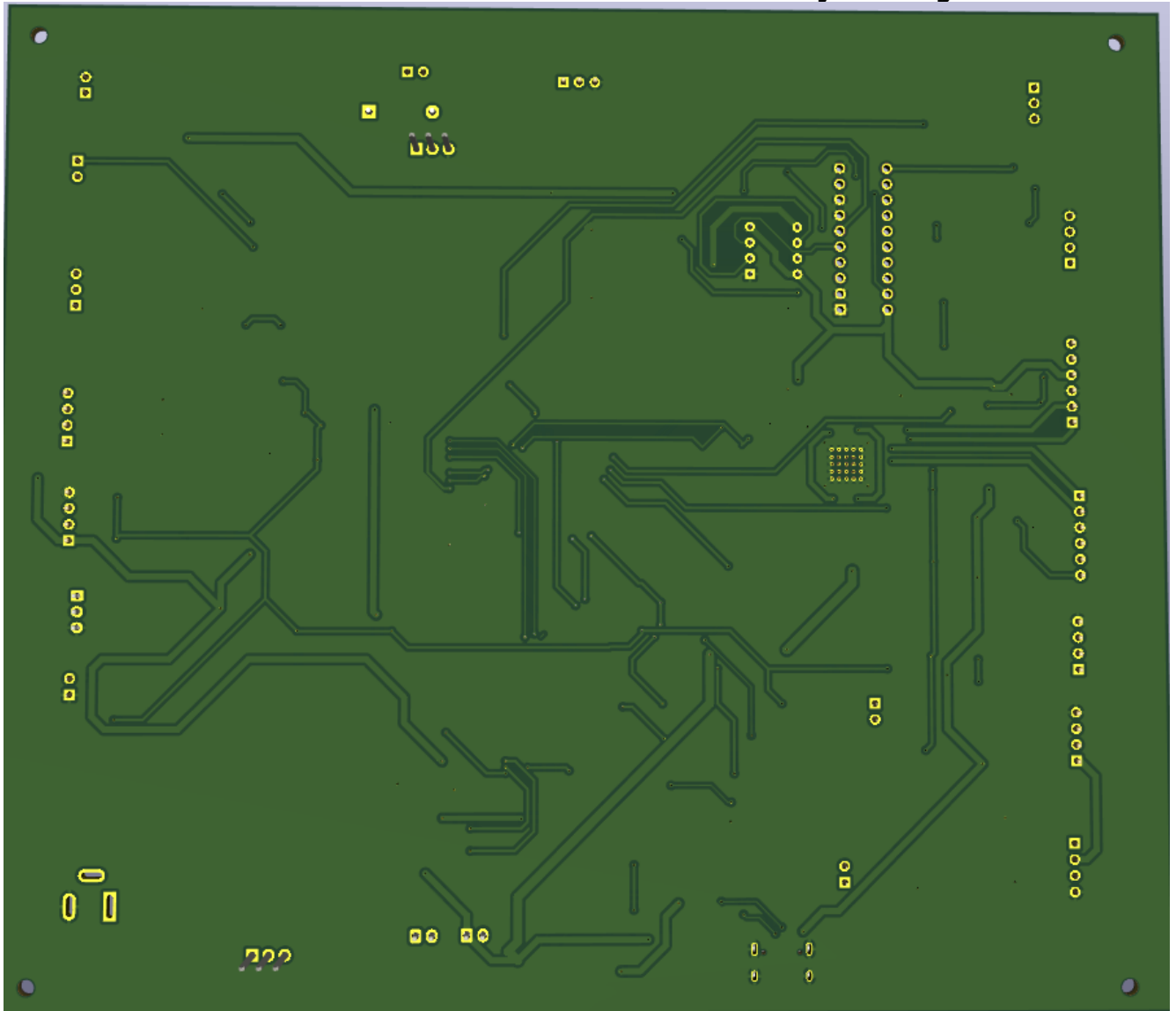


*Ilustración 114: PCB completa*

Se puede apreciar que tanto la capa inferior como en la capa superior en ningún momento a habido ángulos rectos, ya que la impresión a la máquina le costaría o no los podría dibujar bien, todos los ángulos son curvos entonces la PCB estaría bien diseñada.



