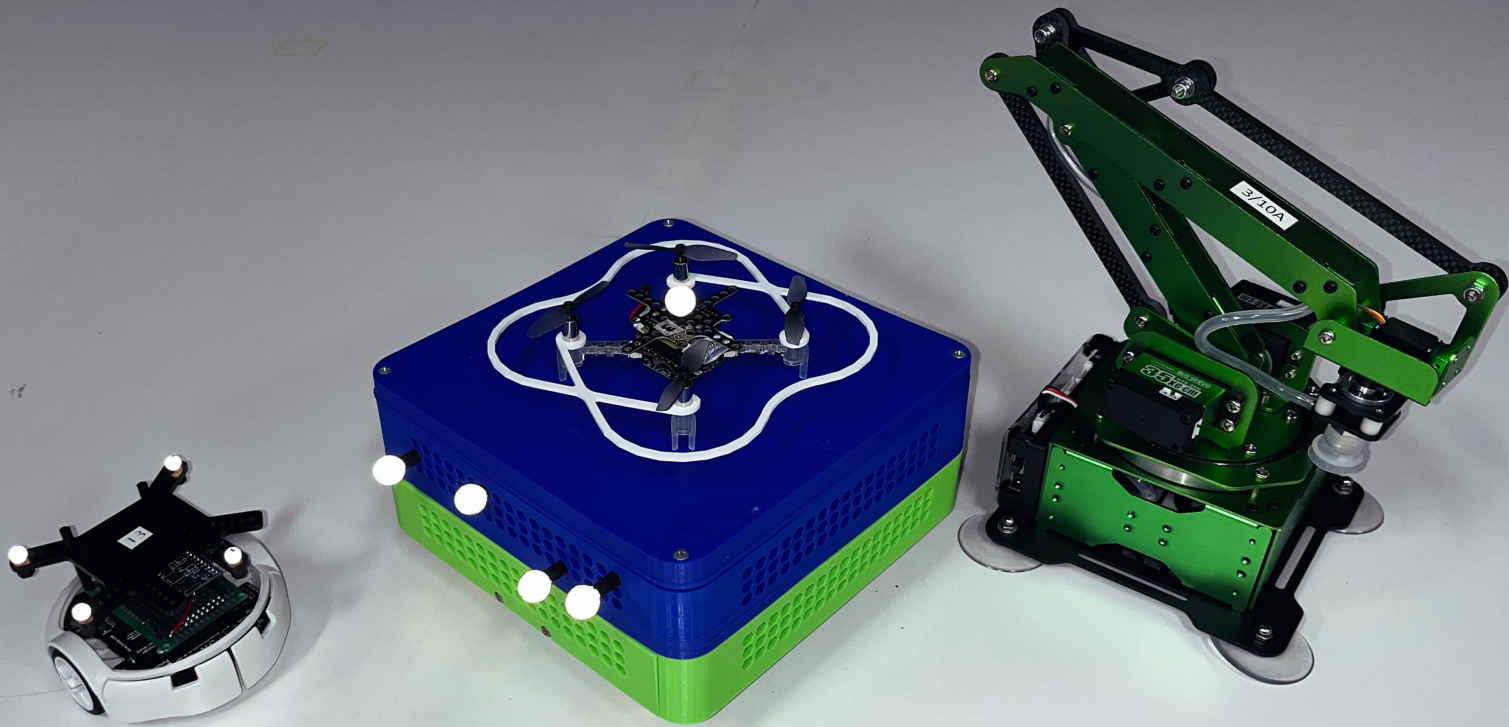


---

# Diseño e implementación de una plataforma móvil omnidireccional para el desarrollo de aplicaciones dentro del ecosistema Robotat

---

Christian Alessandro Campos López





UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Diseño e implementación de una plataforma móvil  
omnidireccional para el desarrollo de aplicaciones dentro del  
ecosistema Robotat**

Trabajo de graduación presentado por Christian Alessandro Campos  
López para optar al grado académico de Licenciado en Ingeniería  
Mecatrónica

Guatemala,

2025



UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Diseño e implementación de una plataforma móvil  
omnidireccional para el desarrollo de aplicaciones dentro del  
ecosistema Robotat**

Trabajo de graduación presentado por Christian Alessandro Campos  
López para optar al grado académico de Licenciado en Ingeniería  
Mecatrónica

Guatemala,

2025



Vo.Bo.:

(f)   
M.Sc. Miguel Enrique Zea Arenales

(f)   
M.Sc. Carlos Alberto Esquit Hernández

Fecha de aprobación: Guatemala, 28 de noviembre de 2025.

Todos sabemos que parte del viaje es el final, y este trabajo representa la culminación de un recorrido de cinco años que reunió aprendizaje, amistad, noches de desvelo y pequeñas grandes alegrías. Agradezco profundamente a la Fundación Juan Bautista Gutiérrez por creer en mí dandome la oportunidad de vivir este viaje. Extiendo también mi gratitud a los profesores que dedicaron su tiempo y vocación a mi formación. Hago una mención especial a mi asesor, Miguel Zea, por su acompañamiento durante este proyecto y por las lecciones brindadas, y a Magda Moscoso, por sus enseñanzas, su confianza y apoyo durante estos años.

A mis amigos: Marcela, Gustavo, Diana, Angie, Melissa, Bryan, Ale, Victor y Pablo, les agradezco los momentos de risa, de cansancio y de tristeza compartidos. A Marre y Camila, cada una, con sus diferencias y similitudes, aportó algo invaluable a mi vida. Les agradezco por estos últimos años que fueron, sin duda alguna, lo mejor que me pasó en la universidad. A Cruz, más que a nadie, gracias por tu amistad, por tu compañía y por todas las horas de trabajo que atravesamos juntos.

A Briana, te debo un agradecimiento profundo por tu paciencia y amor. Viviste mi estrés, mis desvelos, mi preocupación y mi cansancio. Sin tu apoyo incondicional esta meta hubiera sido mucho más difícil de alcanzar. Agradezco también a mi padre por su respaldo constante, y a mi hermano por ser un ejemplo a seguir. Asimismo agradezco al resto de mi familia, quienes me permitieron recargar fuerzas y continuar adelante. A mis mascotas: Frankie, Kira y Lucky, las recuerdo por la compañía que me brindaron en momentos difíciles. Y a mis abuelitos, Papá Q y Vilma, quienes no tuvieron la oportunidad de ver el final de este camino, los llevo conmigo en cada logro.

Finalmente, a mi madre, por tu esfuerzo incansable. Sé que mi abuelito estaría orgulloso de mí, pero también sé que estaría orgulloso de ti. Porque entendería que este logro es posible gracias a la grandiosa madre que eres y siempre has sido. No se suponía que yo estuviera aquí, pero gracias a ti, llegamos juntos a este punto. Este logro es tanto tuyo como mío.

Más allá de las fórmulas, los circuitos y los proyectos, aquello que permanecerá conmigo son las personas que le dieron sentido a este recorrido y que hicieron que cada hora invertida valiera la pena. Dedico esta tesis a quienes formaron parte de este viaje y que, con su presencia, me recordaron por qué valió la pena llegar hasta aquí. Gracias por estar ahí.

<b>Prefacio</b>	<b>I</b>
<b>Índice de figuras</b>	<b>IV</b>
<b>Índice de cuadros</b>	<b>V</b>
<b>Resumen</b>	<b>VI</b>
<b>Abstract</b>	<b>VII</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Antecedentes</b>	<b>2</b>
<b>3. Justificación</b>	<b>4</b>
<b>4. Objetivos</b>	<b>5</b>
4.1. Objetivo general . . . . .	5
4.2. Objetivos específicos . . . . .	5
<b>5. Alcance</b>	<b>6</b>
<b>6. Marco teórico</b>	<b>7</b>
6.1. Análisis de decisiones multi-criterio (MCDA) . . . . .	7
6.2. Motor paso a paso Nema 17 . . . . .	9
6.3. ESP32 . . . . .	10
6.4. Baterías de Li-ion . . . . .	12
6.5. Robots móviles omnidireccionales de cuatro ruedas . . . . .	12
<b>7. Estudio comparativo para la selección de componentes</b>	<b>15</b>
7.1. Criterios de selección del modelo de los motores . . . . .	15
7.2. Alternativas para la selección de los motores . . . . .	16
7.3. Aplicación del método y selección final de los motores . . . . .	17
7.4. Criterios de selección del controlador físico del motor . . . . .	18

7.5.	Alternativas del controlador físico del motor . . . . .	19
7.6.	Aplicación del método y selección final del controlador físico del motor . . . . .	19
<b>8.</b>	<b>Diseño y manufactura electromecánica de la plataforma omnidireccional</b>	<b>21</b>
8.1.	Diseño electrónico . . . . .	21
8.2.	Diseño mecánico . . . . .	24
8.3.	Manufactura e integración de componentes locales en la plataforma móvil . . . . .	26
<b>9.</b>	<b>Integración de la plataforma móvil al ecosistema Robotat</b>	<b>31</b>
9.1.	Módulos de las tareas dentro del ESP32 ( <i>FreeRTOS</i> ) . . . . .	31
9.2.	Control de los motores en lazo abierto . . . . .	32
9.3.	Control de los motores en lazo cerrado . . . . .	33
9.4.	Comunicación con el ecosistema Robotat . . . . .	36
9.5.	Configuración dentro del Robotat . . . . .	38
9.6.	Control de trayectorias . . . . .	38
<b>10.</b>	<b>Conclusiones</b>	<b>43</b>
<b>11.</b>	<b>Recomendaciones</b>	<b>44</b>
<b>12.</b>	<b>Referencias</b>	<b>45</b>
<b>13.</b>	<b>Anexos</b>	<b>49</b>
13.1.	MCDA . . . . .	49
13.2.	Tamaño de las pistas PCB basado en el estándar IPC-2221A . . . . .	54
13.3.	Llantas omnidireccionales impresas en 3D . . . . .	55
13.4.	Registros y configuración del AS5600 . . . . .	56
13.5.	Estadísticas de las trayectorias . . . . .	57
13.6.	Repositorio de GitHub . . . . .	58
13.7.	Vídeos . . . . .	58
13.8.	Planos y manuales de la plataforma móvil . . . . .	58

---

## Índice de figuras

---

1.	Plataformas móviles dentro del Robotat . . . . .	3
2.	Ejemplos de robots omnidireccionales . . . . .	3
3.	Diagrama de conexión recomendada del DRV8825 . . . . .	9
4.	Encoder magnético AS5600 . . . . .	10
5.	ESP32 . . . . .	11
6.	Representación de marcos de referencia y ruedas mecanum . . . . .	13
7.	Modelo cinemático de un robot omnidireccional de 4 ruedas . . . . .	14
8.	Resultado del MAUT/MAVT para la selección de los motores . . . . .	17
9.	Resultado final del MAUT/MAVT para el control de los motores . . . . .	19
10.	Esquemático del circuito de alimentación . . . . .	22
11.	Diagrama de bloques del funcionamiento de la plataforma móvil . . . . .	23
12.	Esquemático de circuito de control . . . . .	24
13.	Niveles de la plataforma móvil en <i>Fusion 360</i> . . . . .	25
14.	Diseño de las ruedas omnidireccionales . . . . .	26
15.	Ensamble completo . . . . .	27
16.	Impresión 3D del chasis . . . . .	28
17.	Manufactura placas PCB . . . . .	28
18.	Diagrama de control en lazo abierto de los motores . . . . .	32
19.	Diagrama de control en lazo cerrado de los motores . . . . .	33
20.	Pseudocódigo de control del robot . . . . .	37
21.	Ubicación de los marcadores en los laterales de la plataforma . . . . .	38
22.	Comparativa trayectorias . . . . .	40
23.	Comparativa trayectoria del mapa . . . . .	41
24.	Error de posición durante las trayectorias . . . . .	42
25.	Placa del circuito de alimentación . . . . .	55
26.	Placa para el circuito de control . . . . .	55
27.	Impresión 3D de las llantas . . . . .	56

---

## Índice de cuadros

---

1.	Matriz de comparación de los motores . . . . .	16
2.	Matriz de comparación de los controladores físicos . . . . .	18
3.	Presupuesto de potencia del circuito de alimentación . . . . .	22
4.	<i>Bill of materials</i> : precios . . . . .	29
5.	<i>Bill of materials</i> : distribuidores . . . . .	30
6.	Error velocidad en lazo abierto . . . . .	35
7.	Error velocidad en lazo cerrado . . . . .	35
8.	Comandos y funciones del sistema . . . . .	36
9.	Análisis de trayectorias . . . . .	42
10.	Comparativa de motores . . . . .	51
11.	Comparativa de controladores para motores paso a paso . . . . .	52
12.	Análisis comparativo de los modelos de motores . . . . .	53
13.	Análisis comparativo de los modelos de motores . . . . .	54
14.	Grosor de las pistas . . . . .	54
15.	Registros importantes del AS5600 . . . . .	57

Este trabajo presenta el diseño, implementación e integración de una plataforma móvil omnidireccional dentro del ecosistema Robotat de la Universidad del Valle de Guatemala, capaz de seguir trayectorias e integrarse con el sistema de captura de movimiento y otros agentes robóticos. El proyecto surge de la necesidad de contar con una herramienta local, accesible y reproducible para experimentación en control y docencia.

La metodología incluyó un estudio comparativo para seleccionar actuadores y controladores, el diseño y manufactura digital de la plataforma, el desarrollo de la electrónica basada en el microcontrolador ESP32 y la implementación de algoritmos de control en lazo cerrado vía Wi-Fi. Se realizaron pruebas experimentales de seguimiento de trayectorias y evasión de obstáculos dentro del Robotat, obteniendo un RMSE entre 5.11 mm y 36.91 mm y un valor mínimo de error máximo de 10.19 mm. Los resultados incluyen un prototipo funcional, código fuente, planos CAD, manuales de usuario y un listado de materiales con proveedores locales.

**Palabras clave:** diseño, plataforma omnidireccional, robótica móvil, control en lazo cerrado, esp32.

This paper presents the design, implementation and integration of an omnidirectional mobile platform to operate within the Robotat ecosystem of the Universidad del Valle de Guatemala, capable of trajectory tracking and interoperability with the motion-capture system and other robotic agents. The project addresses the need for a local and reproducible tool for experimentation in control and education.

The methodology included a comparative analysis for selecting actuators and drivers, a digital design and manufacturing, the development of electronics based on the ESP32 microcontroller, and the implementation of closed-loop control algorithms with Wi-Fi. Experimental tests of trajectory tracking and obstacle avoidance were carried out within the Robotat, obtaining a RMSE range from 5.11 mm to 36.91 mm, with the smallest maximum-error value measured at 10.19 mm. The results include a functional prototype, source code, CAD drawings, user manuals and a bill of materials with local suppliers.

**Keywords:** design, omnidirectional platform, mobile robotics, closed loop control, esp32.

El ecosistema Robotat de la Universidad del Valle de Guatemala proporciona infraestructura de captura de movimiento y espacios de prueba para diferentes agentes robóticos. Sin embargo, el mismo carece de una plataforma diseñada específicamente para integrarse de forma directa con dicho ecosistema y servir como base para el desarrollo de aplicaciones e investigaciones como un agente secundario.

En este trabajo se presenta el diseño, la implementación y la integración de una plataforma móvil omnidireccional concebida para operar dentro del entorno Robotat. La metodología incluyó un análisis multicriterio para la selección de componentes, un diseño y prototipado desarrollado bajo criterios de modularidad y reproducibilidad, priorizando componentes disponibles en el mercado local y técnicas de manufactura digital como la impresión 3D y el diseño de placas de circuito impreso (PCB). Asimismo, se desarrolló la electrónica basada en el microcontrolador ESP32 y la implementación de comunicación vía Wi-Fi para la interoperabilidad con el sistema de captura del ecosistema Robotat y otros agentes robóticos. La validación experimental consistió en pruebas de seguimiento de trayectorias y evasión de obstáculos dentro ecosistema Robotat, en las que se registraron métricas de desempeño posicional.

El documento describe el proceso de diseño y manufactura, las evaluaciones realizadas y los entregables del proyecto: un prototipo funcional, código fuente documentado, planos CAD, un manual de usuario y una lista de materiales con proveedores locales. En las secciones de resultados y pruebas se analiza el comportamiento de la plataforma en tareas de seguimiento de trayectorias y se presentan las limitaciones técnicas y logísticas detectadas durante el desarrollo. También se comparten consideraciones prácticas para reproducir el prototipo en el entorno académico y posibles trabajos futuros que se podrían realizar con la plataforma móvil.

Cuando se habla de proyectos de robótica realizados dentro de la UVG, se debe comenzar hablando del ecosistema de experimentación robótico denominado Robotat. Este laboratorio institucional está orientado a la realización de aplicaciones e investigaciones con diversos agentes robóticos y provee la infraestructura necesaria para evaluar algoritmos y plataformas en un entorno controlado [1].

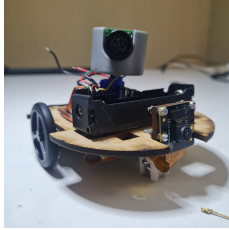
El ecosistema Robotat dispone de 6 cámaras *OptiTrack* que proporcionan la medida espacial de referencia (pose) de los agentes sobre el área de ensayo. Donde el proceso operativo según [2] es el siguiente: las cámaras emiten y detectan radiación infrarroja para observar los marcadores reflectantes colocados en los agentes. Esto genera varias vistas en 2D que, mediante la calibración geométrica de las cámaras, triangulan las posiciones en 3D de cada marcador en el sistema de coordenadas del laboratorio. Además, conociendo los arreglos geométricos de los marcadores para cada cuerpo rígido, el *software* estima la pose completa (posición y orientación) mediante un ajuste que minimiza el error de correspondencia, transmitiendo en tiempo real la pose estimada al resto del sistema, permitiendo su uso en control, visualización y registro de datos [2].

Gracias a la implementación de este sistema de captura, en la UVG se ha buscado implementar diversas plataformas móviles propias en tres modalidades distintas. La primera fue parte de un mega-proyecto colaborativo [3], la segunda modalidad fue desarrollada por Lima [4] y la tercera por Rodríguez [5]. Estos trabajos tuvieron como objetivo diseñar e implementar una plataforma móvil de bajo costo y dimensiones reducidas, con capacidad de sensor su entorno. En todos los casos se mantuvo un sistema de locomoción basado en tres ruedas: una rueda libre y dos ruedas controladas con motores DC (Figura 1a).

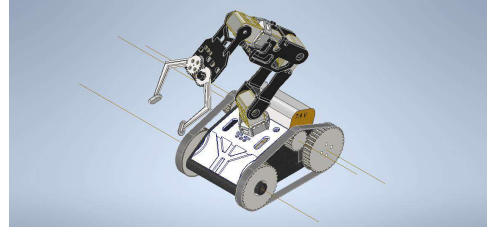
Adicionalmente, dentro de la UVG también se trabajó en una plataforma móvil para su implementación dentro del sistema de captura de movimiento del Robotat. Este fue un trabajo de Diego González [6]. Lo más destacado de este trabajo fue el proceso de diseño y manufactura, del cual podemos destacar el sistema de locomoción que se hizo a través de un sistema mecánico de bandas (Figura 1b).

