



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN MECATRÓNICA

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN MECATRÓNICA**

TEMA

**“ROBOT MÓVIL DE ALTO TORQUE: DESARROLLO
ELECTRÓNICO”**

LÍNEA DE INVESTIGACIÓN: Innovación y transferencia tecnológica / Automatización

AUTOR: Edwin Sebastián Mediavilla Borja

DIRECTOR: PhD. Carlos Xavier Rosero Chandi

Ibarra – Ecuador

2025

UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA

IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	100456332-4		
APELLIDOS Y NOMBRES:	Mediavilla Borja Edwin Sebastián		
DIRECCIÓN:	Imbabura, Otavalo		
EMAIL:	esmediavillab@utn.edu.ec		
TELÉFONO FIJO:	2-520-703	TELÉFONO MÓVIL:	0996666162

DATOS DE LA OBRA	
TÍTULO:	“ROBOT MÓVIL DE ALTO TORQUE: DESARROLLO ELECTRÓNICO ”
AUTOR:	Mediavilla Borja Edwin Sebastián
FECHA:	27/02/2025
SOLO PARA TRABAJOS DE TITULACIÓN	
PROGRAMA:	☒ PREGRADO ☐ POSGRADO
TÍTULO POR EL QUE OPTA:	Ingeniero en Mecatrónica
DIRECTOR:	PhD. Carlos Xavier Rosero Chandi



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN MECATRÓNICA

AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Edwin Sebastián Mediavilla Borja, con cédula de identidad Nro. 1004563324, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de integración curricular descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

Ibarra, a los 27 días del mes de febrero de 2025.

EL AUTOR

Firma:

Nombre: Edwin Sebastián Mediavilla Borja



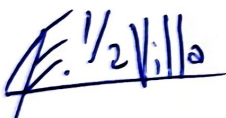
UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN MECATRÓNICA

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de esta y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 27 días del mes de febrero de 2025.

EL AUTOR

Firma: 

Nombre: Edwin Sebastián Mediavilla Borja



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN MECATRÓNICA

**CERTIFICACIÓN DIRECTOR DEL TRABAJO DE INTEGRACIÓN
CURRICULAR**

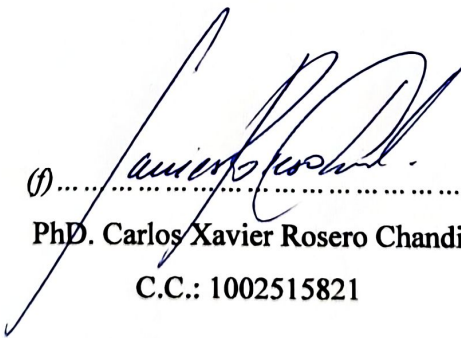
Ibarra, 26 de febrero de 2025

PhD. Carlos Xavier Rosero Chandi

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de titulación, el mismo que se ajusta a las normas vigentes de la Unidad Académica de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

(f).....

PhD. Carlos Xavier Rosero Chandi

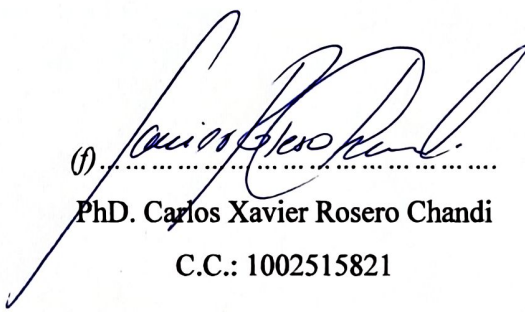
C.C.: 1002515821



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN MECATRÓNICA

APROBACIÓN DEL COMITÉ CALIFICADOR

El Tribunal Examinador del trabajo de titulación “ROBOT MÓVIL DE ALTO TORQUE: DESARROLLO ELECTRÓNICO” elaborado por EDWIN SEBASTIÁN MEDIAVILLA BORJA, previo a la obtención del título de INGENIERO EN MECATRÓNICA, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:


.....

PhD. Carlos Xavier Rosero Chandi

C.C.: 1002515821


.....

MSc. Luz María Tobar Subia Contento

C.C.: 1002444204

UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN MECATRÓNICA

AGRADECIMIENTO

A la Universidad Técnica del Norte, institución que ha sido el pilar fundamental en mi formación académica y profesional. A través de su compromiso con la educación y la investigación, me ha brindado las herramientas necesarias para alcanzar este importante logro.

A mi director, PhD. Xavier Rosero, quien no solo ha sido una guía fundamental en la culminación de este proyecto, sino también un docente ejemplar a lo largo de mi carrera. Sus enseñanzas han trascendido las bases teóricas, inculcándome el valor de la proactividad, el compromiso con el bienestar de los demás y la importancia de cuestionar y actuar frente a las injusticias. Su orientación ha sido clave en mi crecimiento tanto académico como personal.

A mi asesora, MSc. Luz María Tobar, por su invaluable respaldo. Su acompañamiento ha sido fundamental para el desarrollo y estructuración de este trabajo.

A los docentes de la carrera de Mecatrónica, quienes, con su experiencia, dedicación y compromiso con la enseñanza, me han proporcionado los conocimientos y herramientas necesarias para llevar a cabo esta investigación. Gracias a su labor, he podido desarrollar habilidades técnicas y analíticas que han sido esenciales en mi formación profesional.

Edwin Sebastián Mediavilla Borja

UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN MECATRÓNICA

DEDICATORIA

Este importante logro, que representa un peldaño más en la construcción de mis sueños, quiero dedicarlo a mi madre Nancy Borja y mi padre Edwin Mediavilla, las personas que más admiro en este mundo. Su amor incondicional, esfuerzo y dedicación han sido la base sobre la cual he construido cada uno de mis pasos en la vida. Gracias por enseñarme con su ejemplo que la perseverancia y el compromiso son la clave para alcanzar cualquier meta. Sin su apoyo inquebrantable, esta etapa no habría sido posible. Espero que este logro sea solo un reflejo de todo lo que han sembrado en mí y el inicio de muchos más que podamos compartir juntos.

A mis hermanos, Joseph H., Johan M. y Emiliano M., quienes han sido mi fuerza y mi compañía en cada paso del camino y han estado a mi lado desde siempre. Su apoyo y presencia han sido un pilar fundamental, recordándome siempre la importancia de la familia y el valor de seguir adelante sin importar los desafíos.

A mis mejores amigos, David De la C., Ismael M., Hillary U., Joseph P., Maiky G. y Kevin L., quienes han compartido cada momento importante de mi vida, incluidas las victorias y los tropiezos. Su amistad ha sido un refugio y una fuente de motivación constante, y saber que cuento con ustedes en cada paso hace que todo valga la pena. Más que amigos, son mis hermanos, una familia que la vida me regaló y con quienes he construido recuerdos inolvidables.

A mis compañeros de universidad, Andrew G., Pedro H. y Julián G., más que un equipo, una verdadera hermandad. Sin las interminables noches de proyectos, las risas entre el cansancio y el compromiso mutuo, este camino habría sido mucho más difícil. Esta experiencia universitaria no habría sido la misma sin ustedes, y les agradezco por demostrar que juntos podemos superar cualquier obstáculo.

Edwin Sebastián Mediavilla Borja

Resumen

El presente documento aborda el desarrollo de un sistema electrónico para un robot móvil de alto torque, centrado en garantizar la alimentación, el control y la comunicación necesarios para un funcionamiento eficiente. Se integraron componentes clave como motores de alto torque, un banco de baterías, controladores basados en Arduino Mega y una Raspberry Pi 5 con ROS 2 para la gestión de datos. El proyecto se lleva a cabo en colaboración con otras áreas de diseño mecánico y visión artificial, cada una descrita en documentos específicos. Los resultados obtenidos evidencian que la solución propuesta es viable para su implementación en diferentes entornos, brindando una plataforma accesible y de fácil mantenimiento. El robot demostró un desempeño adecuado en términos de autonomía, respuesta a comandos y adaptabilidad a diferentes condiciones de carga. Como conclusión, el desarrollo de este sistema electrónico ofrece una contribución significativa a la formación de ingenieros en mecatrónica, al proporcionar una herramienta práctica sin las restricciones económicas y técnicas de los sistemas comerciales.

Palabras clave: Robot Móvil, Alto Torque, Arduino, Raspberry Pi, ROS 2.

Abstract

This paper addresses the development of an electronic system for a high-torque mobile robot, focused on ensuring the power, control and communication required for efficient operation. Key components such as high torque motors, a battery bank, Arduino Mega based controllers and a Raspberry Pi 5 with ROS 2 for data management were integrated. The project is carried out in collaboration with other areas of mechanical design and machine vision, each described in specific documents. The results obtained show that the proposed solution is viable for implementation in different environments, providing an accessible and easy to maintain platform. The robot demonstrated adequate performance in terms of autonomy, response to commands and adaptability to different load conditions. In conclusion, the development of this electronic system offers a significant contribution to the training of mechatronics engineers by providing a practical tool without the economic and technical restrictions of commercial systems.

Keywords: Mobile Robot, High Torque, Arduino, Raspberry Pi, ROS 2.