*Ilustración 115: PCB vista 3D capa superior*



*Ilustración 116: PCB vista 3D capa inferior*

## 4.9. COMO ORIENTAR LA PROGRAMACIÓN

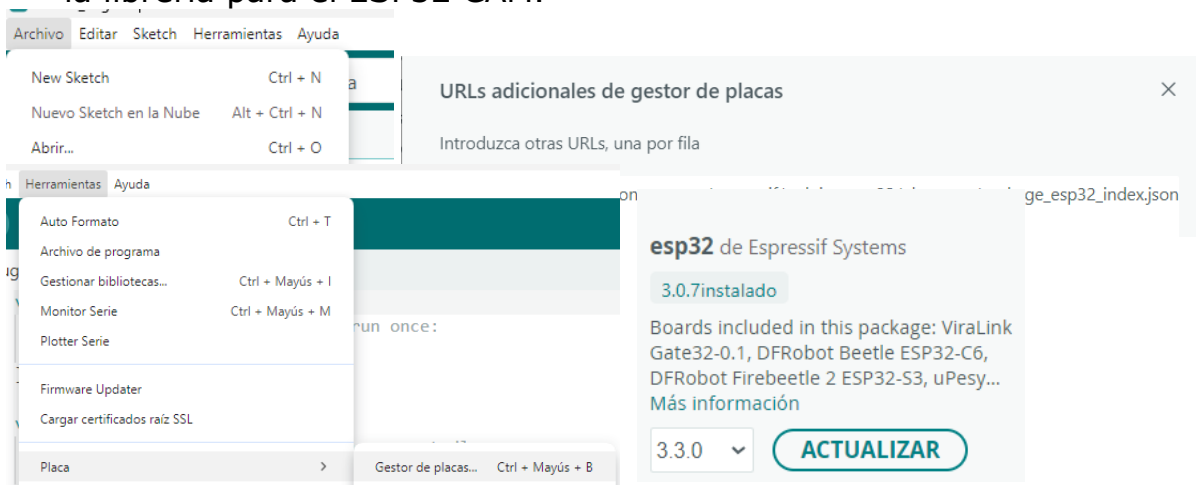
### 4.9.1. ESP32 CAM

Para programar la Cámara he usado un programa basado en inteligencia artificial llamado EDGE IMPULSE, que entrena la cámara para detectar un objeto, entrena modelos de ML o Deep learning y luego los optimiza para que se ejecuten en microcontroladores en forma de librerías.

A continuación, detalló el proceso de cómo se hace:



Se deben instalar una librería que es la que se va a encargar de tomar las imágenes. Pero antes de hacer eso hay que asegurarse de tener instalado la librería para el ESP32 CAM.

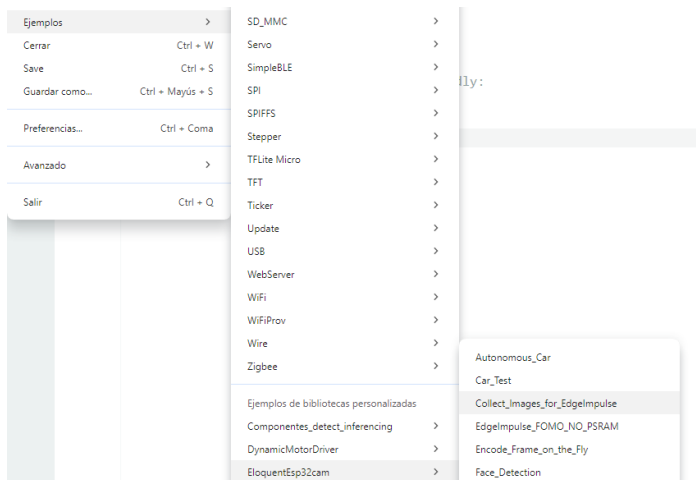


Luego procedemos a configurar los parámetros para poder usar la cámara de esta manera

Trabajo fin de grado

Placa: "ESP32 Wrover Module"	>
Puerto: "COM5"	>
Reload Board Data	
Obtener información de la placa	
<hr/>	
Core Debug Level: "None"	>
Erase All Flash Before Sketch Upload: "Disabled"	>
Flash Frequency: "80MHz"	>
Flash Mode: "QIO"	>
Partition Scheme: "Huge APP (3MB No OTA/1MB SPIFFS)"	>
Upload Speed: "921600"	>

Una vez instalado eso ya procedemos a tomar las imágenes para el reconocimiento del objeto.



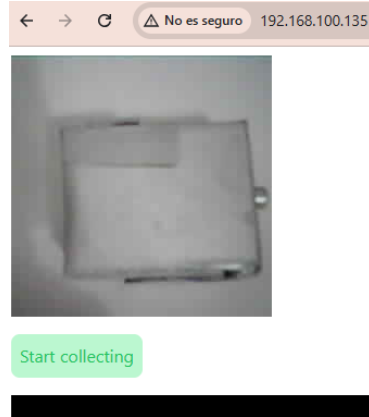
Para ello vamos a ir ejemplo y vamos a tomar un programa de una librería que antes instalamos, que nos va a tomar las imágenes para Edge impulse

Para la librería esta de eloquente ponemos esta configuración, luego dentro ponemos los datos de nuestro wifi y nuestra contraseña y cambiamos a la opción aithinker, y subimos el programa.

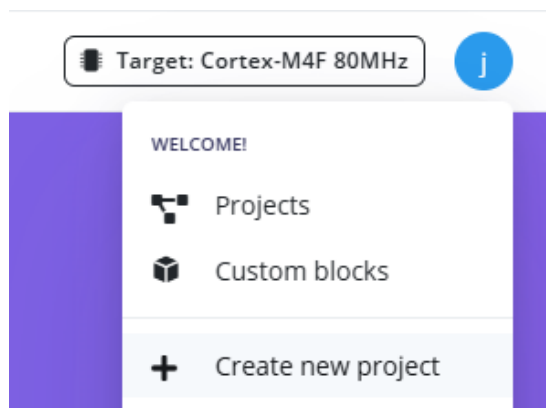
Después de eso copiamos la dirección que nos da por el puerto serie.

Placa: "AI Thinker ESP32-CAM"	>
Puerto: "COM3"	>
Reload Board Data	
Obtener información de la placa	
<hr/>	
CPU Frequency: "240MHz (WiFi/BT)"	>
Core Debug Level: "None"	>
Erase All Flash Before Sketch Upload: "Disabled"	>
Flash Frequency: "80MHz"	>
Flash Mode: "QIO"	>
Partition Scheme: "Huge APP (3MB No OTA/1MB SPIFFS)"	>

Después de eso elegimos un objeto y lo que tenemos que hacer es tomarle fotos desde diferentes ángulos, para que así la cámara se pueda entrenar bien y pueda reconocer el objeto de manera más precisa, guardamos los archivos en una carpeta.