**Figura 1.** Plataformas móviles dentro del Robotat

(a) Robot para robótica de enjambre.



(b) Diseño final de la plataforma móvil [6].



Nota. Robot para robótica de enjambre 1a obtenido de [5] y diseño final de la plataforma móvil 1b obtenido de [6].

Fuera del ámbito universitario local existen múltiples desarrollos de plataformas móviles con enfoques diversos. Por el enfoque que se escogió para este trabajo, la búsqueda se centró en aquellos que usaron ruedas omnidireccionales como sistema de locomoción. Un ejemplo relevante es el *AxeBot*, desarrollado por Paim, Costa y Nascimento [7], un robot móvil autónomo que buscaba competir en la *RoboCup Small Size League*. El proyecto se dividió en cuatro partes: el diseño del chasis, el diseño de las ruedas, el diseño del dispositivo de disparo y el ensamblaje general. Lo destacado de este proyecto es que contaba con 3 ruedas omnidireccionales, sensores, servomotores, motores DC, baterías y un sistema informático para el control de todos sus dispositivos con algoritmos propios.

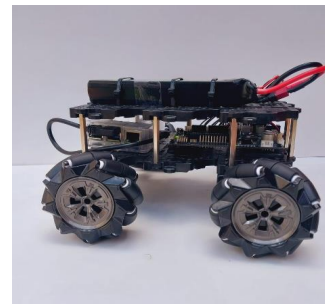
Por otro lado, en la Universidad del Azuay en Ecuador, Brayan Castillo [8] en su trabajo de graduación, buscó la implementación y análisis de algoritmos de control de una plataforma robótica omnidireccional. En dicho robot se utilizaron cuatro ruedas omnidireccionales; este diseño tenía el objetivo de lograr movimientos instantáneos en cualquier dirección con pocas maniobras, pues el enfoque del proyecto era el del control de la trayectoria del robot, permitiendo realizar trayectorias preestablecidas buscando el mínimo error posible.

**Figura 2.** Ejemplos de robots omnidireccionales

(a) *AxeBot*.



(b) Plataforma móvil omnidireccional para algoritmos de control.



Nota. El *AxeBot* (a) obtenido de [7] y el Robot omnidireccional (b) obtenido de [8].

Una de las aplicaciones más relevantes en la robótica moderna es el desarrollo de plataformas móviles versátiles, capaces de desplazarse de manera eficiente en distintos entornos. Estas plataformas pueden actuar como agentes secundarios para complementar aplicaciones que realizan otros robots.

La UVG ha trabajado durante años con diferentes sistemas robóticos. Estos han abarcado desde manipuladores seriales para aplicaciones de *pick and place*; el control de drones, una infraestructura segura para los mismos; robots exploradores para navegar en terrenos irregulares; y robots que buscan usar algoritmos de vehículos autónomos o robótica de enjambre. Sin embargo, actualmente no existe una plataforma móvil que permita ampliar la funcionalidad de dichos robots o servir como un agente secundario para estas aplicaciones.

Por esto mismo, se busca desarrollar una plataforma móvil omnidireccional, capaz de integrarse dentro del sistema de captura de movimiento del Robotat. Esto le permitiría, en un futuro, su implementación en otros proyectos: en la navegación autónoma, actuando como obstáculo; funcionando como plataformas móviles de aterrizaje para los drones y para convertir los robots estáticos seriales en manipuladores móviles. Para ello se utilizarán componentes disponibles localmente y en la universidad, además de incluir impresión 3D para reducir costos y facilitar la fabricación.

### 4.1. Objetivo general

Diseñar e implementar una plataforma móvil omnidireccional capaz de operar dentro del sistema de captura de movimiento del Robotat.

### 4.2. Objetivos específicos

- Implementar la conectividad con el ecosistema Robotat para permitir el seguimiento de la plataforma en un entorno de prueba controlado.
- Diseñar el sistema electromecánico de la plataforma modular con 4 ruedas omnidireccionales que tenga un movimiento y consumo energético eficiente.
- Desarrollar una plataforma móvil que utilice componentes accesibles localmente y manufactura digital ágil y de bajo costo
- Realizar un estudio comparativo de motores, algoritmos de control y drivers para seleccionar la configuración de la plataforma móvil.

En este trabajo se abordó el diseño, la implementación y la integración de una plataforma móvil omnidireccional destinada a operar dentro del ecosistema Robotat de la UVG. El propósito principal fue entregar un prototipo funcional y reproducible con componentes accesibles en el entorno académico local, que sirviera como plataforma experimental para pruebas de control y como agente secundario en proyectos de robótica y experimentos colaborativos en la universidad. Este prototipo recibió el nombre *OmniBotCC* durante su desarrollo.

Las limitaciones principales del proyecto estuvieron determinadas por recursos externos: acceso restringido al sistema Robotat y a sus horarios de uso, presupuesto orientado a prototipado académico y la disponibilidad de componentes en el mercado local. Además, ciertas características de desempeño dependieron de la selección final de actuadores y de la fuente de alimentación, lo cual condicionó parámetros como capacidades dinámicas y autonomía del prototipo.

Fuera del alcance del presente trabajo quedaron explícitamente tareas de mayor complejidad o que podrían requerir recursos adicionales: el desarrollo completo de algoritmos avanzados de navegación autónoma más allá de las validaciones de movimiento omnidireccional; validaciones en entornos externos al Robotat; el diseño para producción en serie y certificaciones comerciales y el desarrollo de subsistemas no esenciales para la operación básica de la plataforma. Estas actividades se identificaron como posibles trabajos futuros.

En el desarrollo de este trabajo se utilizaron herramientas de inteligencia artificial como apoyo en redacción, revisión de estilo (ChatGPT [45]) y en correcciones y sugerencias de código (versión de Copilot para GitHub [46]), buscando mantener siempre la autoría y análisis propio.

El marco teórico presentado a continuación contiene los fundamentos necesarios para comprender las decisiones de diseño adoptadas en el desarrollo de la plataforma móvil omnidireccional. Por ello, se incluyen métodos de análisis de decisiones multicriterio, que facilitaron la comparación estructurada de alternativas y la selección de los componentes más adecuados para el proyecto. Además, se presentan los conceptos relacionados con los motores, sensores y el microcontrolador empleados, junto con los principios de cinemática, control y planificación de movimiento de plataformas omnidireccionales. Estos elementos permitieron modelar el comportamiento del robot y definir las estrategias que permitieron su integración y operación dentro del ecosistema Robotat.

## 6.1. Análisis de decisiones multi-criterio (MCDA)

El Análisis de Decisiones Multi-Criterio (MCDA por sus siglas en inglés *Multi-Criteria Decision Analysis*) es un conjunto de métodos orientados a la toma de decisiones cuando existen múltiples criterios de evaluación. Este enfoque permite comparar alternativas considerando simultáneamente criterios cuantitativos y cualitativos y, de este modo, apoyar la selección de la solución más adecuada al problema planteado [9].

### 6.1.1. Pasos a seguir al realizar un MCDA

Siguiendo la estructura de [10] la secuencia a seguir es:

1. **Identificar el problema:** definir de forma general el problema a resolver.
2. **Estructurar el problema:** plantear las posibles alternativas y los criterios de evaluación (cuantitativos o cualitativos).

3. **Construcción del modelo:** asignar valores a cada alternativa según los criterios definidos y dar un peso relativo a cada criterio.
4. **Aplicación del modelo y elección:** calcular la puntuación de cada alternativa combinando las calificaciones con los pesos. Terminando con la selección de una alternativa.

Es importante aclarar que la decisión final puede no depender exclusivamente de la puntuación numérica. Queda en el criterio del evaluador determinar la mejor decisión con base en la información obtenida en el estudio comparativo.

### 6.1.2. Teoría de Utilidad (Valor) Multiatributo (MAUT/MAVT)

Esta teoría (MAUT/MAVT por sus siglas en inglés *Multi-Attribute Utility/Value Theory*) transforma las distintas unidades de los criterios (coste, impacto ambiental, etc.) en un valor dentro de un rango normalizado [10]. Para cada criterio  $i$  se define una función de valor  $v_i(a)$  que mapea el desempeño de la alternativa  $a$  en el criterio  $i$  a un rango normalizado. Un peso  $w_i$  que refleja la importancia relativa del criterio dentro del problema, con  $\sum_{i=1}^n w_i = 1$ .

Según [10] la puntuación final de cada alternativa  $a$  se obtiene como:

$$V(a) = \sum_{i=1}^n w_i v_i(a) \quad (1)$$

La mejor alternativa es la que maximiza el valor de la puntuación  $V(a)$ . La calidad del resultado depende en gran medida de la definición de las funciones  $v_i(a)$  y de la asignación de pesos  $w_i$ , por lo que la selección y justificación de estas funciones son pasos críticos del método [10].

### 6.1.3. Proceso Analítico Jerárquico (AHP)

Este submétodo (AHP por sus siglas en inglés *Analytic Hierarchy Process*) utiliza comparaciones pareadas para la ponderación de los criterios [10]. En la matriz de comparaciones  $A = [a_{ij}]_{n \times n}$  se tiene:

$$a_{ij} = \text{grado en que el criterio } i \text{ domina a } j, \quad a_{ji} = \frac{1}{a_{ij}}, \quad a_{ii} = 1.$$

El AHP concluye calculando el autovector que representa los pesos de los criterios. Partiendo de la matriz de comparaciones pareadas el autovector principal  $w$  y su autovalor asociado  $\lambda_{\text{máx}}$  satisfacen la ecuación [10]:

$$A w = \lambda_{\text{máx}} w. \quad (2)$$

Para obtener  $w$  y comprobar que los resultados son consistentes se procede así:

1. Resolver numéricamente la ecuación de autovalores de  $A$  y tomar el autovector correspondiente al mayor autovalor  $\lambda_{\text{máx}}$ .
2. Normalizar  $w$  para que sus componentes sumen 1.
3. Comprobación de consistencia para garantizar que las comparaciones no sean excesivamente inconsistentes usando el índice de consistencia [10].

$$\text{CI} = \frac{\lambda_{\text{máx}} - n}{n - 1} < 0.1. \quad (3)$$

## 6.2. Motor paso a paso Nema 17

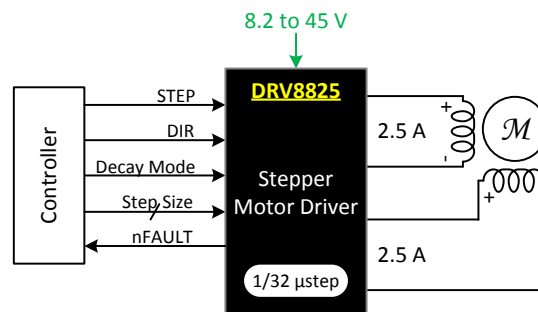
Una vez establecido el marco metodológico para la toma de decisiones mediante MCDA, se procedió a aplicarlo en la selección de los componentes críticos del sistema. Por lo que a continuación se encuentra la información necesaria para comprender su funcionamiento y comprender ciertas configuraciones, que resultaron ser las más adecuadas para la plataforma desarrollada.

El Nema 17 es un motor paso a paso bipolar con 200 pasos por revolución ( $1.8^\circ$  por paso). La designación 17 hace referencia a su tamaño principal (1.7 pulgadas o 43 mm), pero teniendo longitudes entre 23 mm y 63 mm [11]. Al ser máquinas sin escobillas, requieren controladores que conmutan y alimentan los devanados en el orden y tiempo adecuados. Permitiendo incrementos angulares muy pequeños en cada uno de sus pasos [12].

Gracias a esa precisión intrínseca, los Nema 17 son adecuados para aplicaciones que demandan movimientos repetibles en configuración *open-loop*. Sin embargo, cuando se necesita mayor fiabilidad ante pérdidas de paso o un mejor rendimiento dinámico, se suele complementar con retroalimentación (*close-loop*) mediante sensores magnéticos, lo que mejora la exactitud y la respuesta del sistema.

### 6.2.1. Controlador DRV8825

**Figura 3.** Diagrama de conexión recomendada del DRV8825



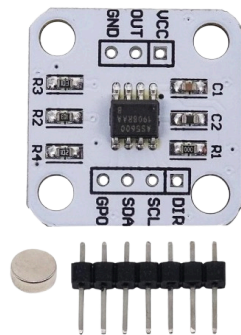
Nota. Este diagrama muestra una forma común de conexión del controlador DRV8825 para motores paso a paso. Este diagrama fue obtenido de [13].

El DRV8825 es un controlador que permite manejar motores paso a paso bipolares con hasta 45 V y 2.5 A por bobina (con disipador de calor integrado o una ventilación adecuada) y tiene la capacidad de realizar micropasos con un valor de hasta 1/32 de paso [13] y puede ser controlado con un microcontrolador con señales de 3.3 V o 5 V. La Figura 3 muestra la conexión recomendada de los pines del controlador. Las especificaciones de las funciones de los pines (alimentación, STEP, DIR, ENABLE y pines MS1–MS3 para microstepping) se describe en la documentación del fabricante [13].

### 6.2.2. AS5600

El AS5600 (Figura 4) es un sensor magnético de posición rotacional de 12 bits. Este funciona principalmente con comunicación I<sup>2</sup>C tanto para su configuración como para la lectura de sus datos. Sin embargo, gracias a las opciones de configuración dentro de sus registros I<sup>2</sup>C, se puede utilizar el pin OUT como señal analógica o PWM. Los pasos a seguir para su configuración se encuentran especificados en la documentación del fabricante [14]. Este sensor permite medir y controlar la posición de motores paso a paso con alta resolución, proporcionando la retroalimentación necesaria para implementar operación en *close-loop*.

**Figura 4.** Encoder magnético AS5600



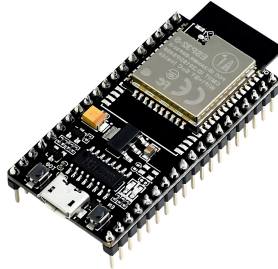
Nota. Imagen obtenida de [15].

### 6.3. ESP32

Tras presentar los motores, el controlador y el encoder, resulta necesario introducir el microcontrolador responsable de coordinar estos componentes. El ESP32 (Figura 5) es un microcontrolador de alto rendimiento con conectividad Wi-Fi, Bluetooth y Bluetooth LE. Cuenta con un procesador de doble núcleo que opera hasta 240 MHz, con 30 pines accesibles (24 de ellos digitales), múltiples puertos de entrada/salida y soporte para los protocolos de comunicación SPI, I<sup>2</sup>C y UART. Funciona con un voltaje recomendado mínimo 3.0 V y máximo 3.6 V o bien con 5 V si la placa integra un regulador, y requiere una corriente mínima de 0.5 A, soportando hasta 1.1 A en total (con un máximo de 40 mA por salida). Incorpora un módulo Wi-Fi de 2.4 GHz con transmisor y receptor con calibraciones digitales para corregir errores de modulación y mejorar la calidad de la señal [16]. Este controlador

puede programarse en múltiples entornos como Arduino IDE, MicroPython, PlatformIO y el SDK de Espressif.

**Figura 5.** ESP32



Nota. Imagen obtenida de [17].

### 6.3.1. *lwIP*-UDP

ESP-IDF, el SDK de Espressif para el ESP32, integra librerías de código abierto como *lwIP*, un *stack* ligero TCP/IP para sistemas embebidos. Para comunicación UDP (*User Datagram Protocol*), se utiliza la API tipo *BSD sockets*, la cual es no orientada a conexión y no garantiza entrega ni orden, pero resulta ideal en aplicaciones de baja latencia como telemetría, control en tiempo real o difusión (*broadcast/multicast*) [18]. El uso de UDP en *lwIP* implica crear un *socket* con la familia y protocolo adecuados, asociarlo a un puerto local y luego recibir datos en modo bloqueante o con tiempo de espera. Finalmente, se debe cerrar el *socket* para liberar recursos, y en aplicaciones sensibles a la latencia es recomendable desactivar los modos de ahorro de energía del Wi-Fi.

### 6.3.2. ESP32-*FreeRTOS*

El ESP32 ejecuta *FreeRTOS* como sistema operativo de tiempo real integrado en ESP-IDF, proporcionando un planificador por prioridades, colas, semáforos, temporizadores de *software* y asignación a un núcleo mediante funciones como `xTaskCreatePinnedToCore`. ESP-IDF añade utilidades que permiten estructurar aplicaciones en tareas independientes y comunicadas por colas para evitar condiciones de carrera [19]. Para un correcto funcionamiento se debe asignar mayor prioridad a tareas temporizadas de control y menor a tareas de mantenimiento o registro (evitando prioridades excesivas), fijar las tareas de control a un núcleo y dejar la pila de red (*lwIP*) en el otro para reducir fluctuaciones, usar `vTaskDelay()` o similares para ceder CPU y evitar bucles y en la comunicación entre interrupciones y tareas utilizar las variantes como `xQueueSendFromISR` o `xTaskNotifyFromISR`.

## 6.4. Baterías de Li-ion

El rendimiento de los motores paso a paso, del controlador DRV8825 y del microcontrolador ESP32 depende de una fuente de alimentación capaz de suministrar corrientes transitorias elevadas y de mantener niveles de voltaje estables bajo variaciones de carga. En este sentido, la selección de la tecnología de baterías constituye un componente crítico del diseño energético de la plataforma. Como señala Frenzel, los sistemas electrónicos requieren fuentes de corriente directa que garanticen estabilidad y continuidad operativa, y las baterías cumplen esta función al almacenar y liberar energía mediante procesos electroquímicos internos [20].

Dentro de las opciones disponibles, las baterías de ion-litio ofrecen características particularmente adecuadas para sistemas móviles. Según [21], estas presentan un voltaje nominal de 3.7V por celda, lo que permite la creación de paquetes compactos con menos celdas que otras tecnologías. Además, son capaces de suministrar corrientes sostenidas y picos de potencia elevados, y exhiben una baja tasa de autodescarga que contribuye a una vida útil prolongada. Sin embargo, su alta densidad energética implica riesgos si no se emplean circuitos de protección adecuados. Pues las condiciones de sobrecarga, sobrecorriente o elevación térmica pueden conducir a fallas severas, incluida inflamación o explosión.

## 6.5. Robots móviles omnidireccionales de cuatro ruedas

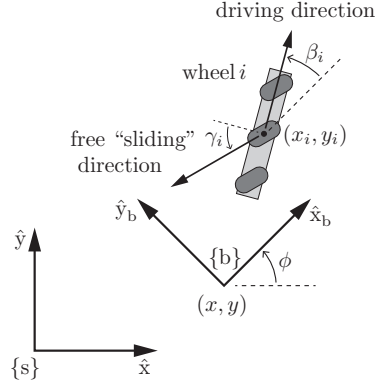
Una vez descritos los elementos que permiten accionar, sensar y alimentar el sistema, es necesario abordar la configuración mecánica que determina su movilidad en el plano. En este proyecto, dicha movilidad se logra mediante una disposición de cuatro ruedas omnidireccionales. Las ruedas omnidireccionales, a diferencia de las convencionales, permiten al robot moverse en cualquier dirección sin necesidad de girar (como se puede visualizar en la Figura 6 ). Esto se debe a que proveen cierta resistencia (muy baja) en otras direcciones. Por lo mismo, suelen usarse en aplicaciones donde el espacio es limitado o se requiere un control muy específico del robot [22].