Índice general

Capítulo I Introducción.....	1
1.1 Planteamiento del problema	1
1.2 Objetivos.....	2
1.2.1 General.....	2
1.2.2 Específicos.....	2
1.3 Alcance	3
1.4 Justificación	3
Capítulo II Revisión literaria	4
2.1 Antecedentes.....	4
2.2 Bases teóricas	7
2.2.1 Robótica móvil	7
2.2.2 Arquitectura del sistema electrónico	7
2.2.2.1 Motores DC	7
2.2.2.2 Baterías de plomo-ácido	10
2.2.2.3 Raspberry Pi 5	10
2.2.2.4 Arduino Mega.....	11
2.2.2.5 ROS 2	12
Capítulo III Marco metodológico	14
3.1 Enfoque de la investigación.....	14
3.2 Diseño de la investigación.....	14
3.2.1 Fase 1: Diseño del sistema electrónico base para un robot que integre los componentes clave.....	15
3.2.2 Fase 2: Implementación del sistema electrónico diseñado	16
3.2.3 Fase 3: Validación del sistema electrónico implementado para el robot	16

Capítulo IV Desarrollo y análisis	18
4.1 Especificaciones del sistema electrónico a diseñar	18
4.1.1 Requerimientos funcionales	18
4.1.2 Requerimientos de seguridad.....	18
4.1.3 Requerimientos de interfaz de usuario	19
4.2 Diseño del sistema electrónico	19
4.2.1 Plataforma robótica por emplear	19
4.2.2 Motores.....	20
4.2.2.1 Descripción del motor Docyke S550.....	20
4.2.2.2 Modo de operación del motor.....	21
4.2.2.3 Control de señales PWM	22
4.2.2.4 Análisis del torque del motor.....	22
4.2.3 Sistema de adquisición de datos para la medición de velocidad	24
4.2.4 Alimentación	25
4.2.4.1 Esquema y conexión serie-paralelo	25
4.2.4.2 Consumo y autonomía estimada.....	26
4.2.4.3 Alimentación independiente para la Raspberry Pi 5	27
4.2.5 Tipos de cable	29
4.3 Programación y control del robot	30
4.3.1 Lógica de control y flujo del programa en Arduino	30
4.3.1.1 Conectividad Bluetooth.....	31
4.3.1.2 Pines físicos	32
4.3.2 Aplicación móvil	32
4.3.2.1 Interfaz gráfica.....	33
4.3.2.2 Diagrama de bloques	34

4.3.3	Instalación del sistema operativo para la Raspberry Pi 5	35
4.3.3.1	Comunicación entre Arduino y la Raspberry Pi 5.....	35
4.3.4	Montaje de ROS 2	35
4.3.4.1	Creación del nodo medidor de velocidades.....	36
4.3.4.2	Diagrama de flujo del nodo	37
4.3.4.3	Visualización de parámetros.....	38
4.4	Ensamblaje del sistema.....	39
4.4.1	Montaje de elementos electrónicos.....	40
4.4.2	Cableado del sistema	41
4.4.3	Fabricación de PCB	42
4.4.4	Etiquetado	43
4.5	Pruebas de rendimiento	43
4.5.1	Pruebas de consumo de corriente en línea recta en superficie plana.....	45
4.5.2	Pruebas de consumo de corriente en rotación sobre el propio eje en superficie plana.....	46
4.5.3	Pruebas de consistencia de velocidad en línea recta en superficie plana	47
4.5.4	Pruebas de consistencia de velocidad en rotación sobre el propio eje en superficie plana.....	48
Capítulo V Conclusiones y recomendaciones		50
5.1	Conclusiones.....	50
5.2	Recomendaciones	50
Bibliografía.....		52
Anexos		56
Anexo A.....		56

Índice de figuras

Fig. 2.1 Estructura de un motor DC con escobillas [18].	8
Fig. 2.2 Estructura de un motor DC sin escobillas [18].	9
Fig. 2.3 Baterías de plomo-ácido de distintos tamaños y capacidades.	10
Fig. 2.4 Características de la Raspberry Pi5.	11
Fig. 2.5 Arduino Mega y sus características.	12
Fig. 2.6 Nodos de ROS.	13
Fig. 4.1 Estructura mecánica desarrollada en el trabajo de titulación “Robot móvil del alto torque: desarrollo mecánico”.	19
Fig. 4.2 Motor DOCYKE S550.	20
Fig. 4.3 Entradas de motor DOCYKE S550.	21
Fig. 4.4 Switch de selección del modo de operación.	21
Fig. 4.5 Switch de inversión de giro.	22
Fig. 4.6 Módulo FC-03 con rueda de 20 orificios.	24
Fig. 4.7 Diagrama de conexión del módulo FC-03 con el Arduino Mega.	24
Fig. 4.8 Batería de plomo-ácido de válvula regulada 12V9AH AGM.	25
Fig. 4.9 Banco de baterías. (a) Esquemático eléctrico. (b) Diagrama de cableado.	25
Fig. 4.10 Switch interruptor de 2 posiciones On/Off.	26
Fig. 4.11 Power bank marca FOCHWE de 30800 miliamperios hora.	27
Fig. 4.12 Jetson nano 2 GB developer kit diseñada por NVIDIA.	28
Fig. 4.13 Bornera de distribución de amperajes elevados.	29
Fig. 4.14 Diagrama de cableado motores y banco de baterías.	29
Fig. 4.15 Diagrama de flujo del programa desarrollado para Arduino Mega.	31
Fig. 4.16 Diagrama de cableado para la comunicación bluetooth.	31
Fig. 4.17 Diagrama de pines empleados del Arduino Mega.	32
Fig. 4.18 Entorno online de MIT App Inventor.	33
Fig. 4.19 Interfaz principal de control del robot.	33

Fig. 4.20	Menú de selección de dispositivos bluetooth.....	34
Fig. 4.21	Diagrama de bloques de la aplicación móvil.	34
Fig. 4.22	Conexión física entre Raspberry Pi 5 y Arduino Mega.	35
Fig. 4.23	Diagrama de instalación de ROS 2.	36
Fig. 4.24	Diagrama de creación de nodo de ROS2.	36
Fig. 4.25	Diagrama de flujo del nodo.....	37
Fig. 4.26	Diagrama de comandos para iniciar el nodo.	38
Fig. 4.27	Nodo activado y enviando datos.	38
Fig. 4.28	Gráfica generada por el comando <i>rqt</i> de las velocidades.....	39
Fig. 4.29	Estructura mecánica del robot móvil de alto torque.	39
Fig. 4.30	Acople del motor con encoder.	40
Fig. 4.31	Ubicación de baterías y power bank.	40
Fig. 4.32	Ubicación del microcontrolador y el minicomputador. (a) Arduino Mega. (b) Raspberry Pi 5.	41
Fig. 4.33	Cableado para alimentación de motores.	41
Fig. 4.34	Diseño de PCB en Fritzing.	42
Fig. 4.35	PCB manufacturada.	42
Fig. 4.36	Cables etiquetados.	43
Fig. 4.37	Conexión de periféricos a la microcomputadora.	43
Fig. 4.38	Conexión del multímetro en serie al banco de baterías.	44
Fig. 4.39	Robot soportando 185 kg adicionales.	44
Fig. 4.40	Gráfica carga (kg) vs corriente (A) en línea recta.....	45
Fig. 4.41	Gráfica carga (kg) vs corriente (A) en rotación sobre el eje.	46
Fig. 4.42	Gráfica de variación de velocidades con 185 kg de carga en línea recta.....	47
Fig. 4.43	Gráfica carga (kg) vs velocidades (m/s) en línea recta.	48
Fig. 4.44	Gráfica de variación de velocidades con 185 kg de carga giro en su propio eje. .	48
Fig. 4.45	Gráfica carga (kg) vs velocidades (m/s) rotación sobre el eje.	49

Índice de tablas

TABLA 1: Valores de PWM empleados para el movimiento de los motores.....	22
TABLA 2: Comparación de la fuerza aplicada y la requerida para el correcto movimiento del robot.....	23
TABLA 3: Consumo de corriente en línea recta.	45
TABLA 4: Consumo de corriente en rotación sobre el propio eje.	46
TABLA 5: Consistencia de velocidad en línea recta.	47
TABLA 6: Consistencia de velocidad en rotación sobre el propio eje.....	49

Capítulo I

Introducción

1.1 Planteamiento del problema

En el panorama educativo actual, el estudiante de ingeniería debe estar al tanto de todos los beneficios y retos que la robótica móvil ofrece, además de entender cómo esta se integra en la industria ecuatoriana. Esto plantea un desafío para la Universidad Técnica del Norte, ya que en el contexto de la Educación 4.0 se requiere aplicar enfoques distintos de los empleados hasta ahora en la enseñanza de los estudiantes, sobre todo en programas de ingeniería [1]. La falta de plataformas de educación acerca de esta rama de la robótica crearía una barrera que limitaría el aprendizaje, disminuyendo así las habilidades que el estudiante necesita poseer para ser competente en el ámbito laboral que le rodea.

En lo que va de siglo, se ha podido observar un continuo progreso en el ámbito de la robótica móvil. Este avance ha abarcado diversos logros importantes, desde la concepción de robots inspirados en la naturaleza, vehículos de exploración espacial y la continua investigación en vehículos autónomos. Todo esto ha desembocado en un claro incremento de las aplicaciones industriales que estos robots pueden desempeñar. Las innovaciones previamente mencionadas son fruto de las mejoras en procesadores, sensores, así como a la mejora en la eficiencia de las baterías utilizadas por los robots [2].

Hoy en día, la interacción del ser humano con la tecnología se da de una manera orgánica. En lo que respecta a la enseñanza de estas nuevas tecnologías, existe un fenómeno creciente de utilización de la robótica educativa para desarrollar las competencias de los estudiantes.

Todo esto contribuye al desarrollo de destrezas y técnicas de pensamiento lógico-matemático, mejora la creatividad y la comunicación [3].

La robótica educativa combina teoría y práctica, condensa conocimientos en informática, matemáticas y física. La robótica educativa combina dos aspectos fundamentales: el educativo y el tecnológico. La robótica, tanto como disciplina académica como área de aplicación práctica, engloba una amplia gama de conocimientos modernos en varias especialidades de ingeniería [4].