Vamos a Edge impulse y creamos un nuevo proyecto



Luego subimos los archivos, les ponemos el nombre a cada una de las piezas el nombre con el que queramos distinguirlo.

SAMPLE NAME	LABELS	ADDED
resistencia_1754940915616	resistencia	ago 11 2025, 21:44:48
resistencia_1754940916363	resistencia	ago 11 2025, 21:44:48
resistencia_1754940916612	resistencia	ago 11 2025, 21:44:48
resistencia_1754940914119	resistencia	ago 11 2025, 21:44:48
resistencia_1754940915113	resistencia	ago 11 2025, 21:44:48



Luego y cogemos el objeto que queremos que nuestra cámara detecte y lo encerramos con este cuadrito en las diferentes posiciones y si es necesario con diferentes fondos para que la Cámara no confunda el fondo con el objeto.

Aquí la imagen no se procesa pixel por pixel, aquí lo que hace es básicamente que busca patrones matemáticos con bordes, texturas, formas y colores.

En nuestro caso como queremos coger objetos por color y forma, lo ponemos en formato RGB.

	BACKGROUND	LED AZUL	RESISTENCIA
BACKGROUND	100%	0%	0%
LED AZUL	0%	100%	0%
RESISTENCIA	0%	0%	100%

Luego después de todo esto nos da la información ya que ya es capaz de distinguir al cien por cien los dos objetos que le hemos dado en este ejemplo más el fondo.

SELECTED DEPLOYMENT

**Arduino library**  
An Arduino library with examples that runs on most Arm-based Arduino development boards.

Configure your target device and application budget

Target device

Define your target device requirements to inform model optimizations and performance calculations. No device yet? Use the default settings which you can change at any time.

Target device:

Processor family:

Clock rate <sup>?</sup>:  MHz  
Max

Y para terminar le decimos que queremos que nos haga una librería, elegimos la placa con la que vayamos a trabajar nosotros y la descargamos. Y simplemente lo único que tenemos que hacer es ir al arduino a incluir librerías la añadimos, y ya está ya tenemos nuestro algoritmo que puede detectar los objetos que nosotros le hayamos dado con una precisión bastante alta.

Una vez carguemos el archivo veremos que nos va a devolver los siguientes datos por el puerto serie

```
led azul (0.996094) [ x: 40, y: 32, width: 8, height: 16 ]
```

Es decir, nos devuelve el tipo de objeto y sus coordenadas cartesianas, eso nos viene bien las coordenadas, ya que podremos calcular la ruta que va a llevar el robot.

### 4.9.2. Conexión I2C

### Trabajo fin grado

El ESP32 tiene pines predeterminado para I2C, pero prácticamente podemos usar cualquier pin para la I2C usando el comando wire

```
#define SDA_CAM 14
#define SCL_CAM 15
#define I2C_SLAVE_ADDR 0x08

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  Wire.begin(SDA_CAM, SCL_CAM); // Maestro
```

Aquí sobre la librería que hemos generado con el edge impulse, pondremos esto para definir los pines y la dirección de manera manual.

```
ei_printf(" %s (%f) [ x: %u, y: %u, width: %u, height: %u ]\r\n",
bb.label,
bb.value,
bb.x,
bb.y,
bb.width,
bb.height);
if (bb.value > 0.8) { // Solo enviamos si la probabilidad es alta
  Wire.beginTransmission(I2C_SLAVE_ADDR);
  Wire.write((uint8_t *)bb.label, strlen(bb.label)); // Enviar toda la cadena
  Wire.endTransmission();
}
```

Luego le tenemos que enviar al esp32 s3 wroom 1 los datos obtenidos por ejemplo aquí le enviamos que si la probabilidad de que sea un objeto (que es la que he entrenado para la Cámara) es mayor al 80% le enviamos la información de ese objeto, tipo de objeto y coordenadas, que es lo que vamos a usar.

### 4.9.3. Driver motores motrices

Para esto me inspirare en tres librerías para que sea más sencillo.

```
#include "TB67H420FTG.h"
#include <ESP32Encoder.h>
#include <Wire.h>
```

Las partes más importantes para poder acercarse a los objetos de manera correcta, la primera calcular el error horizontal eso básicamente se hace por PID, es una operación matemática que lo que hace es corregir la ruta del robot para que no se desvíe

$$\text{error} = \text{valor deseado} - \text{valor actual}$$

```
float centerX = 0.0;
float errorX = targetX - centerX;
int adjustment = KP_X * errorX;
```

Y bueno el KP\_X significa que cuanto más lejos este del centro más rápido gira el motor para corregir.

```
int motorLeft = constrain(MAX_SPEED - adjustment, 0, MAX_SPEED);
int motorRight = constrain(MAX_SPEED + adjustment, 0, MAX_SPEED);
```

**Trabajo fin de grado**

Y este código lo que hace es controlar el giro de los motores, básicamente determina si un motor debe de girar a una velocidad determinada, es decir si un motor debe de girar más rápido que otro con lo cual se hace un giro, y si está centrado pues que los dos giren a la misma velocidad. Como los valores que devuelve la ESP32 CAM son coordenadas X e Y, la distancia de los objetos se deben medir con el sensor de ultrasonidos. Y en esa parte ya se pueden usar los encoders para medir la distancia que hay hacia el objeto, quedando algo así.

```
long posAvg = (posLeft + posRight) / 2;
long errorZ = targetZ - posAvg;
int adjustZ = KP_Z * errorZ;

int motorLeft = constrain(adjustZ - adjustX, 0, MAX_SPEED);
int motorRight = constrain(adjustZ + adjustX, 0, MAX_SPEED);
```

#### 4.9.4 Driver motor paso a paso

Este sencillo de programar se puede hacer de manera manual, haciendo secuencias con los digitalWrite.