### 6.5.1. Modelado cinemático

En [23] se detalla a profundidad el análisis cinemático de las plataformas móviles con ruedas omnidireccionales (tanto para 3 como 4 ruedas). Este modelado busca obtener la relación entre el vector de velocidad  $\dot{q} = (\dot{\phi}, \dot{x}, \dot{y})$  con las velocidades de sus ruedas.

En la Figura 6 se puede observar como se compone la velocidad final de cada rueda con la velocidad de dirección (dada por la circunferencia central) y la velocidad de deslizamiento (dada por los rodillos). Donde el ángulo  $\gamma$  representa la dirección hacia donde el deslizamiento ocurre, que está dada por el ángulo de los rodillos con respecto a la circunferencia central, que para cuatro ruedas tiene un valor de  $45^\circ$ .

**Figura 6.** Representación de marcos de referencia y ruedas mecanum



Nota. Se muestran el marco fijo  $\{s\}$ , del chasis  $\{b\}$ , y la disposición de las llantas mecanum con sus respectivas direcciones de avance  $\beta_i$  y deslizamiento  $\gamma_i$ . Imagen obtenida de [23].

Según [23] la función de control de una plataforma móvil omnidireccional sobre un marco de referencia "fijo" se consigue al encontrar la relación entre la velocidad de cada rueda  $\{u_i\}$  y  $\dot{q}$  de la siguiente manera:

$$\mathbf{u}_i = h_i(\phi) \dot{\mathbf{q}} = \begin{bmatrix} \frac{1}{r_i} & \frac{\tan \gamma_i}{r_i} \end{bmatrix} \begin{bmatrix} \cos \beta_i & \sin \beta_i \\ -\sin \beta_i & \cos \beta_i \end{bmatrix} \begin{bmatrix} -y_i & 1 & 0 \\ x_i & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \end{bmatrix}. \quad (4)$$

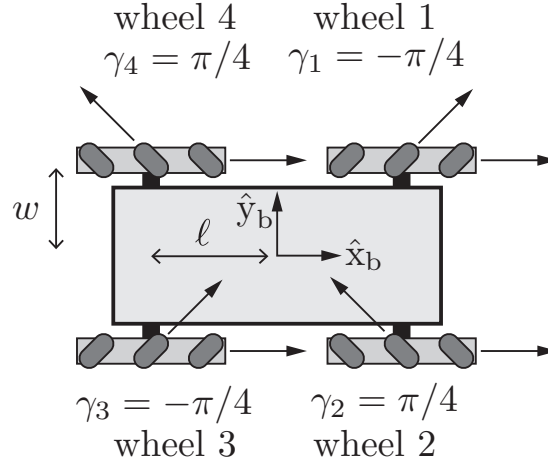
Este proceso, tiene como resultado la función descrita en (5).

$$h_i(\phi) = \frac{1}{r_i \cos \gamma_i} \begin{bmatrix} x_i \sin(\beta_i + \gamma_i) - y_i \cos(\beta_i + \gamma_i) \\ \cos(\beta_i + \gamma_i + \phi) \\ \sin(\beta_i + \gamma_i + \phi) \end{bmatrix}^T. \quad (5)$$

Esto se simplifica si se usa el modelo mostrado en la Figura 7, pues al ser simétrico y no tener velocidades de deslizamiento repetidas, el resultado de evaluar la matriz tiene un rango de 3, lo que muestra la capacidad de control en cada uno de los grados de libertad ( $\dot{q} = (\dot{\phi}, \dot{x}, \dot{y})$ ) en el plano  $H(0)$ . Dicho resultado se muestra en (6), que es la función final que se usa para el control de la plataforma móvil.

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = H(0) \mathbf{V}_b = \frac{1}{r} \begin{bmatrix} -\ell - w & 1 & -1 \\ \ell + w & 1 & 1 \\ \ell + w & 1 & -1 \\ -\ell - w & 1 & 1 \end{bmatrix} \begin{bmatrix} \omega_{bz} \\ v_{bx} \\ v_{by} \end{bmatrix} \quad (6)$$

**Figura 7.** Modelo cinemático de un robot omnidireccional de 4 ruedas



Nota. Este modelo muestra las relaciones geométricas y cinemáticas de un robot simétrico con 4 ruedas mecanum. Imagen obtenida de [23].

### 6.5.2. Control de movimiento

Con el objetivo de minimizar el error y habilitar tareas como el seguimiento de trayectorias, en robótica se emplea habitualmente un control en lazo cerrado. Una implementación común para las plataformas omnidireccionales combina una acción de realimentación anticipada (*FeedForward*) con un controlador PI que corrige los errores de seguimiento. Un esquema de este tipo para la velocidad del cuerpo se expresa mediante la siguiente ecuación [23]:

$$\dot{q}(t) = \dot{q}_d(t) + K_p(q_d(t) - q(t)) + K_i \int_0^t (q_d(\tau) - q(\tau)) d\tau, \quad (7)$$

donde  $K_p$  y  $K_i$  son matrices diagonales de ganancia,  $q_d(t)$  es la trayectoria de referencia y  $q(t)$  es la configuración actual estimada a partir de los sensores. El vector de velocidad  $\dot{q}(t)$  obtenido en (7) puede transformarse en las velocidades angulares de las ruedas  $\mathbf{u}(t)$  empleando la relación cinemática (6).

### 6.5.3. Planificación

Los métodos de planificación se basan en un mapa conocido del entorno; son relativamente sencillos de implementar porque asumen que la localización y la disposición de los obstáculos son conocidas con antelación, de modo que basta generar una trayectoria desde el origen hasta el objetivo. Entre las representaciones más habituales están los grafos (por ejemplo, visibilidad o PRM) y las cuadrículas de ocupación, que indican las celdas libres y ocupadas y permiten aplicar algoritmos de búsqueda en grafos como  $A^*$ ,  $D^*$  o Dijkstra para obtener rutas seguras [24].

---

## Estudio comparativo para la selección de componentes

---

Como se explicó anteriormente, el método MCDA permite evaluar múltiples criterios para tomar decisiones fundamentadas. En este caso, el objetivo era escoger el motor y el controlador físico del motor más adecuado para la plataforma móvil con ruedas omnidireccionales, debido a que esta elección afectó directamente al proceso de diseño e integración de la plataforma móvil. Este proceso inició con la investigación de proveedores locales y la disponibilidad de componentes electrónicos, como requisito para el cumplimiento de uno de los objetivos del proyecto.

### 7.1. Criterios de selección del modelo de los motores

Las plataformas móviles omnidireccionales requieren actuadores que ofrezcan precisión tanto en la velocidad como en la posición, pues el movimiento coordinado de las ruedas depende directamente de la exactitud en la ejecución de las referencias. A esto se suma la necesidad de mantener un diseño lo más compacto posible, sin sacrificar el torque necesario para desplazar la plataforma y soportar cargas adicionales. Además, uno de los objetivos del proyecto es que la solución pueda reproducirse localmente, por lo que los componentes deben ser accesibles en Guatemala y contar con disponibilidad continua.

Con base en la investigación realizada sobre proveedores de electrónica locales y en los requisitos funcionales del proyecto, se definieron los criterios agrupados por nivel de importancia que guiaron la selección final del modelo de motor:

- **Máxima:** precisión en el control de posición y velocidad.
- **Alta:** velocidad de giro adecuada, para que la plataforma pueda desplazarse con fluidez y corregir rápidamente errores.

- **Media-Alta:** torque suficiente para la plataforma y cargas adicionales.
- **Media:** tamaño de motor compacto para facilitar el diseño.
- **Baja:** precio accesible para no aumentar desmedidamente el precio final de la plataforma móvil, complejidad de control manejable y consumo energético eficiente.

### 7.1.1. Ponderación de los criterios

Para la ponderación de los criterios, se utilizó el método *AHP*. Comparando uno por uno los criterios junto a su nivel de importancia para el proyecto. Cabe aclarar que la matriz de comparación se construyó llenando únicamente la parte superior y la parte inferior se completó con recíprocos correspondientes para evitar valores incoherentes.

**Cuadro 1.** Matriz de comparación de los motores

Criterios	Complejidad	Consumo	Precio	Precisión	Tamaño	Torque	Velocidad
Complejidad (control)	1	1/5	3	1/7	1/3	1/3	1/5
Consumo	5	1	7	1/3	3	3	1
Precio	1/3	1/7	1	1/9	1/5	1/5	1/7
Precisión	7	3	9	1	5	5	3
Tamaño	3	1/3	5	1/5	1	1	1/3
Torque	3	1/3	5	1/5	1	1	1/3
Velocidad	5	1	7	1/3	3	3	1

Nota. El cuadro muestra la matriz de comparación *AHP* de los criterios de selección de los motores. Elaboración propia.

Posteriormente, con ayuda de MATLAB, se calculó el vector de ponderaciones utilizando la ecuación (2). Adicionalmente, también se comprobó la consistencia de la matriz de ponderaciones mediante el cálculo del índice de consistencia (CI), aplicando la ecuación (3). El valor de dicha variable fue de 0.047011, valor que cumplía con el criterio de  $CI < 0.1$ . Como resultado, se obtuvo el conjunto de ponderaciones mostrado en el Cuadro 12 en el apartado de Anexos.

## 7.2. Alternativas para la selección de los motores

En el caso de las alternativas, se escogieron motores que estuvieran dentro de un rango aceptado de los criterios anteriormente mencionados. Por ejemplo, un motor con gran torque y un tamaño diminuto parecería adecuado, pero si su velocidad máxima es demasiado baja, no tendría sentido tenerlo en consideración pues no cumpliría con las especificaciones requeridas del motor. La lista de los motores para el estudio comparativo fue la siguiente:

- **DC:** Motor 555, Motor 775.

- **Reductor:** 25GA370, 37D, N20, TT.
- **Brushless:** Motor A2212.
- **Stepper:** Nema 17 (longitud de cuerpo 23 mm), Nema 17 (longitud de cuerpo 40 mm).

### 7.2.1. Calificación de los motores

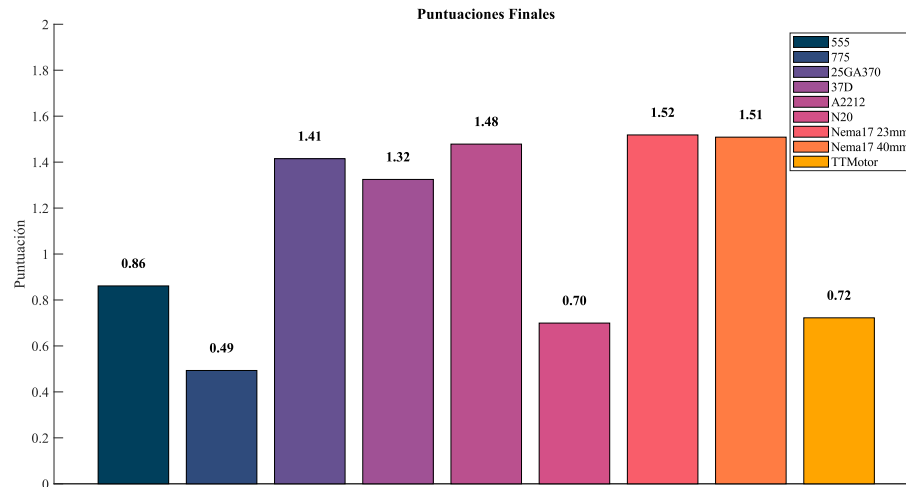
Con base en las especificaciones de las alternativas del Cuadro 10 y las funciones provistas en 13.1.1, los pasos para la calificación de cada modelo de motor fueron:

1. Para los criterios cuantitativos de: consumo, precio, tamaño y torque, se utilizó una regresión lineal normalizada (14).
2. Para la velocidad, la transformación fue logarítmica (15) por la amplia dispersión entre los valores para cada alternativa.
3. Para los criterios cualitativos como precisión y control, se estableció una escala ordinal. En el caso de la precisión se utilizó la escala ordinal positiva presentada en 13.1.1. Para el criterio de control se utilizó la misma escala pero en orden inverso pues se desea un control más sencillo.

### 7.3. Aplicación del método y selección final de los motores

Usando la ecuación (1) para calcular los valores finales de la ponderación junto a las calificaciones y pesos obtenidos en el Cuadro 12, los resultados del estudio comparativo se pueden visualizar en la Figura 8. Estos resultados demostraron que las 2 mejores opciones eran: el Nema 17 (23mm) con nota de 1.52 y el Nema 17 (40mm) con nota de 1.51.

**Figura 8.** Resultado del MAUT/MAVT para la selección de los motores



Nota. Este gráfico muestra el puntaje final de cada motor candidato evaluado bajo el método de análisis multiatributo (MAUT/MAVT). Elaboración propia.

A pesar de la poca diferencia entre las notas de ambas versiones del Nema 17, la selección final fue la versión de 23 mm. No sólo por obtener la puntuación más alta, sino porque es coherente con lo esperado para el proyecto. Esto debido a que al ser pequeño permite una integración fácil dentro de la plataforma móvil. Además, permite la precisión y velocidad adecuadas para el control de las llantas omnidireccionales con un torque que no debería tener problemas para mover la plataforma móvil.

## 7.4. Criterios de selección del controlador físico del motor

Para la elección del controlador, se repitieron los pasos seguidos previamente en la elección del motor. En esta ocasión, el objetivo era identificar el controlador más adecuado para gestionar el comportamiento del motor seleccionado. El controlador del motor debía ser compatible con el stepper Nema 17 (soportando su voltaje y corriente), operable con los niveles lógicos del ESP32 (3.3 V) y permitir control fino de posición y velocidad (como los micropasos). Además debía ser económico para facilitar su réplica, soportar operación continua con protección eléctrica y disipación térmica adecuadas, ser compacto para una fácil integración en la plataforma y minimizar las conexiones sin perder funcionalidad.

En este caso los criterios agrupados por nivel de importancia fueron:

- **Máxima:** control de posición preciso y velocidad no limitada por el controlador.
- **Alta:** tamaño compacto.
- **Media:** disipación de calor para evitar sobrecalentamiento y protección para evitar cortos.
- **Baja:** precio económico.

### 7.4.1. Ponderación de los criterios

La matriz del Cuadro 2 tuvo un índice de consistencia con valor de 0.0082699, lo cual indica que la matriz cumple con el criterio de  $CI < 0.1$ . El vector de ponderaciones se muestra en el Cuadro 13.

**Cuadro 2.** Matriz de comparación de los controladores físicos

Criterio	Disipación	Posición	Precio	Protección	Tamaño	Velocidad
Disipación	1	1/3	2	1	1/2	1/3
Posición	3	1	4	3	2	1
Precio	1/2	1/4	1	1/2	1/3	1/4
Protección	1	1/3	2	1	1/2	1/3
Tamaño	2	1/2	3	2	1	1/2
Velocidad	3	1	4	3	2	1

Nota. El cuadro muestra la matriz de comparación AHP de los criterios de selección de los controladores. Elaboración propia.

## 7.5. Alternativas del controlador físico del motor

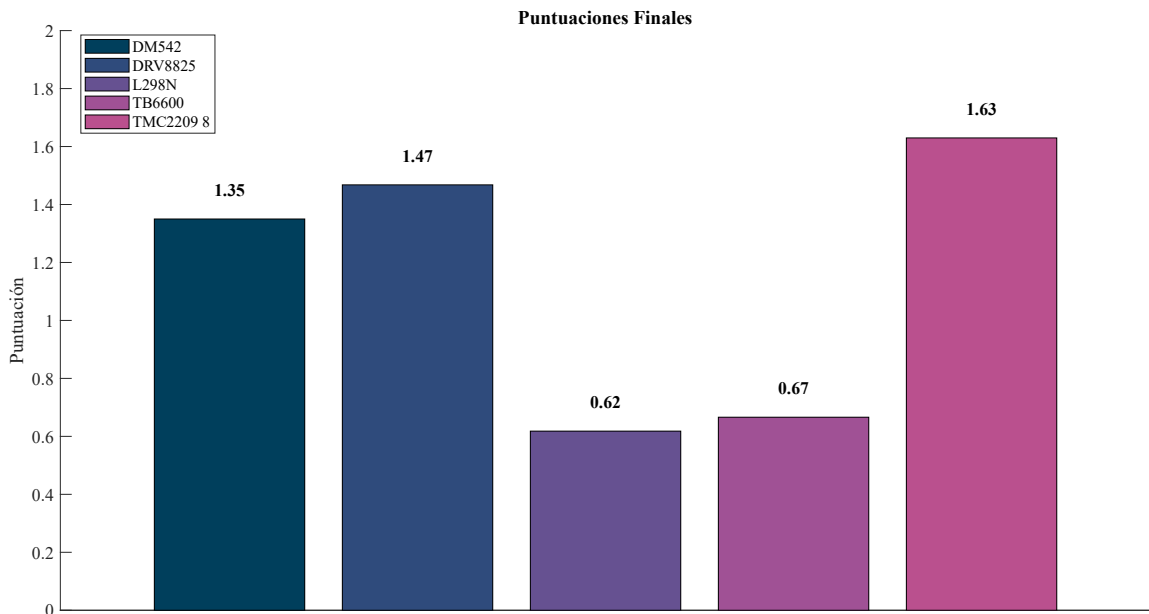
En el caso de las alternativas, se tomaron en cuenta los controladores existentes en el mercado local que fueran compatibles con el ESP32 y el Nema 17. Por lo mismo, las alternativas fueron: DM542, DRV8825, L298N, TB6600 y el TMC2209 (o la versión del TMC2208).

### 7.5.1. Calificación de las alternativas del controlador físico del motor

Basado en las especificaciones del Cuadro 11, se repitieron los mismos pasos que se realizaron para la calificación de los motores. Para la velocidad se utilizó una regresión lineal positiva (14); para el precio y tamaño una transformación logarítmica negativa (16) y para el resto de criterios una escala ordinal positiva.

## 7.6. Aplicación del método y selección final del controlador físico del motor

**Figura 9.** Resultado final del MAUT/MAVT para el control de los motores



Nota. El análisis muestra los resultados obtenidos al comparar distintas opciones de sistemas de control para los motores paso a paso. Elaboración propia.

El análisis del Cuadro 13, visualizado en la Figura 9, mostró que las dos alternativas mejor clasificadas fueron el TMC2209 (puntaje 1.63) y el DRV8825 (puntaje 1.47). Ambos controladores se consideraron viables para su uso en el prototipo. Aunque en este proyecto se optó finalmente por el DRV8825 como el principal controlador debido a su mayor disponibilidad local y a la sencillez de integración en el prototipo, el TMC2209 permanecerá como alternativa válida siempre y cuando el cableado y configuración sea el correcto.