Un problema de los robots móviles que se encuentran actualmente en el mercado, capaces de manejar torques y cargas altas, es su excesivo precio. Esto hace que la experimentación y manipulación de estos vehículos sean costosos, limitando el uso de estos sistemas en el ámbito educativo y posteriormente en el industrial. Otro factor negativo para tomar en cuenta es que la mayoría de las plataformas en el mercado son de código cerrado, específicas de cada marca o utilizan un hardware y firmware difíciles de personalizar [5].

Dado lo anterior, se propone el desarrollo y la implementación del sistema electrónico para un robot móvil de alto torque, el cual usara hardware y firmware de libre acceso, dando así una herramienta de robótica educativa a la universidad, la cual a futuro puede ser manipulada y estudiada para comprender mejor todos los conceptos que conlleva la robótica de altos torques, y de esta manera preparar a los futuros ingenieros para los desafíos de la robótica industrial que se encuentren en el campo laboral.

1.2 Objetivos

1.2.1 General

Desarrollar el sistema electrónico de un robot móvil de alto torque.

1.2.2 Específicos

- Diseñar el sistema electrónico base para un robot que integre los componentes clave.
- Implementar el sistema electrónico diseñado.
- Validar el sistema electrónico implementado para un robot.

1.3 Alcance

El propósito de la investigación es el desarrollo del sistema electrónico para un robot móvil de alto torque. Se procederá al diseño de la electrónica, considerando cuidadosamente la integración de componentes esenciales como baterías, procesadores, sensores, módulos de comunicación, motores y controladores.

La siguiente fase se enfocará en la configuración y programación del sistema de control, garantizando una respuesta efectiva a los comandos. El proceso culminará con pruebas y validación del rendimiento del sistema.

1.4 Justificación

En el ámbito educativo y de investigación, el desarrollo del sistema electrónico para el robot permitirá a los nuevos estudiantes aprender de él y ver su funcionamiento, además de que en futuras investigaciones se pueda trabajar sobre esta plataforma sin la limitante de tener un software y hardware cerrados o con licencia de algún fabricante.

En el ámbito económico, se va a adquirir hardware de buena calidad pero que no sobrepase las limitantes económicas de los estudiantes, y así mantener un equilibrio calidad-precio, abaratando los costes de este tipo de robots, que importados pueden llegar a valer varios miles de dólares.

Finalmente es importante remarcar el ámbito social, ya que al ser esta una herramienta de futuro aprendizaje se está velando por la calidad de la educación en la universidad, y así preparar profesionales capaces con el conocimiento necesario para adaptarse al entorno industrial que los rodea.

Capítulo II

Revisión literaria

2.1 Antecedentes

El desarrollo de robots con elevadas capacidades de torque y resistencia se abordan de manera meticulosa y objetiva, sustentadas por diversos antecedentes que han explorado tanto la construcción como diseño de estos dispositivos. A continuación, se proporciona un detallado análisis de registros que han contribuido en el desarrollo de esta temática.

En [5] se desarrolla un robot móvil de código abierto basado en ROS (ROMR) de bajo costo. Se utiliza hardware disponible en el mercado, tecnologías de fabricación aditiva, perfiles de aluminio motores de corriente continua (DC) sin escobillas de alto torque. Este robot es compatible con el sistema operativo de robots (ROS), y posee una carga útil de 90kg. Además, en este robot se puede contextualizar algoritmos de localización y mapeo simultáneos (SLAM), un requisito previo y esencial para la navegación autónoma de robots. Se validó mediante simulaciones y la construcción física del modelo.

El trabajo presentado en [6] presenta una plataforma de robot móvil de bajo costo con un chasis fijo de cuatro ruedas controlado por una Raspberry Pi y Arduino Uno. El robot móvil es capaz de moverse, navegar y evitar obstáculos en un entorno 2D con capacidad de mapeo.

El proyecto de [7] se basa en utilizar MATLAB Simulink para el desarrollo y la prueba de algoritmos de control en una plataforma robótica. Los algoritmos se implementan en tiempo real en el sistema operativo del robot. Se emplean modelos en Simulink con herramientas como Embedded Coder para generar código y se prueban algoritmos PID

estándar y no estándar para controlar el ángulo de dirección del robot. La implementación y pruebas se realizan en un entorno real, mostrando éxito en las tareas propuestas.

En el trabajo [8] se plantea el desarrollo de un robot móvil autónomo de bajo coste con capacidades de control y dirección remota. Se usa tarjetas Arduino y Raspberry Pi. La comunicación con el robot se establece a través de Wi-Fi y Bluetooth. Se integra un módulo GPS (sistema de posicionamiento global) para el funcionamiento autónomo y un sensor ultrasónico con el propósito de evitar obstáculos y encontrar una nueva ruta. La batería es recargable de forma inalámbrica, lo que garantiza su autonomía durante el desplazamiento. Este robot tiene aplicaciones potenciales en manipulación de cargas, asistencia a pacientes, servicio de comidas y bebidas.

En [9] se presenta el diseño de un robot móvil omnidireccional orientado a la logística industrial en entornos dinámicos y con tráfico intenso. Se desarrolla un sistema de control con algoritmos de alto y bajo nivel, y se implementa una interfaz gráfica para cada sistema. Para el control preciso y robusto de los motores, se utiliza el microcontrolador myRIO como ordenador de bajo nivel. La Raspberry Pi 4 y un PC remoto se emplean para decisiones de alto nivel, como cartografía, planificación de trayectorias y localización, utilizando múltiples sensores Lidar, IMU y datos de odometría. En cuanto a programación, se emplea LabVIEW para el ordenador de bajo nivel y el Sistema Operativo de Robots (ROS) para la arquitectura de software de alto nivel.

El estudio [10] se enfoca en la mejora de un robot móvil mediante la implementación de navegación omnidireccional capaz de seguir una línea y detectar obstáculos. Se usa el microcontrolador Arduino como intermediario para conectar sensores, motores y ruedas. El cuerpo del robot es de forma cuadrada, tiene 4 motores de corriente continua, 8 sensores IR y 2 sensores ultrasónicos, así como 2 controladores para los motores, un Arduino Mega 2560, 4 ruedas Mecanum y una batería de 12V de corriente continua. Se evalúa el rendimiento de giro de las ruedas Mecanum. La simulación demuestra el correcto seguimiento de la línea y la evasión exitosa de obstáculos.

En [11] se describe un diseño de robot omnidireccional que utiliza ruedas de goma normales para lograr un movimiento omnidireccional. El enfoque modular del robot permite su uso en una amplia gama de aplicaciones, como robots de vigilancia, carretillas elevadoras

en almacenes, robots de construcción, robots de limpieza, etc. El artículo se centra en los aspectos de programación y control del robot, así como en el diseño CAD subyacente. El diseño propuesto tiene una ventaja sobre otros diseños, ya que ofrece una mayor movilidad, lo que lo hace adecuado para aplicaciones donde el tiempo de respuesta debe ser mínimo. Además, el artículo concluye con un estudio comparativo entre el robot omnidireccional y un robot de 2 ruedas motrices.

En [12] se propone un modelo de consumo de energía para el robot móvil omnidireccional de tres ruedas (TOMR) que tiene en cuenta la movilidad omnidireccional del robot. El modelo analiza y modela el consumo de energía y potencia del robot en diferentes direcciones, ya que la velocidad de cada rueda cambia drásticamente cuando el TOMR avanza en diferentes ángulos.

En [13] existe el desarrollo de un robot móvil para desplazamientos en entornos no estructurados, empleando visión artificial. El sistema consta de varias etapas como el diseño de hardware, software, la comunicación y la alimentación. Se incorpora componentes como Arduino Mega 2560, Raspberry PI3B, sensores ultrasónicos, Kinect y motores DC. El software se realizó en Python para el procesamiento y se emplea Arduino para el control de los motores.

La tesis planteada en [14] se centra en la creación de un robot móvil autónomo que usa algoritmos de visión artificial. Usa la Raspberry PI Pico para el procesamiento e integra el módulo OpenMV Cam H7 como visión. Mediante el uso de algoritmos el robot demuestra una respuesta correcta. Este robot proporciona un gran recurso educativo para estudiantes interesados en electrónica y automatización.

En [15] se presenta el desarrollo de un robot móvil tipo rover con seis ruedas. Se plantea una estrategia de control para el seguimiento de trayectorias, verificada mediante simulación numérica. El robot cuenta con sensores de posición, un módulo GPS y una unidad de medición inercial, los datos obtenidos se procesan en una Raspberry Pi 4 y son utilizados para el correcto desplazamiento del robot. Se llevan a cabo pruebas de campo en lazo abierto y en lazo cerrado.

2.2 Bases teóricas

2.2.1 Robótica móvil

La robótica móvil, se centra en el diseño, construcción y operación de robots capaces de desplazarse en entornos cambiantes. Estos sistemas no solo ejecutan movimientos en ambientes no estructurados, sino que también integran sensores para recopilar información detallada sobre su entorno, proporcionando así una representación precisa de su estado actual [16]. Esta disciplina ha experimentado un notable crecimiento gracias a los avances en mecánica, sensores y control, estableciendo la base para una amplia variedad de aplicaciones.

Robots equipados con sistemas autónomos de mapeo y planificación de rutas son utilizados en almacenes automatizados, optimizando la gestión de inventarios y reduciendo los tiempos de espera [17]. Este enfoque pragmático destaca la influencia directa de la robótica móvil en la transformación de procesos logísticos y productivos.

2.2.2 Arquitectura del sistema electrónico

2.2.2.1 Motores DC

Los motores de corriente continua, comúnmente llamados motores DC, convierten la energía eléctrica en energía mecánica a través de interacciones electromagnéticas. Su diseño clásico incluye un estator que puede tener un imán permanente, y un rotor que actúa como electroimán. Ambos componentes están conectados por un conmutador que invierte periódicamente la polaridad, permitiendo así un giro continuo. Una de las principales características de estos motores es la facilidad con la que se puede regular tanto la velocidad como el par, lo que los convierte en una opción ideal para aplicaciones industriales y sistemas que requieren un control preciso de las revoluciones [18].