En donde se tienen que activar con la entrada sleep del driver. El giro en sentido horario o anti horario es en función de lo que se le ordene con AIN1,AIN2, BIN1, y BIN2.

#### 4.9.5. Bomba de vacío

Es bastante sencillo de programar, ya que lo único que hay que hacer es enviarle un estado alto (3.3V) o un estado bajo (0V) en la puerta del MOSFET, y esto se hace con un digitalWrite(pin, estado), Con lo cual realmente no hay ningún problema en programarlo es prácticamente una línea.

#### 4.9.6. Sensor de ultrasonido

```
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
long duracion = pulseIn(ECHO_PIN, HIGH);
```

Bueno aquí lo que se hace es enviar señales en forma de pulsos mediante el Trigger y el Hecho recibe esas señales, lo reciben mediante el comando pulseIn, recibe un entero grande, que es el tiempo que ha tardado en ir la señal al objeto y volver, se mide en microsegundos.

$$velocidad\ sonido = 343 \frac{m}{s} \rightarrow 0.0343 \frac{cm}{us}$$

Como es ida y vuelta se divide entre 2, quedando así

```
float distancia = (duracion * 0.034) / 2;
```

#### 4.9.7. Sensor infrarrojo

Esto es bastante fácil de programar y no requiere ningún tipo de

dificultad, es tan fácil como poner un `digitalRead(pin)`, y te devuelve un uno o un cero.

### 4.9.8. Programación servomotores

Para programar estos servomotores que son tipo PWM, me apoyaré en una librería llamada:

```
#include <ESP32Servo.h>
```

Éste servomotor le tienes que decir la posición en la que tiene que moverse, por ejemplo si quieres que si gires de 0 a 180°

```
for (int pos = 0; pos <= 180; pos++) {  
    myservo.write(pos);  
    delay(15);  
}
```

Obviamente el servo motor gira a la posición que tú le dices, para el despliegue de la pancarta, ya que solo se tiene que mover en una posición hasta que el cartel este a ras de suelo, pero claro para las pinzas para agarrar un objeto, en este caso le he puesto un sensor resistivo que lo que hace es que el varía su resistencia y en función de la resistencia varía el voltaje, por ejemplo:

```
if(sensorValor < UMBRAL) {  
    myservo.write(pos);  
    pos++;  
if(pos > 180) pos = 0;  
    delay(15);  
}.....
```

En dónde `sensorValor` se declararía como un `analogRead`, ya que es un ADC del microcontrolador, ya que tiene que devolver un valor analógico.

### 4.9.9. Sensor de color

Yo cogí un sensor que lo que es básicamente un modelo simplificado del sensor TCS3200, pero no había demasiada información de este sensor. El punto clave para programarlo es que se tiene que hacer con un comando especial que es él "AT+FRE", entonces con esté comando el sensor te devuelve los valores de los colores.

```
void setup() {  
  Serial.begin(115200);  
  
  SENSOR_SERIAL.begin(SENSOR_BAUD, SERIAL_8N1, RXD1, TXD1);  
}  
  
void loop() {  
  unsigned long ahora = millis();  
  
  if (ahora - ultimoEnvio >= intervalolectura) {  
    SENSOR_SERIAL.println("AT+FRE");  
    ultimoEnvio = ahora;  
  }  
  
  while (SENSOR_SERIAL.available()) {  
    char c = SENSOR_SERIAL.read();  
    Serial.write(c);  
  }  
}
```

aquí le pide datos al sensor cada cierto tiempo con el comando especial y luego los lee por el puerto serie. Ya con los datos que te dé lo que hay que hacer es una lógica para comparar el valor ya sea el R (rojo), G (verde), B (azul) o C (neutro); entonces cuanto mayor sea un valor es mucho más probable de que sea ese color hay que aplicar esa lógica.

## 5. EJECUCIÓN DE PRUEBAS Y PROBLEMAS SURGIDOS

Como yo había pedido los componentes, y me llegaron antes de que yo mandara hacer la fabricación de la PCB, pude realizar ciertas pruebas para verificar el correcto funcionamiento, para realizar las pruebas me he valido de un pack de arduino mega 2560, el ESP32 Devkitch N16R8 y diferentes cables que tenía. Han habido circuitos como el de los indicadores LEDS, en el que como eran integrados tan pequeños no podíamos verificar su comportamiento de manera crítica; en ese caso lo verificamos con simuladores como proteus.

Por ejemplo para verificar que la conexión de la I2C era cierta la conecté físicamente, y me funcionaba de manera perfecta.

Para comprobar la conexión del sensor de ultrasonidos, me he valido de adaptadores de niveles comerciales, para hacer la conversión de 5 a 3.3V y me funcionó perfectamente.