El estudio comparativo presentó una base cuantitativa y cualitativa sólida para la selección de los elementos de locomoción del prototipo, cuyas características influyeron directamente tanto al diseño mecánico, como al diseño electrónico. En el siguiente capítulo se desarrolló en detalle el diseño y la manufactura de la plataforma móvil, incluyendo los criterios de dimensionamiento estructural, la selección de materiales y procesos de fabricación, así como el diseño de la electrónica de potencia y control para el ESP32 y los motores seleccionados.

---

## Diseño y manufactura electromecánica de la plataforma omnidireccional

---

En el diseño electromecánico se tuvo en consideración el consumo eléctrico de los motores seleccionados en el capítulo anterior, la disponibilidad de componentes electrónicos localmente y los tamaños que estos componentes ocuparían dentro de la plataforma móvil. Además de reiterar dicho proceso múltiples veces, para permitir el correcto ensamblaje de la plataforma.

### 8.1. Diseño electrónico

Para el diseño electrónico de los circuitos y placas PCB que controlan y alimentan la plataforma móvil se utilizó la herramienta de *Altium Designer*.

#### 8.1.1. Circuito de alimentación

El circuito de alimentación se diseñó tomando como base dos voltajes principales: un voltaje de 5 V para el ESP32 y los componentes que este alimenta a 3.3 V y un voltaje de 10 V destinado a los cuatro motores NEMA 17. El presupuesto de potencia se encuentra en el Cuadro 3.

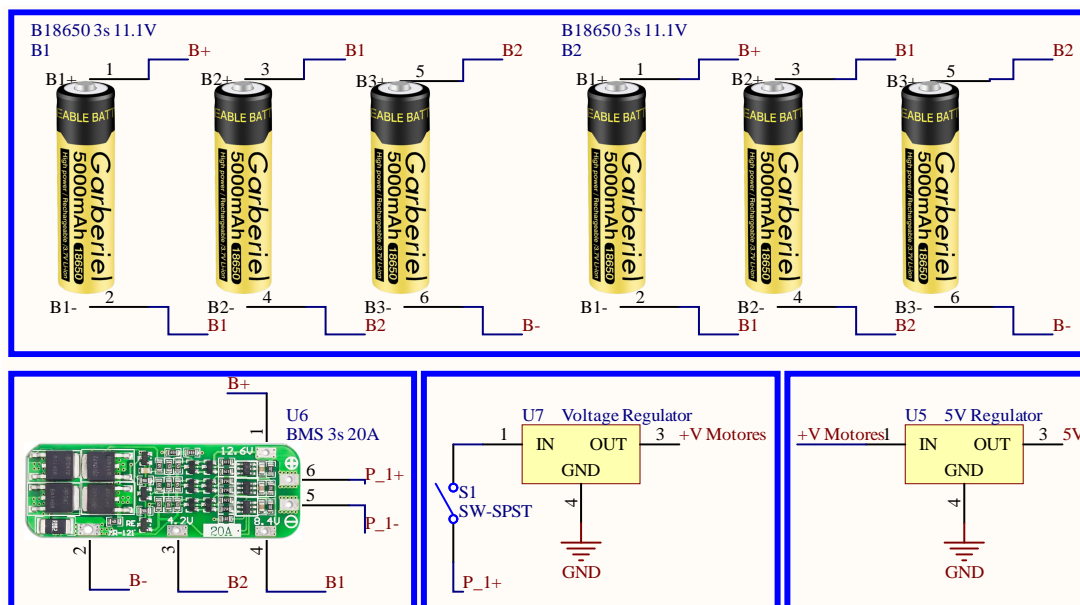
Debido a esta demanda energética, se seleccionó un banco de seis baterías Li-Ion 18650. Para obtener un voltaje nominal entre 11.1 V y 12.6 V, así como para aumentar la capacidad y la corriente disponible, se empleó una configuración **3 en serie y 2 en paralelo** (3S2P). Esta configuración permite alimentar adecuadamente los motores y mantener una descarga estable durante periodos prolongados de operación. Pues cada batería cuenta con una capacidad de operación de 2550mAh (un total de 5,100 mA h). El diagrama completo del circuito de alimentación se muestra en la Figura 10.

**Cuadro 3.** Presupuesto de potencia del circuito de alimentación

Componente	Voltaje (V)	Corriente Máx (A)	Potencia (W)
<b>ESP32</b>	5	1.1	5.5
<b>Motores NEMA 17</b>	10	8.0	80.0
<b>Sensores AS5600</b>	3.3	0.026	0.0858
<b>Drivers DRV8825</b>	3.3	0.004	0.0132
<b>Total aproximado</b>	—	—	<b>85.6</b>

Nota. Se asume el consumo máximo de 1.0 A por fase de cada motor.

**Figura 10.** Esquemático del circuito de alimentación



Nota. Elaboración propia.

Dicho circuito de alimentación se conecta a 2 módulos reguladores de voltaje que son los encargados de alimentar la placa de control de la Figura 26, permitiendo tener un voltaje de 5 V para la alimentación del ESP32 y un voltaje de 10 V para los controladores y motores paso a paso. Este último valor se escogió debido a que la recomendación es que el voltaje de entrada de un regulador de voltaje debe superar por al menos 1 V al voltaje deseado en la salida.

Por otra parte, las baterías y el circuito de alimentación debían ser capaces de suministrar un amperaje teórico de 9.1 A (8 A de las 2 fases de los 4 motores y 1.1 A para la alimentación del ESP32). Por lo que las pistas para la placa PCB de la alimentación debían de tener dicha capacidad para no sobrecalentarse y poder alimentar correctamente a todos los componentes. El tamaño de las pistas fue diseñado cumpliendo con el estándar IPC-2221 [25] y las especificaciones indicadas en el documento de especificaciones LPKF para el *MakerLab* [26]. El diseño de dicha placa PCB se puede ver en la Figura 25.

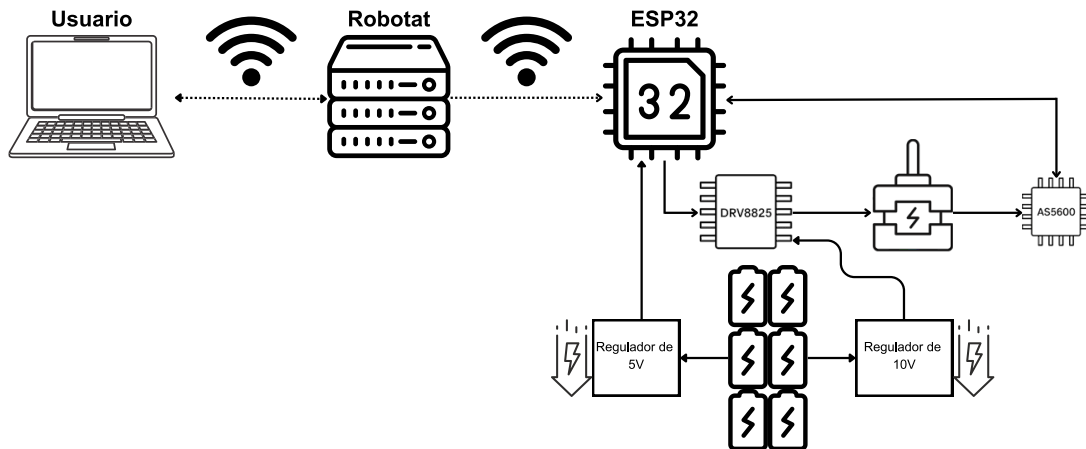
Por último, para la carga y descarga de las baterías se tomó de referencia la hoja de datos de las baterías ICR18650 [27]. En este se indica un valor de carga estándar de 4.20V con 1.3A (por 3 horas) y una carga rápida de 4.20V con 2.6A (por 2.5 horas) por celda. Por lo mismo, el paquete usado para el circuito de alimentación debería cargarse a 12.6V con 2.6A o 5.2A. Sin embargo, se recomienda usar la carga estándar (2.6A) pues desgasta menos las baterías y evita sobrecalentamientos.

Para la descarga real de las baterías se midió un valor de corriente experimental de 1.492 A para todo el circuito. Por lo que se cumplió con lo especificado en la hoja de datos [27], que indica una descarga máxima continua de 5.2 A por celda, equivalente a 10.4 A para el paquete 3S2P. Esto garantizó un margen de seguridad amplio frente al consumo teórico de 9.1 A y experimental de 1.492 A. Además, este nivel de demanda, en conjunto con la capacidad total de 5,100 mA h, proporcionó una autonomía estimada de 3.4 horas de operación continua.

### 8.1.2. Circuito de control

El circuito de control se basó en el diagrama de bloques del proyecto presentado en la Figura 11.

**Figura 11.** Diagrama de bloques del funcionamiento de la plataforma móvil



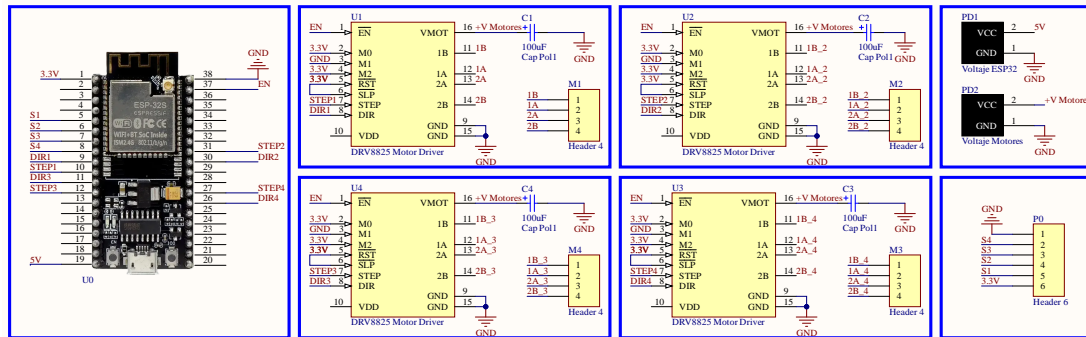
Nota. El ESP32 controla 4 DRV8825 con sus respectivos motores y retroalimentación del AS5600. Elaboración propia.

El objetivo era que el usuario final pudiera controlar la plataforma móvil a través de su propio computador. Esto mediante Wi-Fi usando el servidor del Robotat para enviar los comandos de control al ESP32 de la plataforma móvil. El circuito de control se centró en esta última parte, pues debía ser capaz de controlar los 4 motores conectados a los DRV8825 y también recibir retroalimentación del AS5600 para controlar la velocidad de los motores.

Los pines del ESP32 que se utilizaron para el circuito de control fueron los siguientes:

- **Sensores AS5600:** GPIO34, GPIO35, GPIO32, GPIO33.
- **DIR (DRV8825):** GPIO25, GPIO18, GPIO27, GPIO4.
- **STEP (DRV8825):** GPIO26, GPIO19, GPIO14, GPIO5.
- **EN (DRV8825):** GPIO21 para todos los motores
- LED de conexión Wi-Fi: GPIO2 (LED integrada en el ESP32).

Figura 12. Esquemático de circuito de control



Nota. Elaboración propia.

Cabe destacar que, como se muestra en la Figura 12, los pines de micropasos se conectan directamente a 3 V, tierra y 3 V, por lo que el motor siempre tendrá 1/32 de paso. Esta configuración es igual para cada uno de los controladores DRV8825, es decir, todos los motores tienen los mismos micropasos. De la misma manera, todos los motores son habilitados y deshabilitados por el mismo pin del ESP32. Esta configuración se realizó pensando en la compatibilidad de la placa PCB tanto con el DRV8825 como el TMC2209.

Además, los 4 sensores AS5600 tienen sus pines de *OUT* en modo analógico conectados a pines del ESP32 que forman parte del mismo ADC. Esto se debe a que los canales pertenecientes al ADC0 están relacionados a la comunicación Wi-Fi, por lo que, para evitar interferencias, se utilizó solo el ADC1.

Para la placa PCB mostrada en la Figura 26, se tomaron en cuenta los valores de corriente y voltaje de alimentación para el tamaño de las pistas. Además, se optó por el uso polígonos para disipar de mejor forma el calor de la corriente del voltaje de alimentación de los motores.

## 8.2. Diseño mecánico

El diseño de las piezas de la carcasa de la plataforma móvil que almacenan los circuitos de alimentación y control de la plataforma móvil, al igual que las llantas omnidireccionales, se realizó usando la herramienta de *Autodesk Fusion 360*.

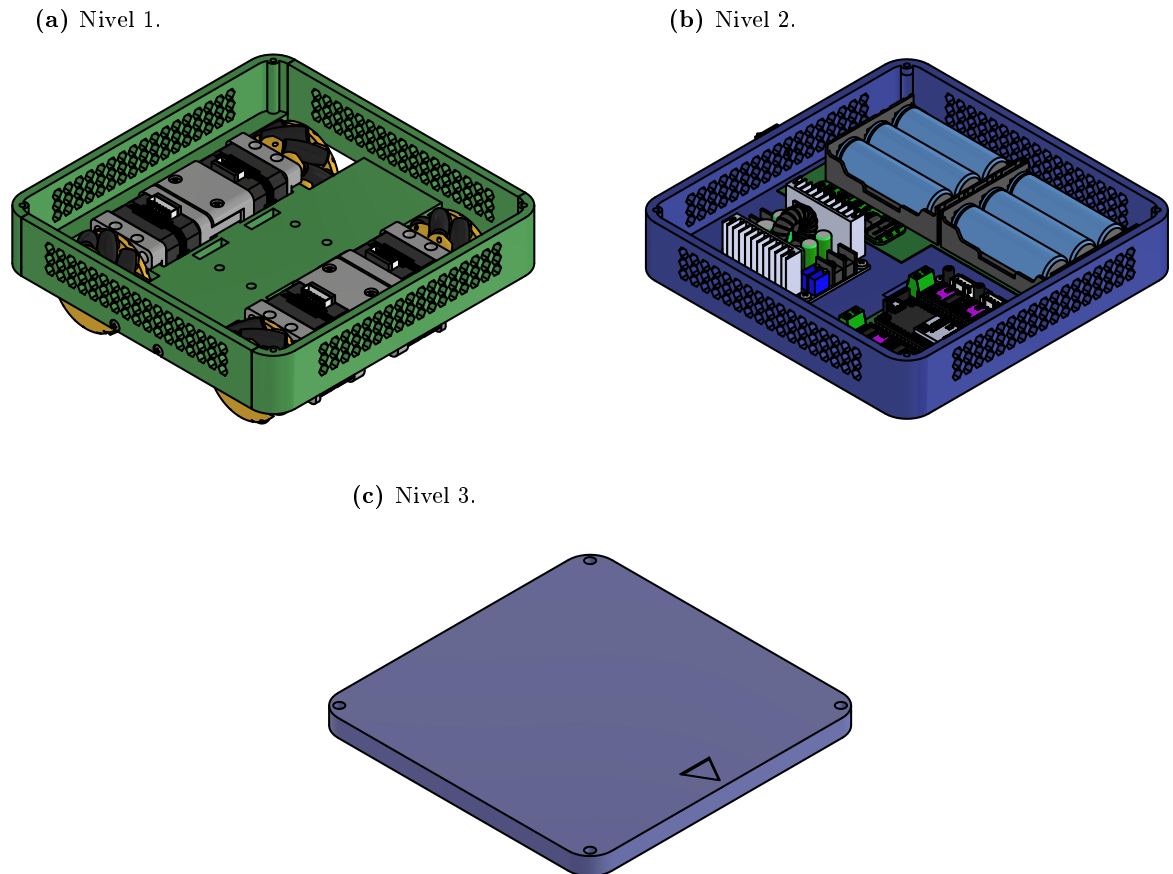
### 8.2.1. Chasis de la plataforma móvil

El chasis se diseñó considerando usos futuros y la integración de los circuitos de control y potencia descritos en la sección 8.1. Incluye tanto componentes comerciales (tornillos, motores, reductores de voltaje, etc.) como piezas diseñadas a medida, las cuales se fabricaron para ser compatibles con los componentes comerciales.

Para optimizar el tamaño y permitir expansiones, se adoptó un diseño modular por niveles: los dos inferiores, de 5 mm de espesor, y la cubierta superior, de 10 mm, para soportar mayores cargas. La disposición se muestra en la Figura 13. Los orificios hexagonales en los niveles facilitan la ventilación de PCB y motores, además de alojar tuercas M3 para los marcadores detectables por las cámaras del Robotat [2], mejorando el posicionamiento y permitiendo la unión futura de plataformas.

Las dimensiones finales fueron  $190 \times 190$  mm de base y 82 mm de altura (112 mm con ruedas). Los planos y medidas detalladas pueden encontrarse en los Anexos.

**Figura 13.** Niveles de la plataforma móvil en *Fusion 360*



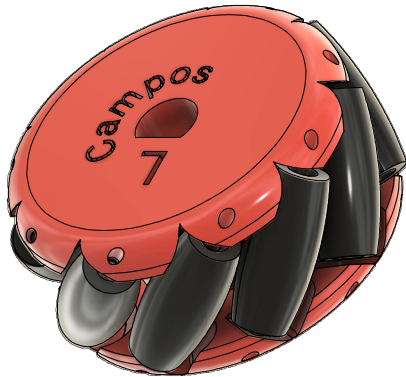
Nota. **Nivel 1:** soportes de motores con AS5600 y llantas omnidireccionales. **Nivel 2:** circuitos de alimentación y de control. **Nivel 3:** plataforma para la colocación de otros agentes. Elaboración propia.

### 8.2.2. Llantas omnidireccionales

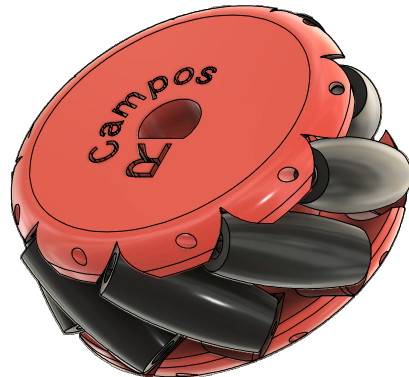
A pesar de los diversos intentos por fabricar localmente llantas *mecanum* a  $45^\circ$  (Figura 14) impresas en 3D, no se alcanzó el comportamiento omnidireccional esperado. Una explicación plausible es que la fricción entre cada rodillo y su eje impidió el libre giro y la transferencia adecuada de fuerzas requerida para el deslizamiento controlado. Esto podría mitigarse agregando rodamientos o un acoplamiento con bajo rozamiento. Las deficiencias de este diseño generaron un comportamiento no homogéneo entre ruedas, que dificultó la síntesis de comandos para lograr movimientos laterales precisos. No obstante, el diseño y fabricación digital facilitó la adaptación rápida del chasis a las ruedas comerciales disponibles, lo que permitió continuar con la validación de la plataforma sin retrasos significativos.

**Figura 14.** Diseño de las ruedas omnidireccionales

(a) Rueda izquierda.



(b) Rueda derecha.