Motores DC con escobillas

La **Fig. 2.1** señala las partes de un motor DC con escobillas. En los motores de corriente continua con escobillas, la conmutación se lleva a cabo de manera mecánica a través de un conjunto de cepillos y un conmutador, que se encargan de invertir la corriente en el rotor para asegurar un giro estable. Según la forma en que se conecten las bobinas del estator y el rotor, pueden presentarse configuraciones en serie, en paralelo o compuestas, cada una de las cuales

ofrece características de rendimiento específicas. Una de las limitaciones más relevantes de estos motores es el desgaste de los cepillos, así como la generación de chispas, factores que pueden reducir su vida útil y limitar su uso en entornos de alto vacío o en aplicaciones con estrictos requisitos de seguridad [18].



Fig. 2.1 Estructura de un motor DC con escobillas [18].

Motores DC sin escobillas

En los motores de corriente continua con escobillas, la conmutación se lleva a cabo de manera mecánica a través de un conjunto de cepillos y un conmutador, que se encargan de invertir la corriente en el rotor para asegurar un giro estable. La **Fig. 2.2** muestra un motor DC sin escobillas, según la forma en que se conecten las bobinas del estator y el rotor, pueden presentarse configuraciones en serie, en paralelo o compuestas, cada una de las cuales ofrece características de rendimiento específicas. Una de las limitaciones más relevantes de estos motores es el desgaste de los cepillos, así como la generación de chispas, factores que pueden reducir su vida útil y limitar su uso en entornos de alto vacío o en aplicaciones con estrictos requisitos de seguridad [18].

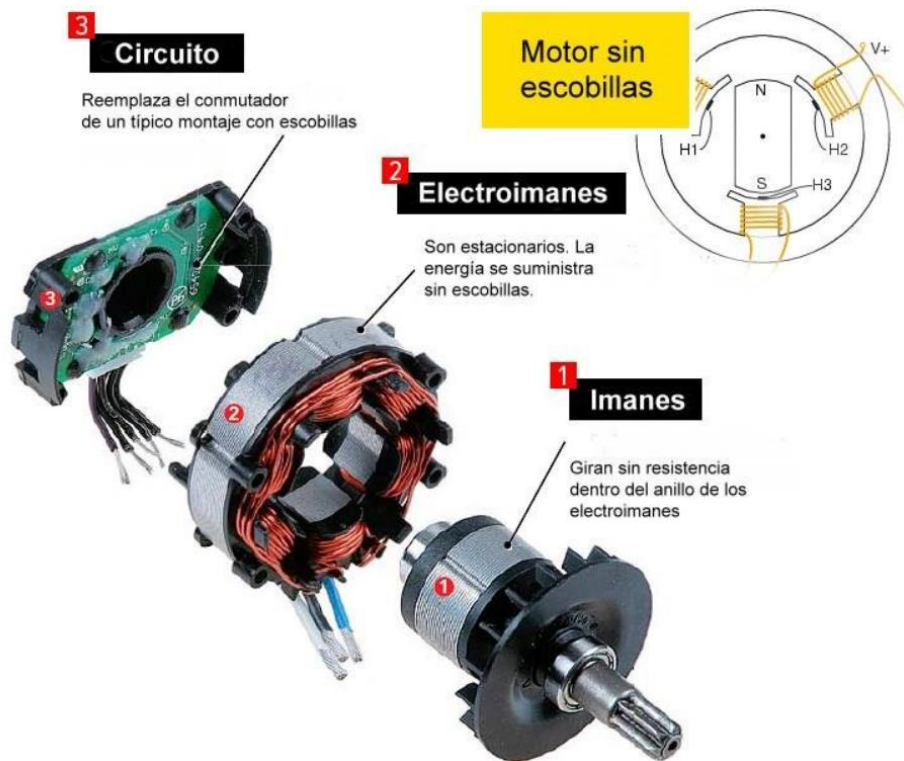


Fig. 2.2 Estructura de un motor DC sin escobillas [18].

Control de motores DC mediante PWM

Para controlar la velocidad de un motor DC, se utiliza la técnica conocida como Modulación por Ancho de Pulso (PWM, por su denominación en inglés). Esta técnica se basa en la generación de una señal periódica, cuyo ciclo de trabajo se ajusta según la potencia o velocidad deseada. En este método, la señal se mantiene en un estado alto de 5V, durante un periodo específico y en un estado bajo el resto del tiempo, lo que permite que la tensión promedio que recibe el motor varíe en función de dicho ciclo de trabajo [19].

Cuando el ciclo de trabajo es mayor, el motor recibe más energía y, por lo tanto, gira a mayor velocidad; en cambio, un ciclo de trabajo más bajo provoca una reducción en su velocidad. Dado que la mayoría de los controladores y microcontroladores solo ofrecen salidas digitales (encendido/apagado), la modulación PWM se convierte en una herramienta esencial para simular señales analógicas y, así, regular de manera continua la velocidad de los motores o la potencia proporcionada a una carga [19].

2.2.2.2 Baterías de plomo-ácido

Las baterías de plomo-ácido como las de la **Fig. 2.3** son dispositivos electroquímicos que emplean plomo y ácido sulfúrico para almacenar energía eléctrica. Su funcionamiento se basa en un electrodo positivo de dióxido de plomo y otro negativo de plomo duro, ambos sumergidos en ácido sulfúrico. Este entorno propicia reacciones químicas que producen sulfato de plomo, así como iones de oxígeno e hidrógeno, permitiendo la conversión de energía química en eléctrica. Existen dos tipos principales de estas baterías: las VLA (ventiladas o de electrolito inundado) y las VRLA (selladas o reguladas por válvula); estas últimas son las más comunes debido a su facilidad de uso. Aunque las baterías de plomo-ácido ofrecen ventajas como un bajo costo de fabricación, amplia disponibilidad en el mercado y facilidad de reciclaje, también presentan algunas limitaciones, como una menor densidad de energía, una vida útil más corta y la necesidad de mantenimiento. Dependiendo del tipo, su duración puede variar desde 5 a 7 años en aplicaciones automotrices, hasta aproximadamente 30 años en sistemas estacionarios [20].



Fig. 2.3 Baterías de plomo-ácido de distintos tamaños y capacidades.

2.2.2.3 Raspberry Pi 5

La Raspberry Pi 5, tal como se muestra en la **Fig. 2.4**, es una microcomputadora de última generación que cuenta con un potente procesador de cuatro núcleos a 2.4 GHz, respaldada por una GPU VideoCore VII y hasta 8 GB de memoria LPDDR4X. Su diseño incluye una gama de interfaces, como puertos USB 3.0, conectividad Wi-Fi de doble banda, Bluetooth,

salidas micro-HDMI y un encabezado GPIO de 40 pines. Estas características hacen posible el desarrollo de proyectos de diversas complejidades, tanto en el ámbito educativo como en el profesional, ofreciendo un entorno versátil para aplicaciones en robótica [21].

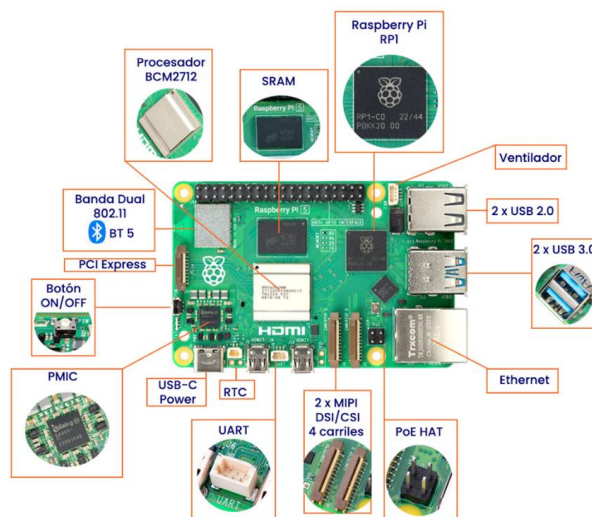


Fig. 2.4 Características de la Raspberry Pi5.

2.2.2.4 Arduino Mega

El Arduino Mega 2560, como se muestra en la **Fig. 2.5**, es una placa de microcontrolador que se basa en el ATmega2560. Esta potente tarjeta está equipada con 54 pines de entradas/salidas digitales, de los cuales 15 pueden configurarse como salidas PWM, además de contar con 16 entradas analógicas y 4 puertos seriales (UART). Incluye un cristal oscilador de 16 MHz, conexión USB, un jack de alimentación, un conector ICSP y un botón de reinicio. Todo esto le permite al microcontrolador operar de manera autónoma; simplemente se necesita conectarlo a un ordenador mediante un cable USB o suministrarle energía a través de un adaptador de corriente o batería [22].