El sensor de infrarrojos la conexión realmente es fácil de hacer, y la programación es fácil, ya que solo te debe enviar el estado alto o bajo. El inconveniente que sí que le encontré cuando hice las pruebas es que si empieza a haber bastante luz le empieza a costar bastante captar los objetos, o sea sí sabía que la luminosidad afecta, pero no pensé que tanto, luego la distancia es corta y la desventaja principal respecto al ultrasonido es que no te mide la distancia del objeto, lo bueno es que no lo tienes que poner en una posición específica como el ultrasonido ya que te puede detectar el suelo como un obstáculo. Entonces viendo todo esto me planteé usar el sensor de infrarrojos para la detección de la pancarta y el de ultrasonido para la detección de los objetos que se van a coger ya que es más efectivo.

En la parte del sensor de color, sí que me resultó un poco más complicado la verdad, porque este es un modelo que realmente no hay información en internet, ya que este es un modelo derivado de las versiones TCS230 TCS3200, Y no tiene las entradas típicas s0...s3, sino que en su interior tiene un microcontrolador que ya incorpora todas esas entradas y en su lugar tiene un pin de transmisión y de recepción (TX y RX). Y a la hora de programarlo hay que usar un comando especial, el "AT-FRE". Este sensor de color funciona en un rango de 3.3-5V, claro con 3.3V funciona perfectamente pero es que cuando lo probé la distancia era corta, entonces preferí conectarlo a 5V y ponerle un adaptador de nivel.

A la hora de programar el servomotor SM-S4303R Realmente no hubo ninguna complicación, es solamente un servo de tipo PWM, en donde le tienes que decir los grados que quieres que te gire y ya está.

En el servomotor de la pinza, FETEECH SCS225, sí que encontré complicaciones, o sea este servomotor es un servomotor que te devuelve retroalimentación, hay una librería en arduino llamada "FTServo", si vas a ejemplos, en la parte "SCSCL" ya que pertenece a la familia SCS, y luego vas al ejemplo que dice "feedback", ahí puedes ver que te devuelve datos como la

---

**Trabajo fin de grado**

corriente y el voltaje entre otros, yo la verdad quería usar este servomotor. Este servomotor es usado en por ejemplo robots tipo cuadrúpedos que pueden caminar. El inconveniente que encontré en este servomotor, es que tiene un cable half dúplex, que básicamente es el mismo cable tiene el transmisor y el receptor (TX y RX), claro para conectar eso se debe buscar una forma por ejemplo mediante un transistor en donde se puede dividir la señal de manera correcta, para que no se envíen a la vez el transmisor y receptor y haya eco; al final el servo me dejó de funcionar seguramente habrá habido un corto en algún sitio, entonces ya dejé de probar este y me decanté por un servomotor de tipo PWM y con un sensor resistivo. Aunque sinceramente no me gustan usar los sensores resistivos, ya que tiene mucha sensibilidad en este proyecto me vale pero en proyectos complejos que tengas que agarrar objetos de mucho más peso ya se complicaría un poco la cosa. El FETEECH SCS225 tiene de bueno qué cuando el servomotor detecta un esfuerzo aumenta la corriente y en función de eso puedes detener el servomotor, esto es más efectivo pero bueno no se pudo.

En cuanto a los drivers tanto de los motores motrices como del motor paso a paso, evidentemente no pude hacer una prueba física directa, ya que para ello tenía que haber pedido algún circuito integrado ya fabricado, Para así probar la programación, en su lugar hice un análisis visual mirando librerías que ya habían y aplicando la lógica en algún otro proyecto que no era lo mismo pero que era similar entonces ya con eso me valía.

Tuve un descuido, es que a la hora de pedir el motor para las ruedas motrices, sin darme cuenta pedí uno de 24V, entonces eso hizo que agregar un nuevo circuito a la PCB, lo cual me retrasó más, y encima me agrega circuitería.

En cuanto a la fuente de alimentación, evidentemente no se puede comprobar, pero no hace falta ya que el propio fabricante te da los circuitos hechos y ya te dice el voltaje que vas a obtener; utilice el fabricante de Texas Instrument para poder verificar circuitos.

También tuve algún que otro descuido, sobre todo de cara a como tuve que agregar circuitería de 24 V, se me pasó cambiar el modelo de la polaridad inversa qué me aguantaba +-20V, por uno que me aguantara +-25V, esto es la verdad fue un fallo qué me retrasó más.

## 6. CUMPLIMIENTOS DE OBJETIVOS

Este apartado tratara dos tipos de objetivos, los objetivos propios de la competición y los objetivos planteados para el TFG.

### 6.1. REGLAMENTO EUROBOT

1- Una de las reglas a cumplir era el perímetro del robot, que con las pinzas desplegadas no podía superar los 1400mm, ese objetivo lo cumplí satisfactoriamente.

2- La altura máxima es de 350, y el botón de emergencia puede alcanzar los 375mm, esta también lo cumplí tal y como lo especificué en el diseño 3D.

3- No utilice objetos inflamables o explosivos que pudieran representar un peligro.

4- El robot lleva una parada de emergencia de fácil acceso que para los actuadores. También lleva un cordón de arranque.

Los objetivos que no se pudieron cumplir son los relacionados con los objetos que se van a manipular, debido a que me costó encontrar un mecanismo que fuera capaz de coger 4 barriles con la tabla, y cuando lo encontré los materiales tardaban en llegarme. Debido a eso decidí coger una pinza con una ventosa para hacer una demostración.

### 6.2. OBJETIVOS GENERALES DEL TFG

#### 1. Diseñar una estructura mecánica ligera, resistente y modular, con un peso inferior a 10 kg:

Tal y como se puede ver en la imagen el peso de mi robot es la mitad del máximo que yo mismo me establecí con lo cual este objetivo lo cumplí.