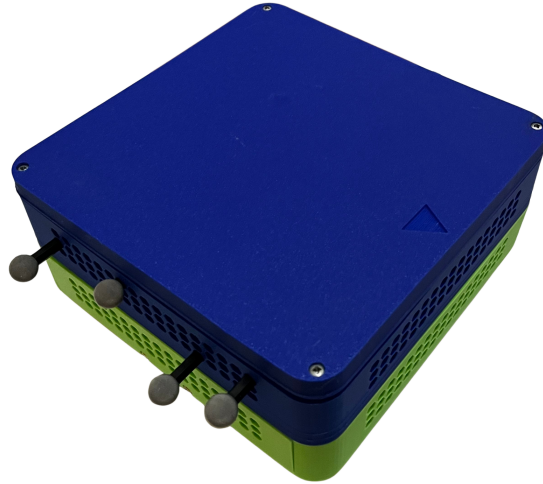


Nota. La diferencia entre la rueda izquierda y la rueda derecha es la dirección del ángulo de los rodillos. Elaboración propia.

### 8.3. Manufactura e integración de componentes locales en la plataforma móvil

Uno de los objetivos específicos en este proyecto fue usar componentes accesibles localmente y manufactura digital ágil y de bajo costo para el desarrollo de la plataforma móvil. Esta restricción condicionó varios aspectos del diseño, ya que limitó la disponibilidad de algunas piezas y la selección de componentes comerciales. No obstante, esa misma restricción permitió ser creativo y priorizar soluciones replicables en el contexto local. El proceso de diseño y manufactura culminó con el ensamble final de toda la plataforma móvil, que se puede visualizar en la Figura 15.

**Figura 15.** Ensamble completo



Nota. La figura muestra el ensamble final de la plataforma móvil. Elaboración propia.

### 8.3.1. Proceso de manufactura

La etapa de manufactura y ensamblaje se estructuró en tres actividades principales: (1) impresión 3D de las piezas estructurales, (2) fabricación y soldadura de las placas de circuito impreso y (3) ensamblado final de la plataforma móvil (véase la Figura 15). Para la primera etapa, se imprimieron y ensamblaron los tres niveles del chasis en PLA por separado como muestra la Figura 16. El primer nivel, contiene los motores y los sensores AS5600 así como el cableado de dichos componentes. El segundo nivel de la plataforma móvil fue diseñado para que las placas PCB estuvieran dentro del mismo. Mientras que el último nivel sella y permite colocar a otro agente robótico en la plataforma móvil.

Respecto a las PCBs se diseñaron y fabricaron dos placas: la de alimentación y la de control, como se mostró en la etapa de diseño. Para el proceso de validación de fabricación de manera ágil y local, se realizaron pruebas con las placas de la Figura 17 producidas en el laboratorio *MakerLab* de la UVG. Las pruebas con estas placas confirmaron el funcionamiento correcto de los circuitos eléctricos de control y alimentación de la plataforma móvil.

Siguiendo el objetivo de realizar toda la plataforma móvil usando componentes que se encuentran a nivel local, en el Cuadro 4 se encuentra el *Bill of materials* (listado de materiales) o BOM con todos los componentes que la plataforma móvil requiere para su construcción. Mientras que, en el Cuadro 5, se encuentran los enlaces de los proveedores locales de todos los componentes.

**Figura 16.** Impresión 3D del chasis

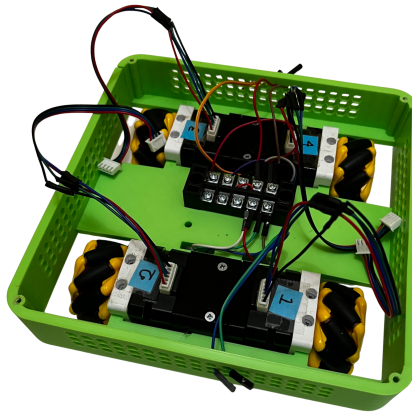
(a) Manufactura nivel 3.



(b) Manufactura nivel 2.



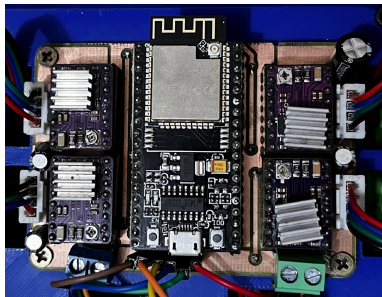
(c) Manufactura nivel 1.



Nota. La figura muestra los niveles del chasis impresos en PLA. Elaboración propia.

**Figura 17.** Manufactura placas PCB

(a) Placa PCB de control.



(b) Placa PCB de alimentación.



Nota. Placas realizada en el *MakerLab* de la UVG. Elaboración propia.

### 8.3.2. Listado de materiales

**Cuadro 4.** *Bill of materials:* precios

Material	Precio Unitario	Cantidad	Precio Total
Nema 17 23mm	Q139.00	4	Q556.00
Ruedas Mecanum (par L-R)	Q99.00	2	Q198.00
Material PLA (Por Kg)	Q175.00	0.50368	Q88.14
Regulador de Voltaje (300W)	Q129.00	1	Q129.00
Regulador de Voltaje 5V	Q27.00	1	Q27.00
AS5600	Q70.00	4	Q280.00
Cables JST XH Hembra (6 a 4 pines)	Q24.00	4	Q96.00
Cable Calibre 14 (Por metro)	Q4.50	2	Q9.00
Tira de Bornera (6Pines)	Q11.00	1	Q11.00
Jumpers Hembra-Macho 20cm (x10)	Q8.00	1	Q8.00
Jack Banana Hembra	Q4.50	2	Q9.00
Placa Alimentación (referencia)	Q35.00	1	Q35.00
BMS (3s 20a)	Q45.00	1	Q45.00
Baterías 18650	Q59.00	6	Q354.00
Sujetador bat. PCB 3s	Q19.00	2	Q38.00
Placa Control (referencia)	Q10.00	1	Q10.00
ESP32	Q125.00	1	Q125.00
DRV8825 (Driver Paso a Paso)	Q38.00	4	Q152.00
Header Hembra (40 pines)	Q4.00	1	Q4.00
Header Hembra (8 pines)	Q3.00	9	Q27.00
Conector Macho 4 pines	Q1.00	4	Q4.00
Switch	Q6.00	1	Q6.00
Bornera 15A	Q2.00	2	Q4.00
Tornillos M3x8 Allen	Q3.53	4	Q14.12
Tornillos M3x8 Planos Phillips	Q1.12	12	Q13.44
Tornillos M3x8 Redonda Phillips	Q1.33	40	Q53.20
Tornillos M3x12 Planos Phillips	Q1.64	12	Q19.68
Tornillos M3x12 Redonda Phillips	Q1.64	12	Q19.68
Tornillos M3x16 Redonda Phillips	Q1.95	4	Q7.80
Tornillos M3x25 Planos Phillips	Q2.65	8	Q21.20
Tuercas M3	Q1.48	44	Q65.12
<b>Total</b>			<b>Q2,429.38</b>

Nota. El cuadro muestra los componentes necesarios para la elaboración de la plataforma móvil junto al proveedor a nivel local. Los precios son los consultados a la fecha del 27 de agosto del 2025. Elaboración propia.

**Cuadro 5.** *Bill of materials:* distribuidores

<b>Componente</b>	<b>Distribuidor / enlace</b>
Nema 17 23mm	LaElectrónica
Ruedas Mecanum (par L-R)	Oxdea
Material PLA (Por Kg)	Creativo3D
Regulador de Voltaje (300W)	LaElectrónica
Regulador de Voltaje 5V	LaElectrónica
AS5600	Oxdea
Cables JST XH Hembra (6 a 4 pines)	LaElectrónica
Cable Calibre 14 (Por metro)	LaElectrónica
Tira de Bornera (6Pines)	LaElectrónica
Jumpers Hembra-Macho 20cm (x10)	LaElectrónica
Jack Banana Hembra	Tettsa
Placa Alimentación (15x20cm de referencia)	MakerLab/LaElectrónica
BMS (3s 20a)	Oxdea
Baterías 18650	LaElectrónica
Sujetador bat. PCB 3s	LaElectrónica
Placa Control (7x10cm de referencia)	MakerLab/LaElectrónica
ESP32	LaElectrónica
DRV8825 (Driver Paso a Paso)	LaElectrónica
Header Hembra (40 pines)	Tettsa
Header Hembra (8 pines)	Tettsa
Conector Macho 4 pines	LaElectrónica
Switch	Tettsa
Bornera 15A	LaElectrónica
Tornillos y Tuercas (varios)	La Casa del Tornillo

Nota. Dentro del texto del distribuidor se encuentran los enlaces de los componentes tomados el 27 de agosto del 2025. Elaboración propia.

El proceso de diseño y manufactura presentado en este capítulo sentó las bases para avanzar hacia la etapa final del proyecto: la integración de la plataforma con el ecosistema Robotat. Con el prototipo completamente ensamblado y funcional, el siguiente capítulo aborda el proceso de integración, el cual incluyó tanto la programación del *firmware* de la plataforma móvil como la implementación del control mediante el sistema de captura de movimiento y el establecimiento de comunicación inalámbrica por Wi-Fi.

---

## Integración de la plataforma móvil al ecosistema Robotat

---

La parte crucial de la plataforma móvil correspondía a su control, encargado tanto de los componentes físicos, como de la comunicación con el sistema Robotat. Para ello, se desarrolló una arquitectura modular de software que permitió integrar la gestión en tiempo real de tareas, el procesamiento de datos de sensores, la comunicación inalámbrica y las estrategias de movimiento, garantizando un funcionamiento coordinado y confiable de la plataforma.

### 9.1. Módulos de las tareas dentro del ESP32 (*FreeRTOS*)

La integración de diferentes tareas dentro del ESP32 se obtuvo mediante la implementación de *FreeRTOS* en el ESP32, que se realizó utilizando la librería provista por ESP-IDF, características se explicaron en el marco teórico.

- Núcleo 0:
  - Tarea de comunicación UDP con el Robotat (Sección 9.4.1): esta tarea recibe y procesa los comandos recibidos por Wi-Fi, modifica las variables y notifica a la tarea de actualización de los temporizadores.
- Núcleo 1:
  - Temporizadores de los DRV8825 (Sección 9.2) : esta tarea es la que permite mandar los pulsos con un cierto período a todos los DRV8825. Sólo cambia el estado de los pines *STEP* de cada uno de los motores.
  - Actualización de los temporizadores de los DRV8825 (Sección 9.2): esta tarea procesa los cambios en alguna de las variables de control. Si hay cambios, se

modifica el período de los DRV8825. Para que se ejecute esta tarea es necesario que la tarea de la comunicación Wi-Fi le notifique.

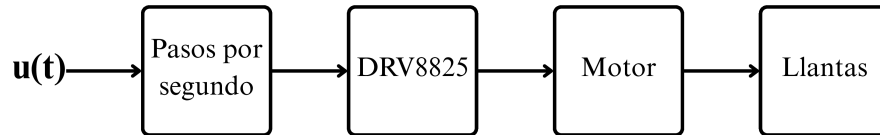
- Lectura y control PID usando los AS5600 (Sección 9.3): se realiza usando el temporizador de alta resolución del ESP32. El error, si está fuera de la tolerancia aceptada, es enviado a la tarea de actualización de los temporizadores.

## 9.2. Control de los motores en lazo abierto

Para el control de los motores paso a paso, como se mostró en el Capítulo 7, se utilizó el DRV8825. El control realizado con el ESP32 funciona de la siguiente manera: se envía un pulso al controlador y cada pulso genera un paso dentro del motor; el período de los pulsos determina la velocidad final del motor. Este sistema se ve representado en la Figura 18, donde  $u(t)$  es la velocidad requerida en rpms ( $\omega$ ) y donde el primer bloque utiliza la ecuación (8) (en microsegundos) para realizar la conversión adecuada. Esto ocurre en un sistema en lazo abierto, pues no hay realmente una certeza de que el paso se realizó correctamente.

$$T_{\mu s} = \frac{60000000}{\text{pasos por revolución} \cdot \text{micropasos} \cdot \omega} \quad [\mu s]. \quad (8)$$

**Figura 18.** Diagrama de control en lazo abierto de los motores



Nota. Elaboración propia.

### 9.2.1. Temporizadores del ESP32

Para poder realizar esta función, se utilizaron los *General Purpose Timers*. Estos son 2 grupos con dos temporizadores cada uno de 64 bits con preescaladores de 16 bits y contadores ascendentes/descendentes de 64 bits con recarga automática [28]. Estos fueron utilizados debido que, para reducir el error lo más posible en lazo abierto, se deben enviar los pulsos en el período requerido de la manera más precisa posible.

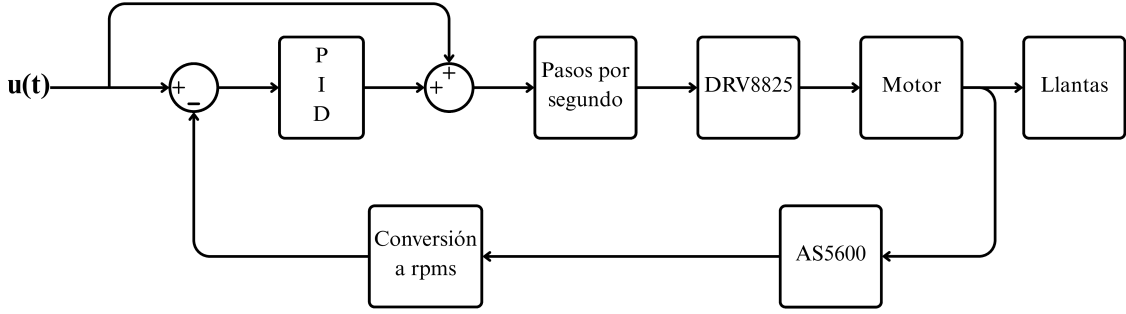
Los motores utilizados operan en un rango de velocidades angulares de  $\omega \in [0, 400]$  rpm. Esto debido a las limitaciones de velocidad propias de los motores, al ruido en el voltaje que generan a grandes velocidades y a la capacidad de procesamiento del sistema. Al sustituir en (8) estos valores, y tomando en cuenta que los motores se controlan con 32 micropasos, se obtuvo que el período más pequeño es de  $T_{\mu s} = 23.43 \mu s$ .

### 9.3. Control de los motores en lazo cerrado

Para mejorar la precisión del motor se añadió el sensor AS5600, lo que permitió implementar un control en lazo cerrado como el mostrado en la Figura 19, que combina retroalimentación y realimentación anticipada. Esta última consiste en iniciar la planta con un valor inicial y sumar el error para alcanzar el valor deseado más rápidamente. La retroalimentación se obtenía mediante la lectura del AS5600, que entrega un valor entre [0–4095] equivalente a [0°–360°], el cual se convierte a rpm ( $\omega$ ) según la ecuación (9), con un período de muestreo de 50 ms gestionado por el ESP32.

$$\omega = \frac{\theta_f - \theta_i}{T_{s\mu s}} \cdot \frac{60 \times 10^6}{2\pi} \text{ [RPM]}. \quad (9)$$

**Figura 19.** Diagrama de control en lazo cerrado de los motores



Nota. Elaboración propia.

El período de muestreo de  $T_s = 50$  ms se eligió balanceando la  $\omega$  máxima y mínima junto al ruido analógico del ADC del ESP32, pues  $T_s$  debe ser lo bastante corto para evitar que el motor avance más de  $360^\circ$  por muestra (11) pero no tan corto en velocidades bajas que el ruido haga parecer inexistente el movimiento (10). Además, el criterio de Nyquist–Shannon obliga a muestrear por encima del doble de la frecuencia máxima asociada a  $\omega_{\text{máx}}$ , evitando aliasing [29]. Sustituyendo los valores dentro de la ecuación (11) se obtuvo  $T_s \in [0.014, 0.075]$  s, por lo que  $T_s = 0.05$  s queda claramente dentro del rango válido.

$$\Delta\theta_{\text{ADC}} = \frac{360^\circ}{2^{12}} = \frac{360^\circ}{4096} \approx 0.088^\circ, \quad (10)$$

$$\frac{\Delta\theta_{\text{ADC}}}{6\omega_{\text{mín}}} < T_s < \frac{360^\circ}{2 \cdot 6\omega_{\text{máx}}} = \frac{30}{\omega_{\text{máx}}}. \quad (11)$$

### 9.3.1. Temporizador de alta resolución del ESP32

Debido a que los motores están utilizando los 4 temporizadores que tiene el ESP32, se optó utilizar el *ESP Timer (High Resolution Timer)*. Esta prestación permite generar temporizadores de *software* de alta resolución (64 bits) que no tienen todas las características que sí tienen los temporizadores de propósito general, pero pueden ser usados para realizar acciones periódicas con una alta resolución [30]. De esta manera se configuró un temporizador para que cada 50 ms lea el valor de todos los AS5600 y compare con la lectura anterior para obtener la velocidad angular (RPM) (9). Por último, se compara la velocidad medida con la velocidad esperada y se obtiene el error de control. Esto queda expresado en la ecuación (12). De esta forma fue posible controlar los cuatro motores en lazo cerrado mediante la ley PID (13).

### 9.3.2. Controlador PID

Debido que el control en lazo cerrado buscaba mejorar la precisión de la velocidad de los motores, se realizaron comparaciones experimentales para encontrar valores adecuados de  $K_p$ ,  $K_i$  y  $K_d$ . Como referencia de velocidad se empleó un tacómetro externo, lo que permitió comparar el desempeño del sistema y afinar las ganancias del PID. La ley de control utilizada corresponde a la forma discreta del PID presentada en (13), basado en lo explicado en [31].

$$e[k] = \omega_{\text{ref}}[k] - \omega_{\text{meas}}[k], \quad (12)$$

$$u_m[k] = K_p e_m[k] + K_i \sum_{n=0}^k e_m[n] \Delta t + K_d \frac{e_m[k] - e_m[k-1]}{\Delta t}. \quad (13)$$

Los resultados experimentales se encuentran en los Cuadros 6 y 7. Los valores de las constantes PID son relativamente pequeños dado que el sistema dispone de una realimentación anticipada. Por tanto el PID solo corrige errores residuales. A pesar de que el controlador regula respecto a la lectura del AS5600 y persiste una discrepancia con el tacómetro externo, esto no afectó a la funcionalidad del PID dado que se calibraron las constantes para reducir el error real no el error calculado.

El control en lazo cerrado logró reducir el error en el rango aproximado de 200 a 10 RPM. Sin embargo, en los rangos más altos, el error no se redujo de forma apreciable (aunque tampoco se incrementó). Esto puede deberse a que el AS5600 junto con el ESP32 presentaban dificultades para calcular las RPM cuando las lecturas angulares estaban cerca de  $0^\circ$  o  $360^\circ$ ; por lo que se limitó el rango utilizado por el controlador a  $[30^\circ, 270^\circ]$ . Además, el periodo de muestreo relativamente alto de 50 ms (20 Hz) también restringe la capacidad de estimar el error a velocidades muy elevadas, ya que la derivación temporal de la velocidad se vuelve menos precisa.