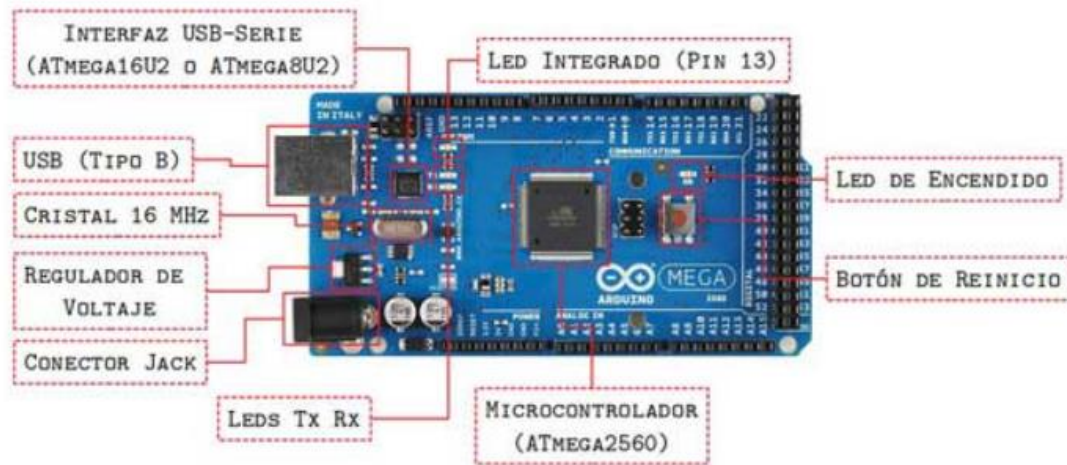


Fig. 2.5 Arduino Mega y sus características.

2.2.2.5 ROS 2

ROS 2 (Robot Operating System 2) representa la evolución del proyecto original de ROS, que comenzó su desarrollo en 2007 con el objetivo de proporcionar herramientas de software para distintos tipos de robots. La principal motivación detrás de esta nueva versión es la necesidad de abordar un rango de aplicaciones más amplio, incluyendo la colaboración entre múltiples robots, el uso en sistemas con recursos de hardware limitados, así como requisitos de tiempo real y redes poco ideales. Además, ROS 2 está diseñado para facilitar la transición de prototipos de laboratorio a entornos de producción, ofreciendo patrones más claros para la construcción y configuración de sistemas [23].

ROS 2 no se presenta como un sistema operativo convencional, sino como un conjunto de bibliotecas y herramientas que operan en conjunto para facilitar la creación de sistemas robóticos complejos. Ofrece diversas características, entre las que se incluyen la comunicación entre nodos, la definición y el intercambio de mensajes, la gestión de paquetes y la introspección del sistema. Todo esto se desarrolla bajo una filosofía modular que permite la integración de diferentes componentes de hardware y software. Gracias a esta arquitectura, ROS 2 fomenta la colaboración entre múltiples desarrolladores y la reutilización del código, estableciendo interfaces estandarizadas que facilitan la conexión entre sensores, actuadores y algoritmos avanzados [24].

Nodos

Un sistema ROS está compuesto por diversos programas que operan de manera simultánea y se comunican entre ellos, conocidos como nodos. Cada uno de estos nodos actúa como un proceso autónomo que utiliza las bibliotecas de ROS, integrándose con el resto de la arquitectura para llevar a cabo tareas específicas. Estas tareas pueden incluir, por ejemplo, la lectura de datos de un sensor, el envío de señales de control a un motor o el procesamiento de información proveniente de un joystick. Al distribuir estos nodos en una o varias computadoras dentro de la misma red, se logra una notable flexibilidad en el diseño de sistemas robóticos, los cuales pueden funcionar de forma centralizada o de manera distribuida [24]. La **Fig. 2.6** ilustra un sistema ROS compuesto por varios nodos interconectados en una red.

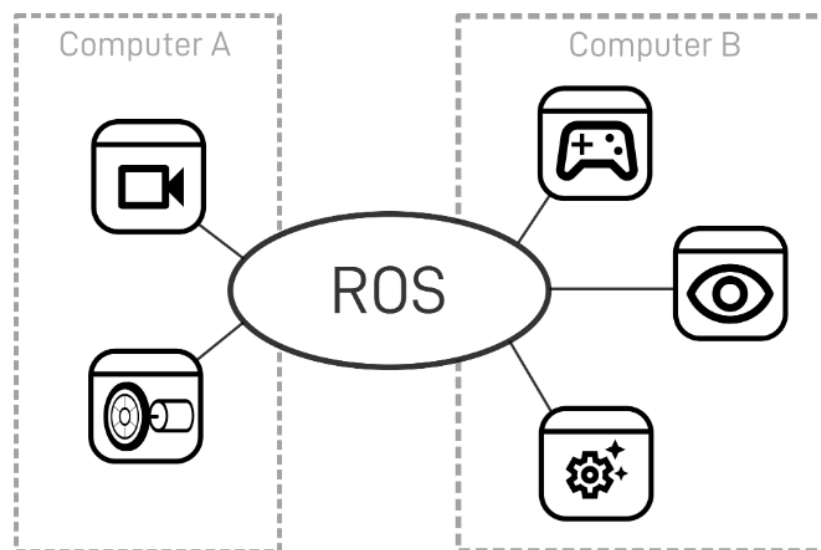


Fig. 2.6 Nodos de ROS.

Tópicos y mensajes

En ROS, los nodos intercambian información a través de tópicos y mensajes. Un tópico funciona como un canal, identificado por un nombre, al que uno o varios nodos pueden enviar mensajes, siendo estos nodos los publicadores. Por otro lado, los suscriptores son aquellos nodos que reciben estos mensajes al suscribirse al mismo tópico [24].

Capítulo III

Marco metodológico

3.1 Enfoque de la investigación

La presente investigación se basa, en primer lugar, en un enfoque experimental que se centra en la manipulación controlada de variables con el fin de evaluar sus efectos. Mediante ensayos en condiciones reguladas, se analiza cómo diversas variables independientes interactúan entre sí, lo que fortalece tanto la validez como la aplicabilidad práctica de los resultados obtenidos.

Complementariamente, se implementa una metodología documental, que implica la recopilación y el examen de información proveniente de fuentes secundarias. Esto permite identificar y organizar datos relevantes que enriquecen el proceso investigativo. De este modo, se establece una base sólida de conocimientos previos que respalda la propuesta y ayuda a resolver las incógnitas planteadas. La combinación de los enfoques experimental y documental, junto con la naturaleza aplicada de este trabajo, asegura un desarrollo integral y una validación robusta del sistema electrónico propuesto para el robot móvil de alto torque.

3.2 Diseño de la investigación

El desarrollo del sistema electrónico se estructura en tres fases metodológicas: diseño, implementación y validación, cada una con actividades específicas que permiten cumplir con los objetivos planteados. Estas actividades se registran y documentan para garantizar el cumplimiento de cada paso del proceso.

3.2.1 Fase 1: Diseño del sistema electrónico base para un robot que integre los componentes clave

Esta fase inicial se enfoca en investigar, seleccionar y estructurar los elementos esenciales que compondrán el sistema electrónico. Aquí se establecen los cimientos técnicos del proyecto.

Actividad 1.1: “Investigación de los componentes electrónicos necesarios, como sensores, actuadores y módulos de comunicación”

Esta actividad implica la recopilación de información técnica sobre sensores, actuadores, controladores y módulos de comunicación que puedan ser utilizados en el sistema electrónico.

Actividad 1.2: “Selección de los componentes electrónicos”

Tras la investigación, se procede a elegir los componentes que cumplen con los requisitos del sistema, priorizando aquellos con mejor relación costo-beneficio y mayor compatibilidad técnica.

Actividad 1.3: “Elaboración de los esquemáticos y diagramas de conexión del sistema electrónico”

Se diseñan los diagramas eléctricos y de conexión, asegurando que todos los elementos se integren correctamente en el sistema. Estos esquemáticos especifican las interconexiones entre la fuente de alimentación, la Raspberry Pi 5, el Arduino Mega y los motores, con énfasis en la correcta distribución de energía y señales.

Actividad 1.4: “Establecimiento de los requisitos de alimentación eléctrica”

Se determinan los requerimientos de alimentación, considerando dos sistemas semindependientes, uno para los motores y otro para los componentes electrónicos.

Actividad 1.5: “Elaboración del circuito de gestión de energía”

Se diseña el circuito encargado de regular y distribuir la energía a los diferentes componentes. Esto incluye la incorporación de elementos de protección para garantizar la seguridad y estabilidad del sistema.

3.2.2 Fase 2: Implementación del sistema electrónico diseñado

En esta etapa se lleva a cabo la materialización del diseño realizado en la fase anterior. Se construye y prueba el sistema electrónico en un entorno controlado para garantizar su funcionalidad básica.

Actividad 2.1: “Adquisición de los componentes electrónicos”

Se procede a la compra de los componentes seleccionados, verificando que cumplan con las especificaciones técnicas establecidas en la fase de diseño. Esto incluye la obtención de motores, baterías, controladores, sensores y demás módulos electrónicos.

Actividad 2.2: “Construcción del prototipo del sistema electrónico en una placa de pruebas”

Se ensamblan los componentes en una placa de pruebas para facilitar la conexión y modificación durante las pruebas iniciales. Este prototipo permite verificar el funcionamiento básico de cada elemento antes de integrarlos en el robot.

Actividad 2.3: “Pruebas preliminares para verificar la funcionalidad básica de cada componente”

Cada componente se prueba individualmente para garantizar su correcto funcionamiento. Estas pruebas incluyen la validación de los motores, el control de velocidad mediante PWM, y la lectura de datos de los encoders.

Actividad 2.4: “Integración y conexión de los módulos electrónicos de manera efectiva en el robot”

Una vez verificada la funcionalidad básica, se realiza la integración de los módulos en el robot. Esto implica conectar los motores, encoders, controladores y la Raspberry Pi 5 según el diseño inicial, asegurando que todos los elementos trabajen en conjunto.

3.2.3 Fase 3: Validación del sistema electrónico implementado para el robot

La última fase se centra en validar el funcionamiento del sistema electrónico en condiciones reales de operación. Se realizan pruebas exhaustivas.

Actividad 3.1: “Realización de pruebas de rendimiento del sistema electrónico en condiciones estáticas y dinámicas”

Se evalúa el rendimiento del sistema en diferentes escenarios. En pruebas estáticas, se verifica el correcto funcionamiento de cada módulo en un entorno controlado. En pruebas dinámicas, se mide el desempeño del robot en movimiento, evaluando parámetros como velocidad, estabilidad y respuesta a los comandos.