Luego también he hecho un robot que esta bien atornillado se sostiene muy bien y no se cae a trozos ni mucho menos y además aguanta perfectamente las vibraciones.

Las plantas del robot están hechas de aluminio con lo cual es capaz de aguantar el peso de los componentes sin ningún tipo de problema.

En conclusión, este objetivo planteado por mi esta mas que cumplido, el robot es estable y no se daña tan fácil.

	A	B
1	componente	peso (kg)
2	motor pololu 37d	0.42
3	servomotor SM-S4303R+miniservo	0.101
4	motor paso a paso nema 17,husillo	0.37
5	ventosa succion	0.05
6	bateria	0.6
7	ruedas goma	0.12
8	rueda bola	0.08
9	on/off	0.006
10	cordon arranque	0.1
11	ultrasonido	0.01
12	sensor color	0.009
13	sensor presion	0.002
14	soporte motores	0.326
15	bomba vacio	0.3
16	SENSOR MPX10DP	0.006
17	ESP32CAM	0.01
18	bms	0.05
19	seta	0.05
20	final carrera	0.03
21	barrel jack	0.00175
22	infrarrojos	0.004
23	PLANTAS ALUMINIO	1
24	peso aprx pcb	0.4
25	soporte paso a paso	0.059
26	engranages	0.15
27	extra aprx(tornillos, cables,soportes...)	1
28	TOTAL	5.25475

Ilustración 117: Peso robot

## 2. Implementar un sistema de propulsión con una aceleración mínima de $0,8 \text{ m/s}^2$ y una velocidad máxima de $1.2 \text{ m/s}$ :

Cuando pruebo el robot los motores me lo corren sin ningún tipo de problema, la aceleración estimo que con un PWM de 20 a 60 sobres 255 ya la hace. Entonces yo puedo decirle por programación que acelere a la cantidad que quiero ya que lo aguanta, la velocidad es un poco de lo mismo yo decido el valor; por lo tanto este objetivo esta cumplido. Pero no es bueno sobrecargar al motor y driver con unos acelerones tan grandes, mejor hacerlo todo progresivo y aparte de que si el terreno no es bueno puede haber mucho deslizamiento con enormes velocidades.

### **3. Incorporar un mecanismo de captura (ventosa o brazo) capaz de agarrar objetos en menos de 3 segundos:**

Incorpore ambos, una pinza y una ventosa que me agarran los objetos bastante rápido, una vez la pinza se abre y tiene al objeto en su zona de agarre, lo sujeta rápido sin ningún tipo de problema, por lo tanto, este objetivo está cumplido.

### **4. Integrar sensores (ultrasónicos, infrarrojos o Cámara) y sistemas de navegación autónoma eficientes:**

Puse sensores de ultrasonido e infrarrojo y una Cámara ESP32 CAM, es un sistema de navegación completo que sirve más que de sobra para localizar, acercarse y agarrar los objetos. El sistema es sencillo pero a su vez bastante completo, por lo tanto, este objetivo lo doy por cumplido.

### **5. Simular y probar el robot en entornos controlados antes de construir el prototipo físico:**

He realizado pruebas en la placa blanca con Arduino mega 2560, en simuladores como Multisim o Proteus. Por lo tanto, este objetivo lo cumplí.

## 7. CONCLUSIONES

Con este trabajo he podido comprobar que fabricar un robot no es solo teoría, sino que hay que probarlo en la práctica y ahí es donde de verdad salen los problemas. Cada componente que integré motores, sensores, pinza, ventosa, cámara, drivers funcionó bien pero antes tuve que probar de manera individual, y al juntarlos se consiguió un sistema completo y funcional.

Otra cosa que me ha parecido muy importante es mirar los proyectos de años anteriores. Analizando cómo lo hicieron otros, qué errores tuvieron y cómo los solucionaron, uno se da cuenta de que no todo es perfecto desde el principio. Eso me sirvió para no repetir fallos y también para sacar ideas que pude aplicar en mi diseño. Aun así, durante el desarrollo también me encontré con problemas: a veces elegí mal un componente, otras veces el diseño eléctrico no salía como lo esperaba, pero esos errores me obligaron a buscar soluciones y aprender más.

En este proyecto se llevó a cabo un robot completo, se consiguió un diseño muy detallado y viable. Con los esquemas, los cálculos y las simulaciones quedan claro que el prototipo se puede fabricar y que cumpliría con la normativa de la competición. Los objetivos principales se cumplieron: tener un sistema compacto, que pueda moverse de manera autónoma, manipular objetos y hacerlo todo dentro de las reglas del Eurobot 2025.

Si tuviera que repetir un proyecto como este, cambiaría mi manera de trabajar. Lo primero sería diseñar desde el inicio para poder pedir los materiales con tiempo, porque si no luego se hace tarde y no se puede avanzar. Mientras llegan los componentes aprovecharía para trabajar en la parte teórica, y cuando los reciba, los comprobaría y comenzaría con la PCB. El mayor problema de este proceso es que, si algo no funciona, hay que parar hasta recibir otra pieza, lo cual retrasa bastante el desarrollo.

En conclusión, este proyecto me ha servido para saber organizarme y para aprender de verdad, ya que al fabricar el robot y al hacer la PCB y ver que funciona, realmente se ve que estas aplicando los conocimientos de manera real; y entiendes que hace un circuito realmente en la práctica, porque no es lo mismo entenderlo en la teoría que en la práctica.