**Cuadro 6.** Error velocidad en lazo abierto

RPMS deseados	RPMS tacómetro	RPMS AS5600	Error calculado	Error real
400	426.1	425.30	-6.32 %	6.53 %
300	312.5	316.59	-5.53 %	4.17 %
200	203.7	203.95	-1.98 %	1.85 %
100	99.7	98.99	1.01 %	-0.30 %
50	55.2	50.55	-1.11 %	10.40 %
25	25	25.02	-0.09 %	0.00 %
10	10.8	9.82	1.83 %	8.00 %

Nota. El cuadro muestra los valores de error del control en lazo abierto. Elaboración propia.

**Cuadro 7.** Error velocidad en lazo cerrado

RPMS deseados	RPMS tacómetro	RPMS AS5600	Error calculado	Error real
400	426.1	423.93	-5.98 %	6.53 %
300	312.5	309.59	-3.20 %	4.17 %
200	201.9	201.6	-0.78 %	0.95 %
100	99.7	99.43	0.57 %	-0.30 %
50	50.2	51.53	-3.07 %	0.40 %
25	25	25.11	-0.44 %	0.00 %
10	9.9	10.95	-9.50 %	-1.00 %

Nota. El cuadro muestra los valores de error del control en lazo cerrado usando:  $K_p = 0.04$ ,  $K_i = 0.004$  y  $K_d = 0.0002$ . Elaboración propia.

### 9.3.3. Configuración AS5600

La configuración de los sensores AS5600 requiere un procedimiento para garantizar su funcionamiento en la lectura del ADC. La hoja de datos [14] indica que el proceso incluye la inicialización del bus I<sup>2</sup>C, la verificación de la presencia del imán y la confirmación de que no existan configuraciones previas. Posteriormente, se ajustan los parámetros del registro de configuración (CONF\_REG) para optimizar la velocidad de salida, el filtrado y el modo de operación en salida analógica, además de fijar el rango angular máximo (360°).

Luego, se graban los valores en memoria mediante comandos (BURN\_SETTING y BURN\_ANGLE), asegurando que la configuración del sensor quede definida de manera permanente. Este procedimiento puede implementarse con cualquier microcontrolador que soporte I<sup>2</sup>C. En este caso la configuración se realizó con un ESP32.

Sin embargo, en algunas versiones del AS5600, este proceso puede no funcionar correctamente. Por lo mismo, se tiene la opción de conectar el pin GPO directamente al voltaje de alimentación. Esto, en algunas versiones, activa la señal analógica correctamente sin necesidad de realizar la configuración con I<sup>2</sup>C. Es importante aclarar que el proceso realizado a través de I<sup>2</sup>C necesita que GPO esté flotando o a tierra.

## 9.4. Comunicación con el ecosistema Robotat

La comunicación con el Robotat se describe en la Figura 11. Básicamente, el usuario envía los comandos de control a través de su computadora hacia la plataforma móvil (ESP32) a través de la red inalámbrica dentro del ecosistema Robotat. La recepción de los comandos se realiza mediante el módulo Wi-Fi que se incorpora dentro del ESP32.

### 9.4.1. Módulo Wi-Fi

Este módulo permite que el ESP32 actúe como un cliente que escucha continuamente los comandos que envía el usuario. La comunicación entre estos dispositivos se realiza mediante el protocolo UDP. Este diseño permite que el ESP32 utilice sus recursos principalmente en la ejecución del control de los motores y sensores, minimizando el procesamiento relacionado con la verificación de la recepción de datos. El módulo Wi-Fi se configura para escuchar en un puerto específico y acepta paquetes entrantes desde cualquier dirección IP autorizada. Esto permite cierta flexibilidad, ya que el robot puede recibir comandos desde distintos dispositivos y programas, como una computadora local.

### 9.4.2. Interpretación de comandos y control inalámbrico

**Cuadro 8.** Comandos y funciones del sistema

(a) Comandos (globales y por motor).

Comando	Descripción / Notas
<code>start/stop</code>	Arrancan o detienen los cuatro motores simultáneamente.
<code>mX_rpm=</code> <code>RPMS</code>	Fija la velocidad del motor X en RPM. Sólo establece la magnitud (la dirección se define por separado).
<code>mX_dir=</code> <code>CW/CCW</code>	Define la dirección de giro del motor X ( <i>CW = Clockwise</i> , <i>CCW = Counter-Clockwise</i> ).
<code>mX_start</code> / <code>mX_stop</code>	Arrancan o detienen de forma individual el motor X.

(b) Funciones de control inalámbrico (Python / MATLAB).

Función	Descripción / Notas
<code>robot_connect()</code>	Configura el puerto/endpoint UDP para comunicarse con el ESP32.
<code>robot_start()</code> / <code>robot_stop()</code>	Inician o detienen el robot usando los últimos valores recibidos.
<code>robot_set_velocities()</code>	Envía un vector con las RPM de las cuatro ruedas (determinantes de $\omega_{bz}$ , $v_{bx}$ , $v_{by}$ ).

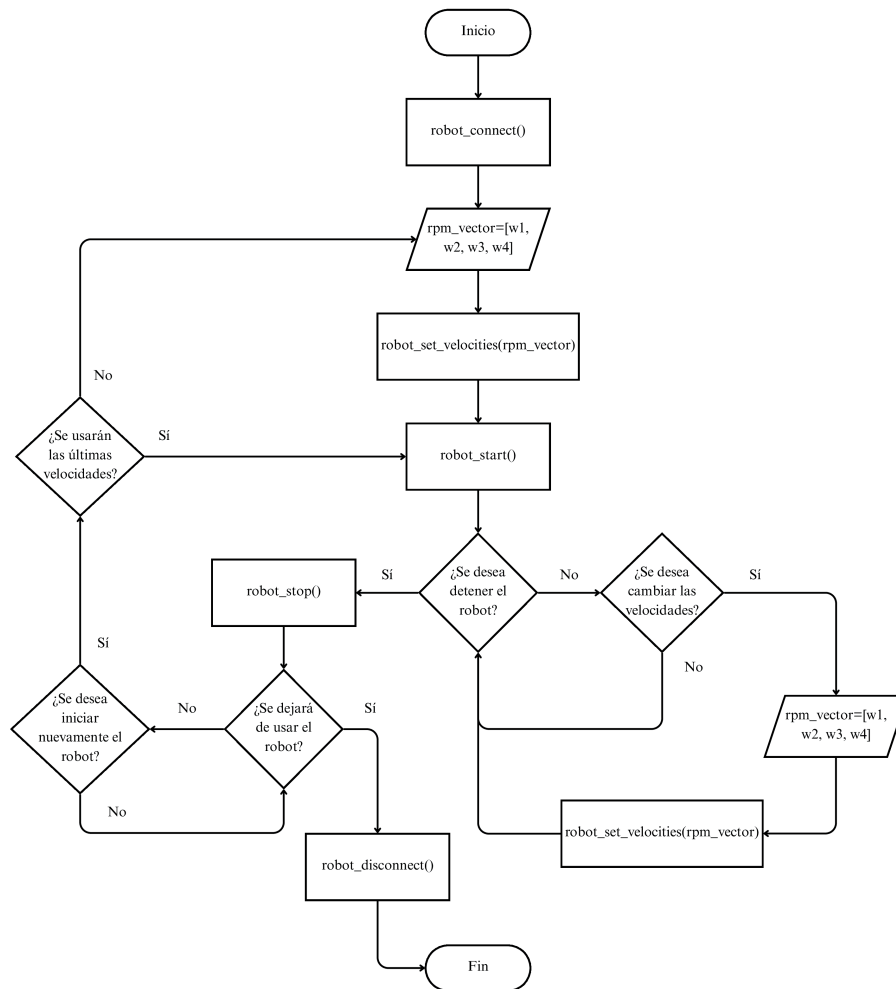
Nota. Elaboración propia.

La interpretación de los comandos recibidos se realiza en una tarea dentro del programa del ESP32. Luego de procesar cualquier comando válido, se notifica a la tarea encargada

del control del motor (`hMotorControlTaskHandle`) para que aplique los cambios correspondientes. Por otra parte, se realizaron funciones/librerías con Python y Matlab para poder mandar dichos comandos de manera más sencilla al ESP32 a través de Wi-Fi. En ambos casos dichas funciones se pueden visualizar en el Cuadro 8.

Para comprender la lógica de control desde la perspectiva del usuario, la Figura 20 presenta de forma general cómo deben utilizarse estas funciones. Es importante destacar que la mayor diferencia entre las funciones `robot_set_velocities()` y `robot_stop()` es que enviar un vector con velocidades iguales a cero usando la primera función detendrá al robot; sin embargo, si inmediatamente se envía otro vector con velocidades distintas de cero, el robot reanudará su movimiento de forma inmediata. Mientras que al usar la segunda función, el robot apagará sus motores, lo que le permitirá ahorrar batería.

**Figura 20.** Pseudocódigo de control del robot



Nota. Elaboración propia.

## 9.5. Configuración dentro del Robotat

Para la configuración de la plataforma móvil dentro del Robotat se colocaron marcadores en los laterales de la plataforma como se ve en la Figura 21. Estos marcadores son los que permiten que las cámaras *OptiTrack* detecten un cuerpo rígido. Gracias al diseño de la plataforma, los marcadores pueden sujetarse mediante tuercas M3, como se explicó anteriormente. Con esta disposición, la plataforma quedó registrada en el sistema Robotat como el agente/cuerpo rígido número 29 (identificador interno del sistema).

**Figura 21.** Ubicación de los marcadores en los laterales de la plataforma



Nota. Soportes con tuercas M3. Elaboración propia.

Para obtener correctamente la posición y orientación de la plataforma dentro del Robotat, se siguieron los siguientes pasos de calibración:

1. Colocar la plataforma en el centro del área de captura del Robotat, con la dirección positiva del eje  $X$  tanto de la plataforma como del Robotat alineadas.
2. Visualizar la posición actual de los marcadores en la interfaz del sistema.
3. Leer los valores de posición y orientación desde MATLAB; estos valores serán las compensaciones para determinar con precisión el centro de la plataforma.

El resultado de este proceso fue:

$$x_{\text{offset}} = 19.1237 \text{ mm}, \quad y_{\text{offset}} = -115.3506 \text{ mm}, \quad \psi_{\text{offset}} = -72.6029^\circ$$

Para el eje  $Z$  se registró la altura real del centro de la plataforma: 72 mm, mientras que las medidas de *pitch* ( $\theta$ ) y *roll* ( $\phi$ ) se consideraron nulas (0), debido a que la plataforma móvil no permite variar estos ángulos ni la altura. En cambio, el ángulo *yaw* ( $\psi$ ) y las posiciones  $X$  y  $Y$  pueden presentar un desplazamiento y por ello se obtuvieron las compensaciones respectivas.

## 9.6. Control de trayectorias

Para comprobar el correcto funcionamiento del control de la plataforma móvil (Sección 6.5.2) se implementaron scripts en MATLAB que obtuvieron la trayectoria deseada  $\mathbf{q}_d(t) =$

$[\mathbf{x}_d, \mathbf{y}_d, \theta_d]^\top$  y la configuración actual  $\mathbf{q}(t) = [\mathbf{x}, \mathbf{y}, \theta]^\top$  publicada por el servidor Robotat. En este caso se optó por 3 diferentes pruebas: una prueba de seguimiento, una prueba que muestra el movimiento omnidireccional de la plataforma móvil y una prueba de planificación para evitar obstáculos. En la primera prueba se ajustaron las ganancias del controlador:

$$K_p = \begin{bmatrix} 1.6 & 0 & 0 \\ 0 & 1.6 & 0 \\ 0 & 0 & 0.8 \end{bmatrix}, \quad K_i = \begin{bmatrix} 0.15 & 0 & 0 \\ 0 & 0.15 & 0 \\ 0 & 0 & 0.15 \end{bmatrix},$$

En las tres pruebas se producían referencias de velocidad para la ley de realimentación anticipada + PI de la Ecuación (7). Esto permitió que, con el modelo cinemático (6), se obtuvieran las velocidades angulares de las ruedas  $\mathbf{u}(t)$  (transformandolas de rad/s a RPM). Las dimensiones físicas usadas en las pruebas fueron  $w = 69.3$  mm,  $l = 53.5$  mm y  $r = 48$  mm. Asimismo, se aplicó una saturación a las ruedas de  $\text{RPM}_{\text{máx}} = 100$  y, si alguna componente excedía el límite, se escalaba el vector de salida para mantener la proporcionalidad entre ruedas y permitir el movimiento esperado. También se aplicó *anti-windup* simple para que el integrador no tuviera acumulación indebida.

Asimismo, debido a las amplias posibilidades de movimiento de un robot omnidireccional, se definieron políticas de rotación: rotación solo al llegar al objetivo o nunca rotar para mantener el eje X siempre alineado. También se utilizó una estrategia de *lookahead* punto a punto, donde la plataforma avanzaba hasta que la distancia al punto objetivo fuera menor a la distancia establecida, usando valores porcentuales del tamaño de la plataforma móvil.

Por otra parte, para comprobar el comportamiento adecuado de la plataforma móvil, se calcularon 2 valores diferentes. El primer valor fue el error de raíz cuadrada media (RMSE por sus siglas en inglés) entre la posición final deseada y la real. Mientras que el segundo valor fue el error máximo entre la trayectoria realizada y la deseada. Ambos parámetros deberían mostrar un valor muy cercano al *lookahead* designado.

### 9.6.1. Trayectoria de seguimiento

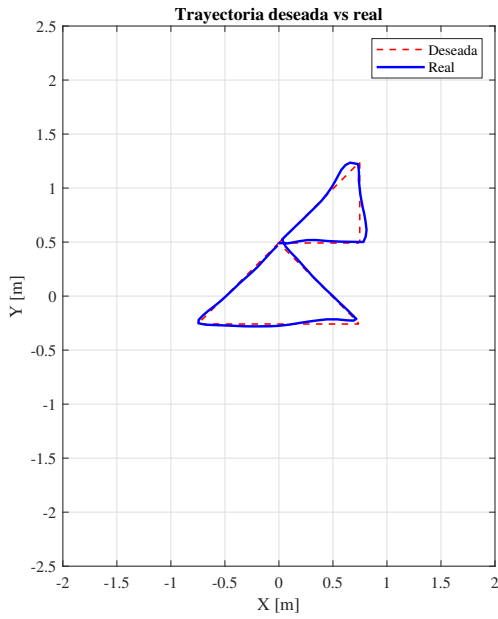
En esta primera prueba el objetivo fue verificar la capacidad de la plataforma móvil para seguir en tiempo real a otro objeto (marcador) dentro del Robotat, manteniendo baja la desviación entre la posición del marcador y de la plataforma móvil. El código obtenía la posición del marcador ( $\mathbf{q}_d(t)$ ) y la posición de la plataforma móvil ( $\mathbf{q}(t)$ ) y calculaba el error de la posición. A partir de estos valores se calculaban las velocidades de referencia de la plataforma y las velocidades de las ruedas en RPM. Esta prueba fue primordial, pues permitió el ajuste inicial de las ganancias del controlador PI.

### 9.6.2. Trayectoria omnidireccional

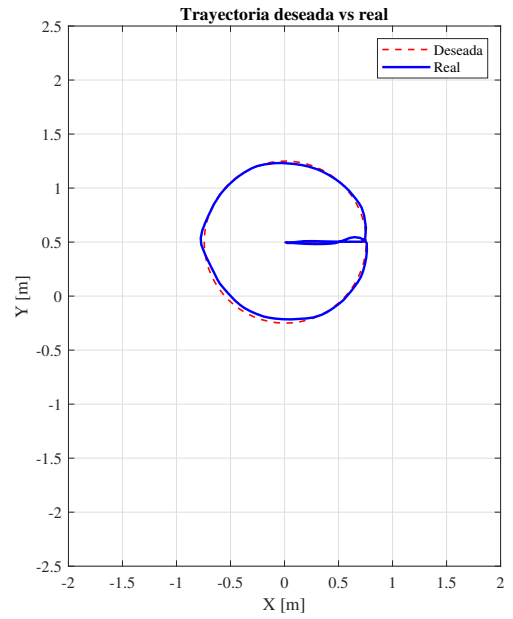
En la segunda prueba el objetivo fue verificar la capacidad de la plataforma como una plataforma móvil omnidireccional. Para ello se decidió utilizar una trayectoria predefinida con movimientos en varias direcciones. Esta trayectoria se puede ver en la Figura 22. Al comparar la trayectoria esperada con la trayectoria obtenida, la prueba demostró el correcto funcionamiento del robot como plataforma omnidireccional.

**Figura 22.** Comparativa trayectorias

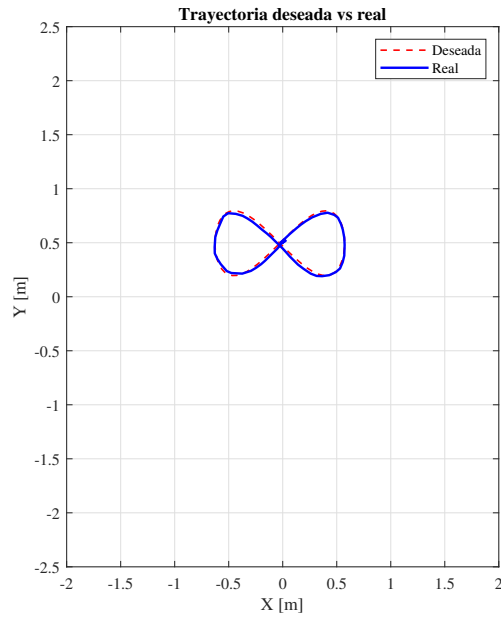
(a) Trayectoria movimientos omnidireccionales.



(b) Trayectoria circular.



(c) Trayectoria en forma de 8.

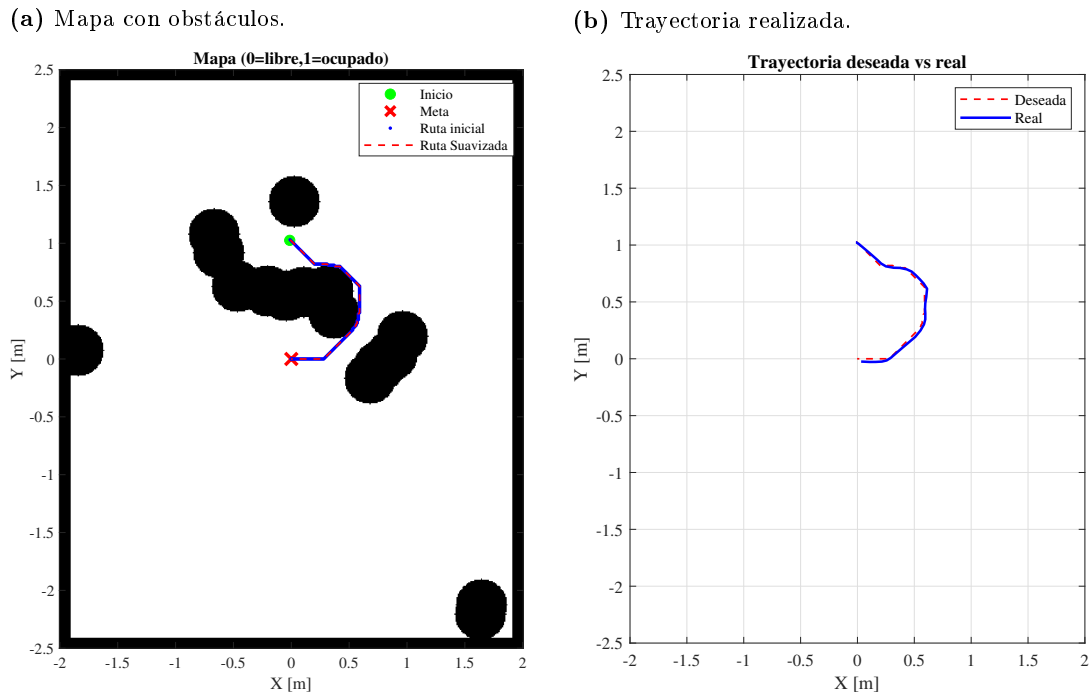


Nota. La figura muestra la comparativa entre la trayectoria deseada y la trayectoria realizada por la plataforma en las trayectorias predefinidas.  
Elaboración propia.