Actividad 3.2: “Verificación de la estabilidad y robustez del sistema”

Se realizan pruebas de estrés para evaluar la capacidad del sistema de operar bajo carga continua y en condiciones adversas. Esto incluye medir la resistencia del sistema a variaciones de voltaje y corriente, así como la durabilidad de los componentes.

Actividad 3.3: “Ajustes y correcciones”

Con base en los resultados de las pruebas, se realizan ajustes en el diseño o en la configuración del sistema. Estos cambios aseguran que el robot cumpla con los requisitos funcionales y técnicos establecidos.

Actividad 3.4: “Ensayo de funcionamiento del sistema final”

Se realiza un ensayo completo en el que el robot ejecuta las tareas asignadas de manera autónoma. Este ensayo evalúa la integración de todos los sistemas y mide el cumplimiento de los objetivos planteados.

Actividad 3.5: “Desarrollo del documento correspondiente al trabajo de integración curricular”

Finalmente, se documenta todo el proceso en un informe técnico que incluye los resultados de las pruebas, las especificaciones finales del sistema y los aprendizajes obtenidos durante el desarrollo del proyecto.

Capítulo IV

Desarrollo y análisis

4.1 Especificaciones del sistema electrónico a diseñar

El sistema electrónico del robot móvil de alto torque debe cumplir varios requisitos funcionales y técnicos para garantizar el funcionamiento adecuado en las condiciones de operación previstas. Estos requisitos se basan en la arquitectura mecánica previamente diseñada y las capacidades de control y autonomía previstas. Los requisitos funcionales definen cómo debe comportarse el sistema.

4.1.1 Requerimientos funcionales

Se definen los requisitos funcionales del sistema, los cuales establecen el comportamiento esperado del robot móvil en aspectos clave como movilidad, autonomía y control. En términos de movilidad, el robot es capaz de desplazarse hacia adelante y hacia atrás, así como de girar con precisión sobre su propio eje. En lo que respecta a la autonomía, el banco de baterías asegura un tiempo operativo continuo aceptable, incluso bajo carga. En cuanto al control, el robot recibe comandos de forma inalámbrica a través de bluetooth desde una aplicación móvil, respondiendo en tiempo real con una latencia mínima y procesando datos de RPM y velocidad lineal.

4.1.2 Requerimientos de seguridad

Para asegurar un funcionamiento seguro del robot, se definen los siguientes parámetros:

- Protección contra sobrecargas en el sistema eléctrico.

- Mecanismos de desconexión de emergencia en caso de fallos.
- Medidas de disipación térmica para los componentes de alto consumo.

4.1.3 Requerimientos de interfaz de usuario

La interacción del usuario con el sistema de una manera intuitiva y eficiente se garantiza mediante una aplicación móvil con una interfaz amigable con botones grandes de control de la dirección del robot y un rango de conectividad adecuado.

4.2 Diseño del sistema electrónico

El diseño del sistema electrónico integra los componentes necesarios para el movimiento, control y monitoreo del robot, considerando tanto la distribución de potencia como la comunicación de datos.

4.2.1 Plataforma robótica por emplear

Para la implementación del sistema electrónico se usa la estructura mecánica desarrollada en [25], la cual está diseñada y probada para los componentes seleccionados en este trabajo. El chasis esta dimensionado para que ningún elemento haga interferencia con otro y todos bien sujetos. Además de contar con pisos y divisiones dentro como se puede observar en la **Fig. 4.1** para que todos los elementos sean de sencillo acceso para su futuro mantenimiento, cambio o ampliación.



Fig. 4.1 Estructura mecánica desarrollada en el trabajo de titulación “Robot móvil del alto torque: desarrollo mecánico”.

4.2.2 Motores

El sistema de tracción para el robot móvil de alto torque está conformado por cuatro motores DOCYKE S550 (**Fig. 4.2**). Estos motores tienen características especiales que los hacen muy robustos y aptos para este tipo de aplicaciones. Los motores se conectan directamente al eje de cada una de las llantas, lo que permite una tracción diferencial de cuatro ruedas. El movimiento en distintas direcciones del robot se da gracias al uso de señales PWM (Pulse Width Modulation), que permiten variar la velocidad y el sentido de giro de cada motor por separado.



Fig. 4.2 Motor DOCYKE S550.

4.2.2.1 Descripción del motor Docyke S550

El motor DOCYKE S550 admite un rango de voltaje de 12V a 24V, con un voltaje nominal de 18V y una corriente nominal de 4A. Como parte del diseño del sistema, se ha elegido usar 24V para la alimentación de los motores, lo que permite un aumento significativo en la velocidad comparándola con la velocidad nominal que indica el fabricante, además cuenta con medidas de protección contra sobrecorrientes y altas temperaturas, las cuales al activarse bloquean al motor e impiden daños sobre componentes claves para su funcionamiento.

Cada motor cuenta con dos entradas bien diferenciadas como se aprecia en la **Fig. 4.3**, la primera entrada se denomina “POWER INPUT”, la cual es la entrada de alimentación principal y va a entregarle al motor el voltaje y la corriente necesaria para mover las distintas cargas, a este puerto se le insertan dos cables uno de +24V y otro cable de GND o negativo.

La segunda entrada se identifica como “SIGNAL INPUT”, este puerto es para controlar el motor y se le conectan tres cables, uno de señal, que permite la modulación mediante PWM, un cable de 5V, que alimenta el circuito interno y un cable GND o referencia a tierra.



Fig. 4.3 Entradas de motor DOCYKE S550.

4.2.2.2 Modo de operación del motor

Los motores cuentan con dos modos de operación que se acceden mediante el switch número 1 que se observa en la **Fig. 4.4**. El modo motor permite al motor girar a una velocidad regulada mediante los PWM, mientras que, el modo servo permite el posicionamiento del eje en un rango de 0° a 360°. Para el sistema en cuestión, el modo de operación seleccionado en todos los motores es el modo motor.

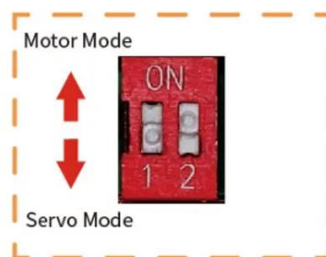


Fig. 4.4 Switch de selección del modo de operación.

El switch número 2 se observa en la **Fig. 4.5**, permite invertir el giro de los motores, esto simplifica el envío de señales PWM, ya que se pueden configurar los motores de tal manera que todos reciban la misma señal, pero que cada lado del robot gire en una dirección opuesta y esto permita la locomoción en todos los sentidos de manera adecuada.

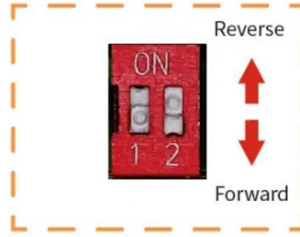


Fig. 4.5 Switch de inversión de giro.

4.2.2.3 Control de señales PWM

El control de la dirección y velocidad de los motores es mediante PWM, estos motores tienen la siguiente configuración

- PWM = 90, motor detenido
- PWM = 180, máxima velocidad en un sentido
- PWM = 0, máxima velocidad en sentido contrario

Usando esta configuración como base, en la **TABLA 1** se presentan los valores óptimos de PWM con los cuales trabaja la programación para mover el robot.

TABLA 1: Valores de PWM empleados para el movimiento de los motores.

PWM	Estado del motor
160	Velocidad máxima hacia adelante
100	Baja velocidad hacia adelante
90	Detenido
80	Baja velocidad hacia atrás
20	Velocidad máxima hacia atrás

4.2.2.4 Análisis del torque del motor

Los motores cuentan con un torque de 55 N·m cada uno y las llantas que se emplean tienen un diámetro de 20.32 cm. El cálculo de la fuerza tangencial generada en la superficie de contacto de la llanta con el suelo cuando el motor está trabajando a dicho torque está dado por

$$Fuerza\ tangencial\ en\ la\ llanta = \frac{Torque\ del\ motor}{radio\ de\ la\ llanta}, \quad (4.1)$$

$$Fuerza\ tangencial\ en\ la\ llanta = \frac{55\ N \cdot m}{0.1016\ m} = 541.4\ N.$$

Para comprobar si los motores son capaces de mover el robot en las distintas condiciones de carga que se le aplica, es necesario analizar la fuerza necesaria para mover dichos pesos.

La fuerza normal aplicada que se necesita para mover el peso del robot y su carga para cada motor está dada por.

$$Fuerza\ por\ llanta = \frac{masa\ total\ del\ robot * gravedad\ terrestre}{4}, \quad (4.2)$$

Así para el peso total del robot de 50 kg, se tiene

$$Fuerza\ por\ llanta = \frac{50 * 9.81}{4} = 122.6\ N.$$

Por otro lado, para un peso total del robot 50 kg más una carga de 100 kg, se tiene

$$Fuerza\ por\ llanta = \frac{150 * 9.81}{4} = 367.9\ N.$$

En la **TABLA 2** se compara estos valores con la fuerza que puede darnos cada motor y ver si nuestro sistema es capaz de mover su propio peso y el peso que se desee adicionar, de esta manera se determina que los motores si son capaces de mover estas cargas.

TABLA 2: Comparación de la fuerza aplicada y la requerida para el movimiento del robot.