## 8. BIBLIOGRAFÍA

SÁNCHEZ, L. M. (2010). *EUROBOT Feed the world* . <https://e-archivo.uc3m.es/rest/api/core/bitstreams/e9724a10-9607-4957-a00b-7cbf25239836/content>

Halvorsen, A., Myren, S. R., & Sperre, A. H. (2012). *Eurobot NTNU 2012: Treasure Island*. <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/260734>

Studený, M. (2012). *EUROBOT Treasure Island*. [https://www.vut.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=56433](https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=56433)

Wikheim, E. (2015). *Eurobot Robomovies*  
[file:///C:/Users/Usuario/Downloads/Wikheim\\_master%20\(2\).pdf](file:///C:/Users/Usuario/Downloads/Wikheim_master%20(2).pdf)

Utrilla Bustamante, C. (2016). *EUROBOT THE BEACH BOTS*.  
<https://repositorio.comillas.edu/xmlui/handle/11531/16654>

Mamblona Nieto, C. J. (2021). *EUROBOT SAIL THE WORLD*.  
<https://ebuah.uah.es/dspace/handle/10017/49277>

Balanta Narvaez, C.A. (2023). *EUROBOT THE CHERRY ON THE CAKE*  
<https://zaquan.unizar.es/record/127353>

Díaz García, L. (2018). *INTEGRACIÓN DE VISIÓN ARTIFICIAL EN UN ROBOT INDUSTRIAL*.  
<https://repositorio.comillas.edu/jspui/bitstream/11531/22556/1/TFG-DiazGarciaLucia.pdf>

Gutiérrez Jiménez, P. (2020). *Construcción de un prototipo de robot velocista*.  
<https://biblus.us.es/bibing/proyectos/abreproy/93294/fichero/TFG-3294+GUTI%C3%89RREZ+JIM%C3%89NEZ%2C+PELAYO.pdf>

---

**Trabajo fin de grado**

Jiménez Jiménez, C. (2017). *Desarrollo de un sistema de control de un robot móvil* [Trabajo de fin de grado, Universidad de Valladolid]. UVaDoc. <https://uvadoc.uva.es/bitstream/handle/10324/25547/TFG-P-644.pdf>

Misiego López, A. (2015). *Control de robots autónomos mediante microcontrolador Arduino* (Trabajo de fin de grado, Universidad de Valladolid, Escuela de Ingenierías Industriales). <https://uvadoc.uva.es/bitstream/handle/10324/14612/TFG-P-318.pdf;sequence=1>

Universitaria de Investigación y Desarrollo. (2009). *Ruedas y movimientos: Tipos y aplicaciones* [Presentación de diapositivas]. SlidePlayer. <https://slideplayer.es/slide/5441246/>

Fernández León, S. (2023). *Estudio e implementación de conducción autónoma en un coche robot con LiDAR* [Trabajo Fin de Grado, Universitat Politècnica de Catalunya]. UPCommons. <http://hdl.handle.net/2117/397211>

Lima, P. U., & Ribeiro, M. I. (2022). *Introduction to robotics* [Course handouts]. Instituto Superior Técnico/Instituto de Sistemas e Robótica. [file:///C:/Users/Usuario/Downloads/02\\_IntroRob%20Perception\\_2223%20\(1\).pdf](file:///C:/Users/Usuario/Downloads/02_IntroRob%20Perception_2223%20(1).pdf)

Universidad de Zúrich, (2025), Departamento de Informática. (s.f.). *Virtual Object (Research)*. Robotics and Perception Group. [https://rpg.ifi.uzh.ch/research\\_vo.html](https://rpg.ifi.uzh.ch/research_vo.html)

Salamanca Céspedes, J. E., & Pérez Castillo, J. N. (2022). LIDAR, una tecnología de última generación, para planeación y desarrollo urbano. *Revista Científica y Tecnológica de la Facultad de Ingeniería, Universidad Distrital Francisco José de Caldas*, 13(1), 67-76. <file:///C:/Users/Usuario/Downloads/Dialnet-LIDARUnaTecnologiaDeUltimaGeneracionParaPlaneacion-4797146.pdf>

Pascual Vega, A. (2019). Comparativa entre motores Brushed y Brushless. <https://openaccess.uoc.edu/server/api/core/bitstreams/44c527b3-2c1f-4f20-9541-bf5567c40a14/content>

IGNACIO IRANZO, J. (2021). DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL PARA MOTORES DE CONTINUA MEDIANTE UN MICROCONTROLADOR ARDUINO.

<https://riunet.upv.es/server/api/core/bitstreams/7872bc0a-70e7-4c7d-95be-b53313197ff1/content>

Malo Yagüe, L. (2011). *Control de un motor paso a paso* [Proyecto Fin de Carrera, Universidad de Zaragoza].

<https://zagan.unizar.es/record/5655/files/TAZ-PFC-2011-107.pdf>

López Morales, J., & Morales Ceballos, J. A. (2014). *Módulo de entrenamiento para el control de posición y velocidad de servomotores mediante PLC con interfaz SCADA* [Trabajo Fin de Máster, Máster en Control de Procesos Industriales].

<https://helvia.uco.es/bitstream/handle/10396/13477/TFM%2520Jos%25C3%25A9%2520Antonio%2520Morales%2520Ceballos%2520y%2520Javier%2520L%25C3%25B3pez%2520Morales.pdf?isAllowed=y&sequence=1>

Panigrahi, P. K., & Bisoy, S. K. (2021). Localization strategies for autonomous mobile robots: A review. *Journal of King Saud University*.