### 9.6.3. Trayectoria con obstáculos

En la tercera prueba el objetivo fue verificar la capacidad de la plataforma de ejecutar trayectorias que requieren evitar obstáculos. Esto debido a que en este tipo de trayectorias, la precisión del movimiento es indispensable para conseguir evitar los obstáculos. En el código se leían las poses desde el Robotat (obstáculos, robot y meta), se generaba la cuadrícula de ocupación, se ejecutaba  $D^*$  para la planificación de la trayectoria y se ajustaba la trayectoria con *splines* (ajuste con polinomios). En esta prueba,  $\mathbf{q}_d(t)$  eran los valores generados por la planificación y  $\mathbf{q}(t)$  la pose actual del robot. En la Figura 23 se muestra la comparación entre  $\mathbf{q}_d(t)$  y la trayectoria seguida por la plataforma.

**Figura 23.** Comparativa trayectoria del mapa



Nota. La figura muestra la comparativa entre la trayectoria deseada y la trayectoria realizada por la plataforma en la planificación con obstáculos. Elaboración propia.

### 9.6.4. Resultados de las trayectorias

En el Cuadro 9 se muestran los datos más relevantes de las trayectorias realizadas. Por otra parte en la Figura 24 puede observarse la evolución del error a lo largo del tiempo. Los mayores valores de RMSE en las primeras trayectorias se deben principalmente a que dichas trayectorias tenían menos puntos de referencia y por lo tanto los saltos entre puntos consecutivos son mayores. En cambio, la última trayectoria contaba con una mayor densidad de puntos (gracias a los *splines*), lo que permitió correcciones más finas y continuas y se tradujo en un RMSE mucho menor (5.11,mm). Estos resultados demuestran que la implementación

dentro del Robotat fue exitosa al contar con la capacidad de realizar trayectorias con una alta precisión, mostrando valores de RMSE de entre 5.11 mm y 36.91 mm y errores máximos de entre 10.19 mm y 70.7 mm.

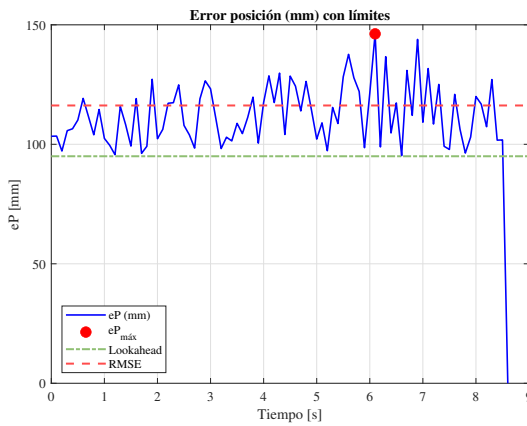
**Cuadro 9.** Análisis de trayectorias

Trayectoria	RMSE (mm)	Error Máximo (mm)
Omnidireccional	21.25	51.22
Círculo	36.91	69.76
Ocho	31.47	70.68
Obstáculos	5.11	10.19

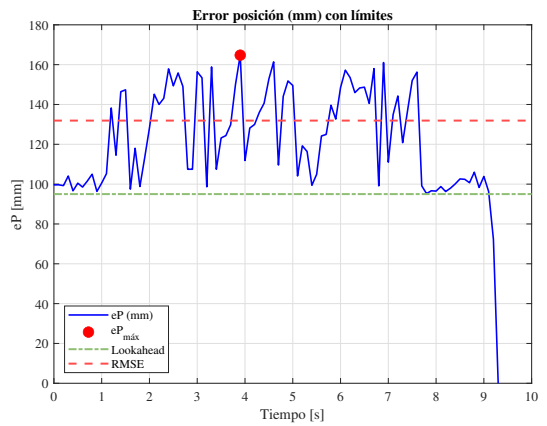
Nota. El cuadro muestra los resultados del análisis de las trayectorias realizadas usando como referencia el valor de *lookahead*. Elaboración propia.

**Figura 24.** Error de posición durante las trayectorias

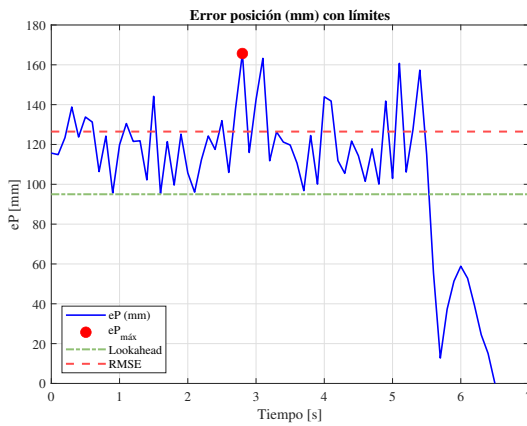
(a) Trayectoria movimientos omnidireccionales.



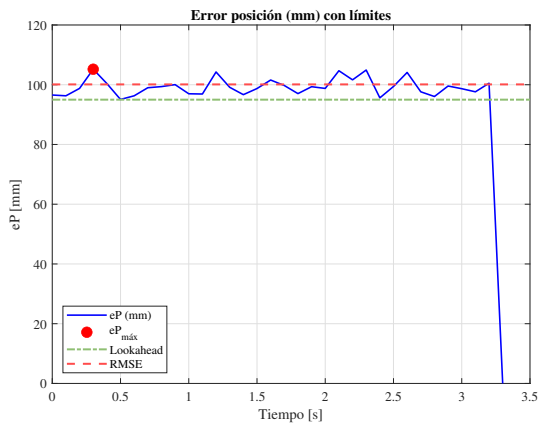
(b) Trayectoria circular.



(c) Trayectoria en forma de 8.



(d) Trayectoria con obstáculos.



Nota. La figura muestra los errores de posición  $e_P$  (mm) frente al tiempo durante la ejecución del seguimiento de trayectorias. Elaboración propia.

- El análisis MCDA permitió seleccionar el motor NEMA 17 de 23 mm (puntaje 1.52) y el DRV8825 (1.47) como la combinación óptima bajo disponibilidad local, validando cuantitativamente su idoneidad para la plataforma.
- El diseño electrónico garantizó una operación estable con un consumo experimental de 1.492 A, una autonomía estimada de 3.4 h y una reducción del error de velocidad hasta valores cercanos al 1 % en el rango útil de operación.
- El diseño mecánico modular, con dimensiones finales de  $190 \times 190 \times 82$  mm, permitió integrar todos los sistemas de control y potencia manteniendo rigidez estructural, ventilación adecuada y posibilidad de expansión.
- Se priorizaron componentes con disponibilidad local y procesos de manufactura digital para maximizar la replicabilidad, con un coste total de Q2,429.38.
- Se logró diseñar un plataforma omnidireccional con capacidad de realizar trayectorias con una alta precisión, mostrando valores de RMSE de entre 5.11 mm y 36.91 mm y errores máximos de entre 10.19 mm y 70.7 mm.

- Se exhorta a explorar aplicaciones avanzadas de la plataforma móvil dentro del ecosistema Robotat, tales como su uso como obstáculo móvil dinámico, ejecución de tareas de *pick-and-place* mediante manipuladores seriales como el MaxArm y colaboración con drones en experimentos de control cooperativo.
- Se sugiere integrar odometría y fusión sensorial en futuros desarrollos, combinando las lecturas del AS5600 con modelos cinemáticos y filtros de estimación como el filtro de Kalman, para mejorar la precisión y velocidad en la localización y seguimiento de trayectorias.
- Se recomienda incorporar sensores de percepción como LiDAR o cámaras, junto a algoritmos de SLAM, permitiendo evadir obstáculos y planificar rutas dentro de entornos no controlados.
- Para futuras iteraciones se aconseja flexibilizar la restricción de componentes locales para evaluar alternativas de mayor rendimiento que permitan incrementar la precisión y robustez de la plataforma.
- Para lograr un diseño replicable de llantas omnidireccionales impresas en 3D, se recomienda investigar materiales de baja fricción y explorar el uso de rodamientos miniatura para los rodillos, con el fin de obtener un comportamiento más homogéneo y cercano a ruedas mecanum comerciales.

- [1] C. P. Montoya, «Robotat: un ecosistema robótico de captura de movimiento y comunicación inalámbrica,» Trabajo de graduación para optar al grado académico de Licenciado en Ingeniería Mecatrónica, Trabajo de graduación, Universidad del Valle de Guatemala, Guatemala, 2022.
- [2] OptiTrack, *OptiTrack Documentation: Motive*, ver. v3.3, Suite 350, 5431 NE 20th Ave, Vancouver, WA 98663, USA, ago. de 2025. dirección: <https://docs.optitrack.com/motive>.
- [3] E. I. Hernández Woc et al., «Reingeniería de Megaproyectos Fase I,» Mega-Proyecto, Universidad del Valle de Guatemala, 2017.
- [4] C. J. Lima Cerdón, «Diseño e implementación de una nueva plataforma móvil para aplicaciones en robótica de enjambre.,» Tesis doct. Universidad del Valle de Guatemala, 2018.
- [5] J. A. Rodríguez Xicar, «Diseño e Implementación de una Plataforma Móvil Para Aplicaciones de Robótica de Enjambre - Fase III,» Tesis doct. Universidad del Valle de Guatemala, 2022.
- [6] D. A. González López, «Diseño e implementación de una plataforma robótica móvil para emplearse dentro de un sistema de captura de movimiento,» Tesis doct. Universidad del Valle de Guatemala, 2023.
- [7] C. C. Paim, A. L. d. Costa y T. P. d. Nascimento, «AxeBot Robot the Mechanical Design for an Autonomous Omnidirectional Mobile Robot,» en *Electronics, Robotics and Automotive Mechanics Conference*, Los Alamitos, CA, USA: IEEE Computer Society, 2009, págs. 187-192. DOI: 10.1109/CERMA.2009.77.
- [8] B. J. C. Otoy, «Control de Trayectoria de un Robot Móvil Omnidireccional en un Entorno Controlado,» Trabajo de graduación, Universidad del Azuay, Cuenca, Ecuador, 2023.

- [9] S. Greco, M. Ehrgott y J. Figueira, eds., *Multiple criteria decision analysis: state of the art surveys* (Multiple Criteria Decision Analysis), 2.<sup>a</sup> ed. United States: Springer Science + Business Media, mar. de 2016, ISBN: 978-1-4939-3093-7. DOI: 10.1007/978-1-4939-3094-4.
- [10] I. Linkov, E. Moberg, B. D. Trump, B. Yatsalo y J. M. Keisler, *Multi-Criteria Decision Analysis: Case Studies in Engineering and the Environment*, 2.<sup>a</sup> ed. Boca Raton, FL: CRC Press, 2020, ISBN: 9780367345334.
- [11] P. Linear, *Stepper Motor Support Datasheet*. dirección: <https://pages.pbcllinear.com/rs/909-BFY-775/images/Data-Sheet-Stepper-Motor-Support.pdf>.
- [12] T. A. Bigelow, *Electric Circuits, Systems, and Motors*, 1.<sup>a</sup> ed. Springer Cham, 2020, ISBN: 978-3-030-31354-8. DOI: 10.1007/978-3-030-31355-5.
- [13] Texas Instruments, *DRV8825 Stepper Motor Controller IC Datasheet (Rev. F)*, 2014. dirección: <https://www.ti.com/lit/ds/symlink/drv8825.pdf>.
- [14] ams AG, *AS5600: 12-Bit Programmable Contactless Potentiometer*, ver. v1-06, Tobelbader Strasse 30, 8141 Premstaetten, Austria, jun. de 2018. dirección: <https://ams.com/AS5600>.
- [15] TinyTronics, *AS5600 Magnetic Angle Sensor Encoder Module*, 2025. dirección: <https://www.tinytronics.nl/en/sensors/magnetic-field/as5600-magnetic-angle-sensor-encoder-module>.
- [16] E. Systems, *ESP32-WROOM-32 Datasheet*, mar. de 2023. dirección: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf)  
[https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf).
- [17] La Electrónica, *Módulo Wi-Fi + Bluetooth ESP32 CH340*, 2025. dirección: <https://laelectronica.com.gt/modulo-wifi-bluetooth-esp32-ch340>.
- [18] Espressif Systems, *lwIP — API Guide (ESP-IDF Programming Guide)*, 2025. dirección: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/lwip.html>.
- [19] Espressif Systems, *FreeRTOS - ESP-IDF Programming Guide; ESP-IDF programming Guide v5.5*, 2025. dirección: [https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/system/freertos\\_idf.html](https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/system/freertos_idf.html).
- [20] J. Louis E. Frenzel, *Practical Electronic Design for Experimenters*. New York: McGraw-Hill Education TAB, McGraw Hill LLC Professional Division, 2020, ISBN: 978-1-26-045616-5.
- [21] T. B. Reddy y D. Linden, *Linden's Handbook of Batteries*, Fourth. McGraw-Hill, 2011, ISBN: 978-0-07-162419-0.
- [22] R. Siegwart, I. R. Nourbakhsh y D. Scaramuzza, *Introduction to Autonomous Mobile Robots* (Intelligent Robotics and Autonomous Agents Series), 2nd. Cambridge, MA: MIT Press, 2011, ISBN: 978-0-262-01535-6.
- [23] K. M. Lynch y F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017, ISBN: 9781107156302.
- [24] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB®* (Springer Tracts in Advanced Robotics), 2nd. Cham, Switzerland: Springer International Publishing, 2017, vol. 118, ISBN: 978-3-319-54412-0. DOI: 10.1007/978-3-319-54413-7.

- [25] *IPC-2221A: Generic Standard on Printed Board Design*, IPC - Association Connecting Electronics Industries, Northbrook, IL, mayo de 2003. dirección: [https://www-eng.1bl.gov/~shuman/NEXT/CURRENT\\_DESIGN/TP/MATERIALS/IPC-2221A\(L\).pdf](https://www-eng.1bl.gov/~shuman/NEXT/CURRENT_DESIGN/TP/MATERIALS/IPC-2221A(L).pdf).
- [26] *MakerLab UVG — Home*, <https://sites.google.com/view/makerlabuvg/home>, Sitio institucional / laboratorio. visitado 27 de ago. de 2025.
- [27] Samsung SDI Co., Ltd., *ICR18650-26J: Especificación del producto*, No. de especificación ICR18650-26J, Versión 2.0. Hoja de datos (PDF)., Samsung SDI, División de Baterías, jul. de 2016. dirección: ICR18650-26J.PDF.
- [28] Espressif Systems, *General Purpose Timer; ESP-IDF programming Guide v5.5*, <https://docs.espressif.com/projects/esp-idf/en/v4.3/esp32/api-reference/peripherals/timer.html>, 2025.
- [29] D. G. Manolakis y V. K. Ingle, *Applied Digital Signal Processing: Theory and Practice*. Cambridge, UK: Cambridge University Press, 2011, ISBN: 9780521110020.
- [30] Espressif Systems, *ESP Timer (High Resolution Timer); ESP-IDF programming Guide v5.5*, [https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/system/esp\\_timer.html](https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/system/esp_timer.html), 2025.
- [31] D. E. Seborg, T. F. Edgar, D. A. Mellichamp y F. J. Doyle, *Process Dynamics and Control*, 4th. Hoboken, NJ: John Wiley & Sons, 2017, ISBN: 9781119285915.
- [32] L. Ningbo Leison Motor Co., *Catálogo de Motores con Escobilla de Corriente Continua*, 2019. dirección: <https://www.nbleisonmotor.com/download.html>.
- [33] L. Ningbo Leison Motor Co., *Catálogo de Motores de Engranajes de Corriente Continua*, 2019. dirección: <https://www.nbleisonmotor.com/download.html>.
- [34] P. Corporation, *37D Metal Gearmotors*, 2019. dirección: <https://www.pololu.com/file/0J1706/pololu-37d-metal-garmotors.pdf>.
- [35] Rhydolabz, *A2212/13T Brushless DC Motor Datasheet*. dirección: [https://www.rhydolabz.com/documents/26/BLDC\\_A2212\\_13T.pdf](https://www.rhydolabz.com/documents/26/BLDC_A2212_13T.pdf).
- [36] H. Technology, *GA12-N20 Mini DC Gear Motor Datasheet*. dirección: [https://www.handsontec.com/dataspecs/motor\\_fan/GA12-N20.pdf](https://www.handsontec.com/dataspecs/motor_fan/GA12-N20.pdf).
- [37] L. Electrónica, *Motor de pasos NEMA 17 de 23mm*. dirección: <https://laelectronica.com.gt/motor-de-pasos-nema-17-de-23mm>.
- [38] A. Industries, *DC Gearbox Motor - TT Motor - 200RPM - 3 to 6VDC*. dirección: <https://www.adafruit.com/product/3777>.
- [39] Leadshine Technology Co., Ltd., *DM542 Digital Stepper Drive User Manual*, 2012. dirección: <https://kitaez-cnc.com/f/dm542.pdf>.
- [40] Handson Technology, *L298N Dual H-Bridge Motor Driver User Guide*, 2021. dirección: <https://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf>.
- [41] Handson Technology, *TB6600 Stepper Motor Driver Datasheet*, 2019. dirección: <https://www.handsontec.com/dataspecs/module/TB6600-Motor-Driver.pdf>.
- [42] TRINAMIC Motion Control GmbH & Co. KG, *TMC2209 Datasheet*, 2023. dirección: [https://www.analog.com/media/en/technical-documentation/data-sheets/tmc2209\\_datasheet\\_rev1.09.pdf](https://www.analog.com/media/en/technical-documentation/data-sheets/tmc2209_datasheet_rev1.09.pdf).