Condición	Peso Total (kg)	Fuerza por Llanta (N)	Fuerza Aplicada por Motor (N)	Capacidad de Movimiento
sin carga	50	122.6	541.4	Suficiente
con 100 kg	150	367.9	541.4	Suficiente

4.2.3 Sistema de adquisición de datos para la medición de velocidad

Para la obtención de datos reales de velocidades en cada motor, se implementa un sistema de medición basados en encoders ópticos rotativos que consiste en el módulo FC-03 y su respectiva rueda de 20 orificios como se puede observar en la **Fig. 4.6**. El sensor óptico del módulo genera una serie de pulsos digitales cada vez que la rueda interrumpe la luz del sensor, el microcontrolador cuenta estos pulsos y con base en el tiempo que ha transcurrido entre interrupciones, calcula la velocidad de giro del motor en RPM (revoluciones por minuto).



Fig. 4.6 Módulo FC-03 con rueda de 20 orificios.

En cada módulo se usan tres pines: el pin de alimentación de 5V, el de referencia a tierra y el pin de datos digitales. Son cuatro módulos que se integran en el robot siguiendo el diagrama de la **Fig. 4.7**. El pin de datos digitales del módulo se conecta a entradas digitales del microcontrolador que disponen de interrupciones para realizar el conteo de los pulsos de manera adecuada, con este fin se usan las entradas 2,3,20 y 21 del Arduino Mega.

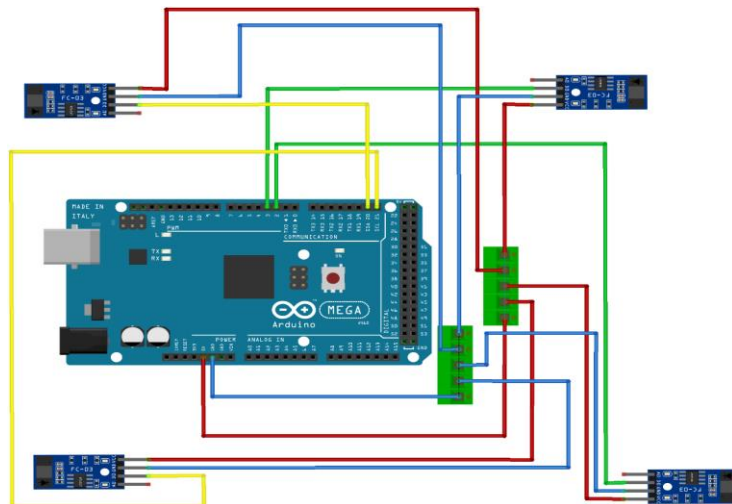


Fig. 4.7 Diagrama de conexión del módulo FC-03 con el Arduino Mega.

4.2.4 Alimentación

Para proporcionar energía al sistema, se ha implementado un banco de baterías compuesto por cuatro baterías de plomo-ácido de 12V y 9Ah. Estas baterías son de válvula regulada y libre mantenimiento, lo que garantiza una mayor seguridad durante las operaciones que realiza el robot. La batería escogida se muestra en la **Fig. 4.8**.



Fig. 4.8 Batería de plomo-ácido de válvula regulada 12V9AH AGM.

4.2.4.1 Esquema y conexión serie-paralelo

A fin de obtener un voltaje y amperaje adecuado para la correcta locomoción del robot, las baterías han sido conectadas en una configuración serie-paralelo como se observa en la **Fig. 4.9**. Se conecta dos baterías en serie sumando así sus voltajes, por lo que dos baterías de 12V en serie resultan en 24V. Se repite el proceso con otras dos baterías obteniendo así dos conjuntos de baterías de 24V y 9Ah. A los conjuntos de baterías se los conecta en paralelo, al conectar en paralelo la capacidad total en amperios-hora se suma, resultando así un banco de baterías de 24V y 18Ah, aumentando así la autonomía del robot.

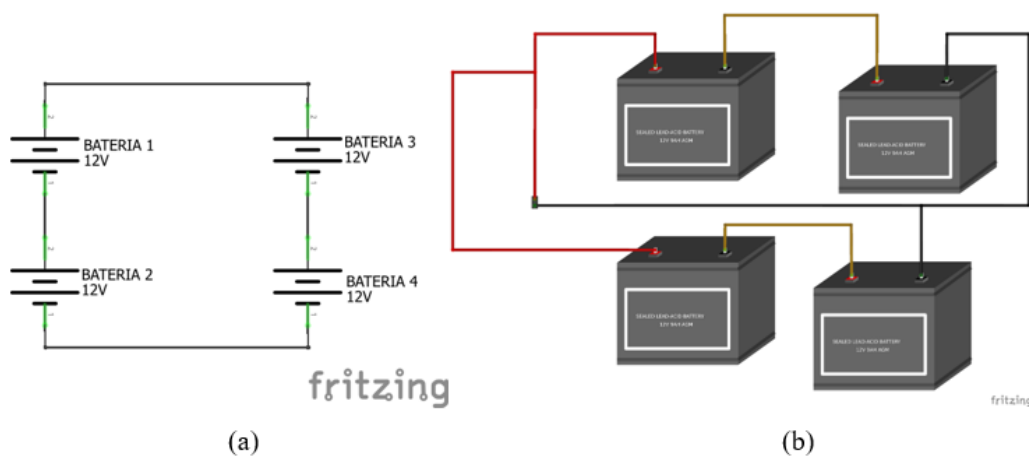


Fig. 4.9 Banco de baterías. (a) Esquemático eléctrico. (b) Diagrama de cableado.

En el terminal positivo del banco de baterías se coloca un switch de dos posiciones como el que se visualiza en la **Fig. 4.10**, este elemento corta el suministro de energía a los componentes apagando así el robot cuando no se esté usando y encendiéndolo cuando sea necesario.



Fig. 4.10 Switch interruptor de 2 posiciones On/Off.

4.2.4.2 Consumo y autonomía estimada

El sistema de suministro de energía está diseñado para entregar energía tanto a los motores como al microcontrolador. La conexión del banco de baterías está configurada para suministrar 24V directamente a los motores, los cuales gracias a la configuración de su tarjeta electrónica robusta suministran 5V a través de sus puertos de control. Esta característica se utiliza para alimentar al Arduino Mega y los módulos adicionales conectados a él, eliminando así la necesidad de un regulador de voltaje externo para estos componentes.

Dado este esquema, la corriente necesaria para el sistema sería la suma del consumo de los 4 motores que da 16 amperios, usando la corriente nominal de 4 amperios de cada motor, y el consumo del microcontrolador con sus módulos que es en promedio de 0.12 amperios, dando un total de 16.12 amperios.

Para calcular la autonomía del banco de baterías se usa

$$\begin{aligned} \text{Autonomía} &= \frac{\text{Capacidad total (Ah)}}{\text{Consumo total (A)}}, & (4.3) \\ \text{Autonomía} &= \frac{18 \text{ Ah}}{16.12 \text{ A}} = 1.12 \text{ horas.} \end{aligned}$$

Bajo condiciones nominales de consumo, el robot puede operar aproximadamente 1.12 horas antes de agotar la capacidad del banco de baterías. Sin embargo, es importante destacar que el consumo nominal de 4 amperios por motor es bajo condiciones de carga bastante elevadas. Así, al usar los motores en terrenos planos con pocos giros y carga, el consumo real es sustancialmente menor, lo que hace que la autonomía del robot se extienda considerablemente.

Para cargar las baterías, es necesario desconectar los cables que en ese momento tenga en sus pines y se procede a cargar una por una con cualquier módulo de carga para baterías de vehículos, es necesario que todas tengan el mismo nivel de carga antes de volver a conectarlas juntas.

4.2.4.3 Alimentación independiente para la Raspberry Pi 5

Para garantizar una alimentación estable y de larga duración a la Raspberry Pi 5, se incorpora un power bank de 30800 mAh como fuente de energía independiente. Este dispositivo cuenta con una pantalla led, como se puede apreciar en la **Fig. 4.11**, permitiendo monitorear el nivel de batería, lo que facilita la gestión de la autonomía. Además, entrega el voltaje y la corriente necesarias para el correcto funcionamiento de la microcomputadora en diferentes escenarios de trabajo.



Fig. 4.11 Power bank marca FOCHEW de 30800 miliamperios hora.

El consumo de corriente de la Raspberry Pi 5 oscila entre los 3 amperios cuando se ejecutan diferentes tareas en el procesador. Con esta información, la autonomía de operación del power bank está dada por

$$Autonomía = \frac{Capacidad\ del\ power\ bank\ (mAh)}{Consumo\ de\ la\ Raspberry\ Pi\ 5\ (mA)}, \quad (4.4)$$

$$Autonomía = \frac{30800\ mAh}{3000\ mA} = 10.26\ horas.$$

El power bank es capaz de proporcionar energía a la Raspberry Pi 5 por aproximadamente 10.26 horas, permitiendo un uso prolongado sin necesidad de recargarlo con frecuencia. El uso de un power bank de alta capacidad ofrece la ventaja de poder trabajar y programar en la Raspberry Pi 5 durante varias horas, sin necesidad de que todo el robot este encendido.

Otra ventaja de tener este power bank es que posee dos salidas de alimentación, dando así la posibilidad de conectar de manera adicional una computadora de IA en placa única como la de la **Fig. 4.12**. Esta sirve para el desarrollo de la visión artificial del robot y evitar que necesite su propia fuente de alimentación si así se desea [26].



Fig. 4.12 Jetson nano 2 GB developer kit diseñada por NVIDIA.

4.2.5 Tipos de cable

El banco de baterías tiene un amperaje considerable, por lo que se toma en cuenta el grosor de los cables empleados. Para el banco de baterías se emplea cable 12 AWG de color negro, los terminales positivo y negativo del banco de baterías se conectan a una bornera de distribución como la que se observa en la **Fig. 4.13**, el uso de este tipo de borneras es para repartir la carga y no generar puntos de calor que quemen el circuito.



Fig. 4.13 Bornera de distribución de amperajes elevados.