[https://www.sciencedirect.com/science/article/pii/S1319157821000550?utm\\_source.com](https://www.sciencedirect.com/science/article/pii/S1319157821000550?utm_source.com)

sunFields. (2025, 29 de abril). *Tipo de baterías ion litio*. <https://www.sfe-solar.com/baterias/ion-litio/tipos/?srsltid=AfmBOoohSN9xkda4mFLyBTgz93kOPwgxhj702Qgu8KSltkj-ZxOpb9bi>

<https://concepto.de/bateria/>

MatWeb. (s.f.). *Material Property Data*. Recuperado el 26 de junio de 2025, de <https://www.matweb.com>

Seok, S., Wang, A., Chuah, M. Y. M., Hyun, D. J., Lee, J., Otten, D. M., Lang, J. H., & Kim, S. (2015). *Design principles for energy-efficient legged locomotion and implementation on the MIT Cheetah robot*. *IEEE/ASME Transactions on Mechatronics*, 20(3), 1117–1129.

[https://dspace.mit.edu/handle/1721.1/108096?utm\\_source.com](https://dspace.mit.edu/handle/1721.1/108096?utm_source.com)

Ijspeert, A. J. (2008). *Central pattern generators for locomotion control in animals and robots*.

<https://www.sciencedirect.com/science/article/abs/pii/S0893608008000804?via%3Dihub>

El País. (2025, 29 de mayo). *Robots con patas que juegan bádminton y hacen parkour logran coordinar percepción y movimiento*. El País.

IBM. (2021, 27 de julio). *What is computer vision?* IBM Think.  
<https://www.ibm.com/think/topics/computer-vision>

Cognex. (s.f.). *Types of machine vision systems*. Cognex.  
<https://www.cognex.com/es-mx/what-is/machine-vision/types-of-machine-vision-systems>

Santos, D., Dallos, L., & Gaona-García, P. (2020). *Algoritmos de rastreo de movimiento utilizando técnicas de inteligencia artificial y machine learning*.  
[https://www.scielo.cl/scielo.php?pid=S0718-07642020000300023&script=sci\\_arttext&utm\\_source.com](https://www.scielo.cl/scielo.php?pid=S0718-07642020000300023&script=sci_arttext&utm_source.com)

Robotics Design Handbook. (2020). *Robust Robotics: ROBOCON Robot Design Handbook* (pp. 291–300).  
<https://www.scribd.com/document/707923561/ROBOCON-2020-Robot-Design-Handbook>

International Journal of Science, Engineering and Management. (2024, agosto). *DC Motor Speed Control Using PI and PID Controller*. IJSEM.  
<https://ijsem.org/article/14%20August%202024%20IJSEM.pdf>

Wei, Y. (2024). *Applications of Ultrasonic Sensors: A Review*. *Proceedings of the 5th International Conference on Signal Processing and Machine Learning*.  
[https://www.researchgate.net/publication/386118233\\_Applications\\_of\\_Ultrasonic\\_Sensors\\_A\\_Review](https://www.researchgate.net/publication/386118233_Applications_of_Ultrasonic_Sensors_A_Review)

Chen, C. et al. (2023). *IR Sensor Models and Experimental Validation*. *Sensors*, 23(1), 44. <https://www.mdpi.com/1424-8220/23/1/44>

Kuczmann, M. (2024). *Review of DC Motor Modeling and Linear Control: Theory with Laboratory Tests*. *Electronics*, 13(11), 2225.  
<https://www.mdpi.com/2079-9292/13/11/2225>

Autsou, S., Kudelina, K., Vaimann, T., Rassölkin, A., & Kallaste, A. (2024). *Principles and Methods of Servomotor Control: Comparative Analysis and Applications*. *Applied Sciences*, 14(6), 2579. <https://www.mdpi.com/2076-3417/14/6/2579>

Trabajo fin grado

**Vanneaux, E. (2024).** *Multiple Input Multiple Output system control (CM4)*. ENSTA Paris. Recuperado de [https://perso.ensta-paris.fr/~manzaner/Cours/AUT202/AUT202\\_CM4.pdf](https://perso.ensta-paris.fr/~manzaner/Cours/AUT202/AUT202_CM4.pdf)

Thrun, S., Fox, D., & Burgard, W. (1999-2000). *Probabilistic Robotics* <https://docs.ufpr.br/~danielsantos/ProbabilisticRobotics.pdf>

Chen, P.; Zhao, X.; Zeng, L.; Liu, L.; Liu, S.; Sun, L.; Li, Z.; Chen, H.; Liu, G.; Qiao, Z.; et al. (2025). *A Review of Research on SLAM Technology Based on the Fusion of LiDAR and Vision*. *Sensors*, 25(5), 1447. <https://doi.org/10.3390/s25051447>

Koenig, S., & Likhachev, M. (2005). *Fast replanning for navigation in unknown terrain*. *IEEE Transactions on Robotics*. <https://doi.org/10.1109/TRO.2004.838026>

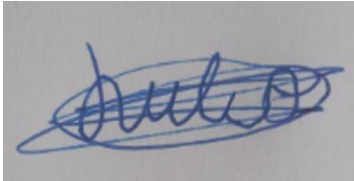
### **Relación de documentos**

( ) Memoria 149 páginas

( ) Anexos 87 páginas

Aranjuez, a 14 de 09 de 2025

Firmado:



Julio Cesar Balanta Narvaez