- [43] N. de Smith, *ANSI IPC-2221A PCB Trace Width Calculator*. dirección: <https://nick.desmith.net/Electronics/TraceWidth.html>.
- [44] T. Chai y R. R. Draxler, «Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature,» *Geoscientific Model Development*, vol. 7, n.º 3, págs. 1247-1250, 2014. DOI: 10.5194/gmd-7-1247-2014. dirección: <https://doi.org/10.5194/gmd-7-1247-2014>.
- [45] OpenAI, *ChatGPT (GPT-5 Thinking)*, Asistente de IA utilizado para apoyo en redacción y revisión de estilo, 2025. dirección: <https://chat.openai.com/>.
- [46] GitHub, *GitHub Copilot*, <https://github.com/features/copilot>, Herramienta de autocompletado y asistencia de programación basada en IA, 2025.

En esta sección se presentan los anexos que complementan y amplían la información del proyecto, organizados siguiendo el mismo orden abordado a lo largo del documento. En primer lugar, se incluyen los procedimientos empleados en el estudio comparativo de los motores y sus respectivos controladores. Posteriormente, se muestran los diseños de las placas de circuito impreso y el proceso de manufactura de las llantas omnidireccionales impresas en 3D. También se incorpora la documentación necesaria para la configuración del sensor AS5600. Más adelante, se proporcionan los enlaces a los repositorios de GitHub que contienen el código utilizado para la programación del ESP32, así como los scripts desarrollados en MATLAB y Python para su control. Finalmente, se adjunta el enlace a los planos de la plataforma móvil, las recomendaciones para su impresión en 3D, el manual de usuario y el manual de ensamblaje de la plataforma móvil.

## 13.1. MCDA

En esta sección se encuentran las ecuaciones usadas para las calificaciones del MCDA. Así como las alternativas y los análisis detallados del MCDA.

### 13.1.1. Ecuaciones para las calificaciones

$$U_i = \pm \frac{2}{\text{máx}(x) - \text{mín}(x)} \cdot (x_i - \text{mín}(x)) \quad (14)$$

Donde  $x_i$  es el valor del criterio para el motor  $i$  y  $x$  es el conjunto de valores para todos los motores. El signo positivo o negativo dependía del impacto del criterio en la decisión.

$$U_i = 2 \cdot \frac{\ln(x_i) - \text{mín}(\ln(x))}{\text{máx}(\ln(x)) - \text{mín}(\ln(x))} \quad (15)$$

Esta transformación también normaliza los valores entre 0 y 2. Su contraparte negativa es:

$$U_i = 2 - 2 \cdot \frac{\ln(x_i) - \text{mín}(\ln(x))}{\text{máx}(\ln(x)) - \text{mín}(\ln(x))} \quad (16)$$

Para algunas calificaciones se utilizó una escala ordinal positiva o su contraparte negativa.

- Alta: 2.0
- Media-Alta: 1.5
- Media: 1.0
- Media-Baja: 0.5
- Baja: 0.0

- Alta: 0.0
- Media-Alta: 0.5
- Media: 1.0
- Media-Baja: 1.5
- Baja: 2.0

### 13.1.2. Alternativas de motores

**Cuadro 10.** Comparativa de motores

(a) Tipo, voltaje y corriente.

Modelo motor	Tipo de motor	Voltaje (V)	Corriente (A)
555 [32]	Motor DC	12	1.8
775 [32]	Motor DC	24	1.70
25GA370 [33]	Motoreductor	12	0.11
37D [34]	Motoreductor	12	0.66
A2212 [35]	Brushless	11.1	10
N20 [36]	Motoreductor	6	0.67
Nema 17 (23mm) [37]	Stepper	4.1	2
Nema 17 (40mm) [11]	Stepper	2.2	3.32
TT Motor [38]	Motoreductor	6	0.25

(b) Tamaño, torque y complejidad de control.

Modelo motor	Tamaño (mm)	Torque máx (Nm)	Complejidad
555	35x35x78	0.02	Baja
775	44.4x44.4x98	0.09	Baja
25GA370	24.4x24.4x63.3	0.147	Media-Alta
37D	92.1x34.8x34.8	0.21	Media-Alta
A2212	28x28x40	0.83	Alta
N20	12x12x34	0.2	Baja
Nema 17 (23mm)	43x43x23	0.14	Media-Alta
Nema 17 (40mm)	43x43x40	0.25	Media-Alta
TT Motor	22.3x22.5x69.5	0.08	Baja

(c) Velocidad, precio y características de control.

Modelo motor	Velocidad máx (rpm)	Precio (Q)	Precisión
555	8000	79	Baja
775	21000	185	Baja
25GA370	200	175	Alta
37D	180	300	Alta
A2212	11100	105	Media-Alta
N20	100	55	Baja
Nema 17 (23mm)	1000	140	Alta
Nema 17 (40mm)	1000	175	Alta
TT Motor	250	19	Baja

Nota. El cuadro muestra características eléctricas, físicas y de control de las alternativas de motores. Elaboración propia.

### 13.1.3. Alternativas de controladores

**Cuadro 11.** Comparativa de controladores para motores paso a paso

(a) Disipación de calor y control de velocidad.

Modelo controlador	Disipación de calor	Velocidad (kHz)
DM542 [39]	Media-Alta	200
DRV8825 [13]	Media	250
L298N [40]	Media-Baja	100
TB6600 [41]	Media-Alta	20
TMC2209/8 [42]	Alta	250

(b) Precisión de posición y tamaño.

Modelo controlador	Precisión de posición	Tamaño (mm)
DM542	Media-Alta	118x75.5x34
DRV8825	Media	20x15x10
L298N	Baja	34x43x27
TB6600	Media	960x560x330
TMC2209/8	Media	20x15x10

(c) Micropasos.

Modelo controlador	Micropasos
DM542**	1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/5, 1/10, 1/20, 1/25, 1/40, 1/50, 1/100, 1/125
DRV8825	1/2, 1/4, 1/8, 1/16, 1/32
L298N	N/A
TB6600	1/2, 1/4, 1/8, 1/16, 1/32 **
TMC2209/8	1/2*, 1/4*, 1/8, 1/16, 1/32, 1/64, 1/128*, 1/256*

(d) Protección eléctrica y precio.

Modelo controlador	Protección eléctrica	Precio
DM542	Alta	Q325
DRV8825	Media-Baja	Q38
L298N	Baja	Q45
TB6600	Alta	Q169
TMC2209/8	Media-Alta	Q70

Nota. \* Requiere modo UART para habilitar algunos micropasos.

\*\* Configuración de micropasos mediante switches físicos, no programable por software.

Elaboración propia.

### 13.1.4. Análisis comparativo

**Cuadro 12.** Análisis comparativo de los modelos de motores

(a) Motores: 555, 775, 25GA370.

Criterio	Peso	555	775	25GA370
Complejidad (control)	4.08 %	2.00	2.00	0.50
Consumo	18.87 %	1.63	1.28	2.00
Precio (Q)	2.30 %	1.57	0.82	0.89
Precisión	39.17 %	0.00	0.00	2.00
Tamaño	8.36 %	1.04	0.00	1.65
Torque	8.36 %	0.00	0.18	0.31
Velocidad	18.87 %	1.64	2.00	0.26
<b>Total ponderado</b>	100 %	0.86	0.49	1.41

(b) Motores: 37D, A2212, N20.

Criterio	Peso	37D	A2212	N20
Complejidad (control)	4.08 %	0.50	0.00	2.00
Consumo	18.87 %	1.88	0	1.95
Precio (Q)	2.30 %	0.00	1.39	1.74
Precisión	39.17 %	2.00	1.50	0.00
Tamaño	8.36 %	0.87	1.72	2.00
Torque	8.36 %	0.47	2.00	0.44
Velocidad	18.87 %	0.22	1.76	0.00
<b>Total ponderado</b>	100 %	1.32	1.48	0.70

(c) Motores: Nema 17 (23mm), Nema 17 (G=40mm), TT Motor.

Criterio	Peso	Nema 17	Nema 17 G	TT
Complejidad (control)	4.08 %	0.50	0.50	2.00
Consumo	18.87 %	1.87	1.89	2.00
Precio (Q)	2.30 %	1.14	0.89	2.00
Precisión	39.17 %	2.00	2.00	0.00
Tamaño	8.36 %	1.60	1.27	1.68
Torque	8.36 %	0.30	0.57	0.15
Velocidad	18.87 %	0.86	0.86	0.34
<b>Total ponderado</b>	100 %	1.52	1.51	0.72

Nota. Elaboración propia.

**Cuadro 13.** Análisis comparativo de los modelos de motores

(a) DM542, DRV8825 y L298N.

Criterio	Peso	DM542	DRV8825	L298N
Disipación	9.78 %	1.50	1.00	0.50
Posición	28.77 %	1.50	1.00	0.00
Precio (Q)	5.92 %	0.00	2.00	1.84
Protección	9.78 %	2.00	0.50	0.00
Tamaño	16.98 %	0.74	2.00	1.53
Velocidad	28.77 %	1.57	2.00	0.70
Total ponderado	100 %	1.35	1.47	0.62

(b) TB6600 y TMC2209/8.

Criterio	Peso	TB6600	TMC2209/8
Disipación	9.78 %	1.50	2.00
Posición	28.77 %	1.00	1.00
Precio (Q)	5.92 %	0.61	1.43
Protección	9.78 %	2.00	1.50
Tamaño	16.98 %	0.00	2.00
Velocidad	28.77 %	0.00	2.00
Total ponderado	100 %	0.67	1.63

Nota. Elaboración propia.

## 13.2. Tamaño de las pistas PCB basado en el estándar IPC-2221A

En esta sección se detalla el cálculo del grosor de pistas de las placas PCB realizadas para la plataforma móvil. Como se explicó en la Sección 8.1, el tamaño de las pistas se calculó usando el estándar IPC-2221. De la misma manera, se usó la calculadora en línea desarrollada por De Smith [43] para el cálculo del grosor mínimo de las pistas que se encuentran en el Cuadro 14.

**Cuadro 14.** Grosor de las pistas

Amperaje (A)	Grosor de pistas (mm)
1.1	0.35
8	5.42
9.1	6.48

Nota. Elaboración propia.

La ecuación que utiliza la calculadora en línea, y la establecida por el estándar IPC-2221 está descrita en (17) donde  $I$  es la corriente  $\Delta T$ : subida de temperatura (se usó  $10^{\circ}\text{C}$  para seguridad) y para pistas externas:  $k = 0.048$ ,  $b = 0.44$ ,  $c = 0.725$ . Luego, el ancho de la pista se calcula usando (18), tomando en cuenta que para cobre de  $1\text{ oz/ft}^2$ , el espesor es

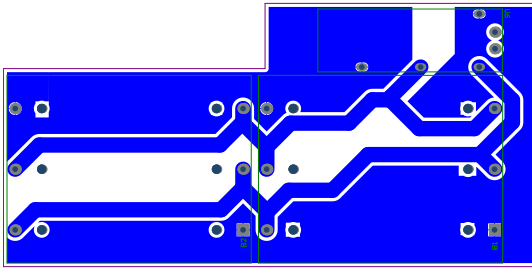
aproximadamente 1.4 mils.

$$\text{Área transversal (mils}^2) = \left[ \frac{I}{k \times \Delta T^b} \right]^{1/c} \quad (17)$$

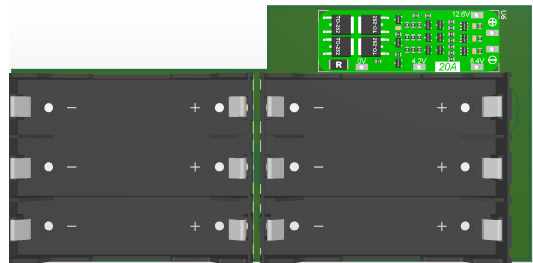
$$\text{Ancho (mils)} = \frac{\text{Área transversal (mils}^2)}{\text{Espesor del cobre (mils)}} \quad (18)$$

**Figura 25.** Placa del circuito de alimentación

(a) Vista del diseño de la placa.



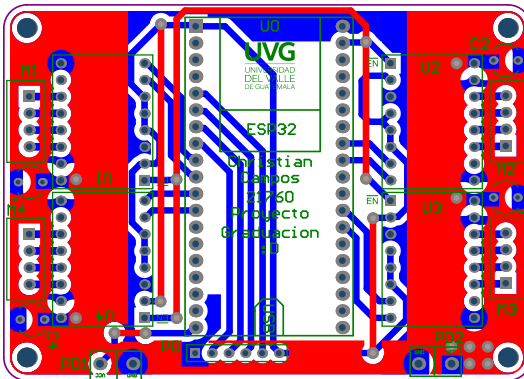
(b) Vista en 3D de la placa.



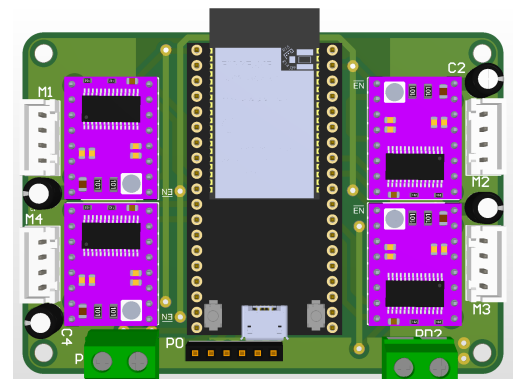
Nota. Elaboración propia.

**Figura 26.** Placa para el circuito de control

(a) Vista del diseño de la placa.



(b) Vista en 3D de la placa.



Nota. Elaboración propia.

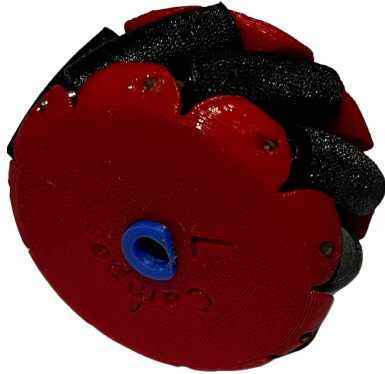
### 13.3. Llantas omnidireccionales impresas en 3D

En esta sección se encuentra la Figura 27, donde se puede visualizar las llantas que se diseñaron e imprimieron en 3D. Sin embargo, dichas llantas omnidireccionales no podían

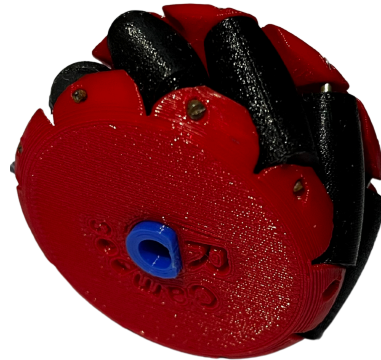
realizar los movimientos omnidireccionales requeridos, por lo mismo se optó por adquirirlas localmente.

**Figura 27.** Impresión 3D de las llantas

(a) Rueda izquierda.



(b) Rueda derecha.



Nota. La figura muestra las llantas impresas en 3D. Se utilizó PLA para la estructura y TPU para los rodillos. Elaboración propia.

## 13.4. Registros y configuración del AS5600

El Cuadro 15 muestra los registros y direcciones para comunicación I2C del AS5600. Para grabar la configuración de posición (ZPOS y MPOS) en la memoria, debe enviarse el valor `0x80` al registro `BURN` (comando `BURN_Angle`). Esto solo puede realizarse hasta tres veces, lo cual se refleja en el valor del registro `ZMCO`. Para grabar los ajustes de configuración (`CONF` y otros), se debe enviar `0x40` al mismo registro (`BURN_Setting`); esta operación solo puede hacerse una vez y únicamente si no se ha realizado ningún `BURN_ANGLE` previamente.

**Cuadro 15.** Registros importantes del AS5600

Nombre	Dirección	Tipo (R/W)	Bits	Descripción
<b>ZMCO</b>	0x00	R	1:0	Número de veces que se ha realizado <b>BURN_ANGLE</b> (máx. 3)
<b>ZPOS</b>	0x01-0x02	R/W	11:0	Posición cero configurada ( <i>offset</i> del ángulo)
<b>MANG</b>	0x05-0x06	R/W	11:0	Rango máximo del ángulo de medición
<b>CONF</b>	0x07-0x08	R/W/P	WD (bit 7)	Activación del <i>Watchdog</i>
			FTH (6:4)	Límite del filtro rápido
			SF (3:2)	Filtro lento
			PWMF (1:0)	Frecuencia PWM
			OUTS (3:2)	Configuración de la salida
			HYST (1:0)	Histéresis
			PM (1:0)	Modo de alimentación
<b>STATUS</b>	0x0B	R	MD (bit 5)	Imán detectado
			ML (bit 4)	Imán muy lejos
			MH (bit 3)	Imán muy cerca
<b>RAW_ANGLE</b>	0x0C-0x0D	R	11:0	Ángulo sin procesar (valor bruto)
<b>ANGLE</b>	0x0E-0x0F	R	11:0	Ángulo corregido según posición cero
<b>BURN</b>	0xFF	W	7:0	Registro especial para grabar configuración en memoria OTP

Nota. Datos obtenidos de [14]. Elaboración propia.

### 13.5. Estadísticas de las trayectorias

Para el análisis de las trayectorias realizadas en comparación con las deseadas se realizaron dos cálculos estadísticos principales. El primero fue el **error máximo**, que corresponde al valor más alto del error puntual entre la posición deseada y la posición real en cada instante de tiempo. Este parámetro permite identificar si el sistema mantiene sus desviaciones dentro de un rango aceptable; un valor excesivamente alto indicaría que la trayectoria se deforma significativamente, perdiendo su forma curva y aproximándose más a un desplazamiento lineal.

Por otra parte, se calculó el error de raíz cuadrada media (*Root Mean Square Error*,

RMSE), el cual proporciona una medida global del desempeño de seguimiento de trayectoria. Este parámetro indica la dispersión promedio del error en torno al valor deseado, penalizando más las desviaciones grandes y ofreciendo una idea clara de la precisión general del sistema de control. Su uso es ampliamente aceptado en la evaluación de modelos y sistemas dinámicos debido a su sensibilidad a los errores grandes y a que refleja de forma más completa la magnitud promedio del error cuadrático [44].

La ecuación empleada para el cálculo del RMSE se define como:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (e_i)^2} \quad (19)$$

donde  $e_i$  representa el error instantáneo entre la posición deseada y la posición real en el instante  $i$ , y  $N$  es el número total de muestras consideradas. Un valor bajo de RMSE implica que el sistema reproduce la trayectoria de manera precisa y consistente, mientras que un valor alto indica una mayor desviación promedio entre ambas trayectorias.

## 13.6. Repositorio de GitHub

En esta sección se encuentra el código desarrollado, tanto para el ESP32 como para el control mediante Python y MATLAB. Dicho código se encuentra disponible en el siguiente [repositorio en GitHub](#).

## 13.7. Vídeos

En esta sección se incluye el enlace a los vídeos demostrativos y de ensamble de la plataforma móvil. Todos estos vídeos se pueden visualizar en el siguiente [enlace](#).

## 13.8. Planos y manuales de la plataforma móvil

En esta sección se incluye el enlace a los planos de la plataforma móvil, junto a las recomendaciones para su impresión 3D, el manual de usuario y el manual de ensamblado de la plataforma móvil. Todos estos documentos se pueden visualizar en el siguiente [enlace](#).