De la bornera a los motores se usa cable 16 AWG de color verde. A cada motor se le conecta el cable de 24V y el cable negativo, según lo ilustrado en la **Fig. 4.14**.

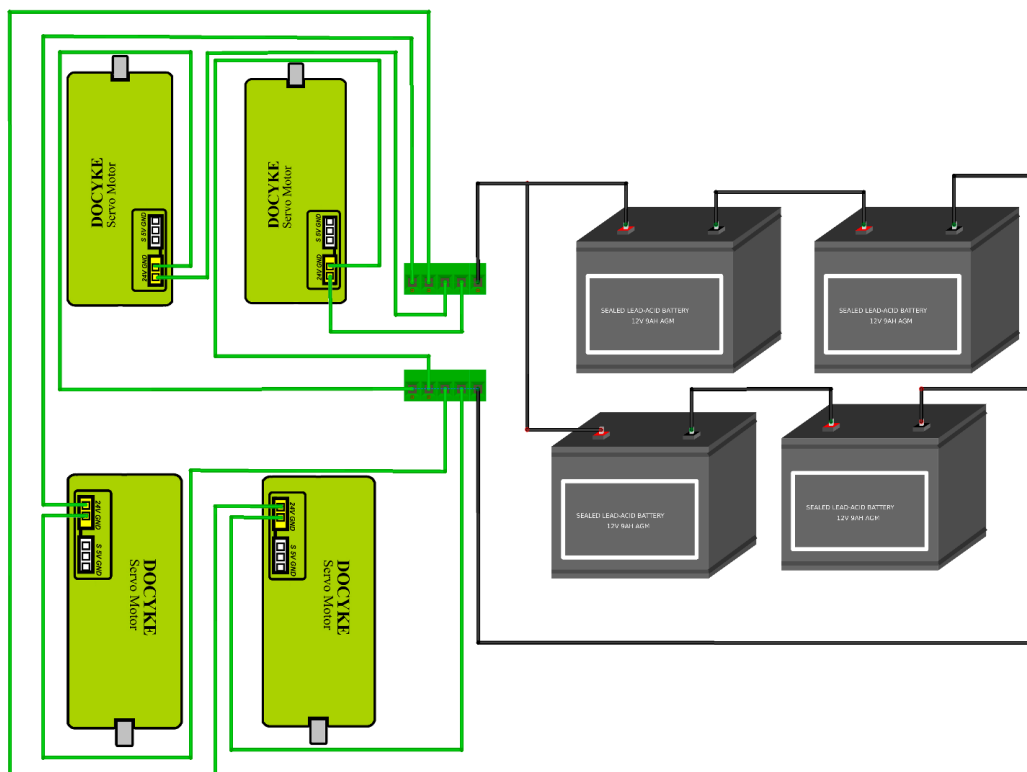


Fig. 4.14 Diagrama de cableado motores y banco de baterías.

4.3 Programación y control del robot

Para la programación y el control del robot se emplea Arduino IDE para el Arduino Mega y ROS 2 ejecutándose en la Raspberry Pi 5. Arduino IDE permite programar el microcontrolador Arduino Mega encargado de generar las señales PWM para controlar los motores y recopilar datos de los sensores. Esto resulta ideal para ejecutar tareas en tiempo real y manejar hardware de forma directa

ROS 2 se utiliza para la integración a un nivel superior, proporcionando una plataforma robusta para la comunicación entre nodos, el procesamiento de datos y la visualización de distintos parámetros. Esta herramienta facilita la interacción entre el robot y otros sistemas, permitiendo coordinar información y escalar la operación a medida que se integran nuevos dispositivos o funcionalidades.

En conjunto, el Arduino Mega se ocupa de las tareas críticas en tiempo real, mientras que la Raspberry Pi 5 ejecutando ROS 2 gestiona la visualización y la comunicación entre nodos, logrando un sistema versátil y escalable.

4.3.1 Lógica de control y flujo del programa en Arduino

El diagrama de flujo de la **Fig. 4.15** ilustra la secuencia de operaciones: inicio, configuración, recepción de comandos, ejecución de acciones, cálculo de RPM y comunicación de datos. El código de este programa se encuentra en el url del anexo A. El programa se organiza iniciando con la configuración de motores y sensores en el *setup*, seguido de la activación de interrupciones para el conteo de pulsos. En el *loop* se procesa la recepción de comandos vía Bluetooth a través de *Serial1* y se ejecutan las acciones correspondientes como avanzar, retroceder o girar mediante la función *write* de la librería Servo. Además, cada 2 segundos se calcula la velocidad en RPM de los 2 motores del lado derecho y se hace un promedio, lo mismo ocurre con los 2 motores del lado izquierdo a partir de los pulsos acumulados de los encoders, enviando esta información a la Raspberry Pi 5 para su visualización y procesamiento.

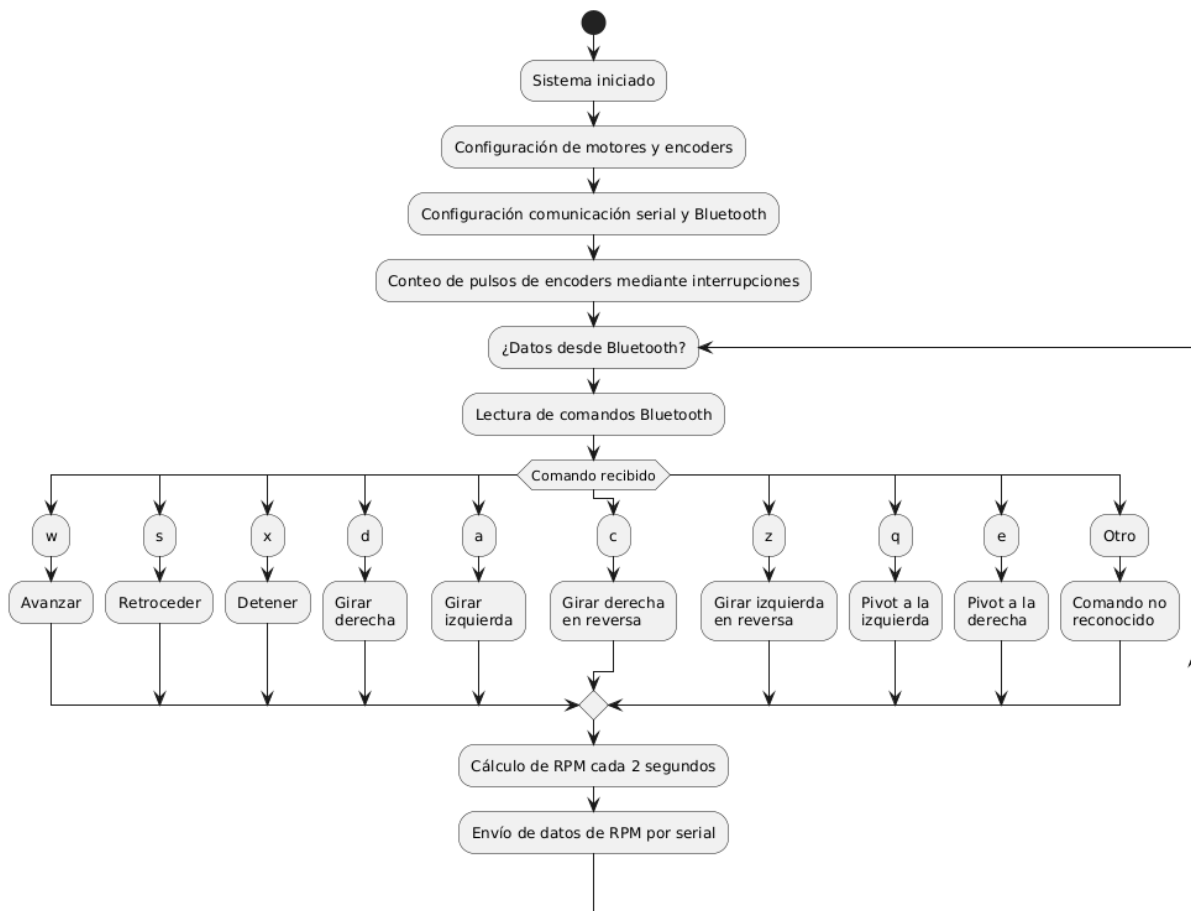


Fig. 4.15 Diagrama de flujo del programa desarrollado para Arduino Mega.

4.3.1.1 Conectividad Bluetooth

El módulo Bluetooth HC-06 se integra en el sistema para transmitir y recibir comandos de forma remota. Se usan los pines 19 y 18 del Arduino Mega para la conexión con el módulo como se aprecia en la **Fig. 4.16**, en lugar de los pines RX y TX tradicionales, lo que optimiza la asignación de recursos y evita interferencias con el monitor serial.

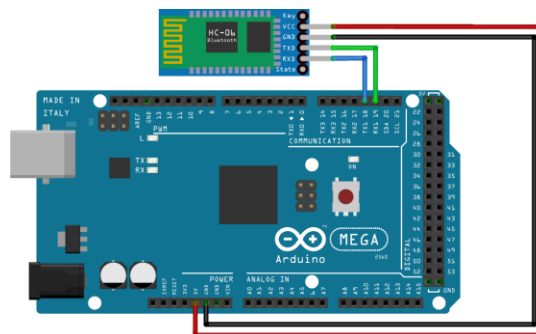


Fig. 4.16 Diagrama de cableado para la comunicación Bluetooth.

4.3.1.2 Pines físicos

Se emplean los pines 9, 10, 11 y 12 para generar las señales de los motores, mientras que los pines 2, 3, 20 y 21 se utilizan para la lectura de datos de los encoders. Además, se asignan los pines 19 y 18 para la comunicación con el módulo Bluetooth, y se utilizan los pines de 5V y GND para la alimentación de todo el sistema de control, dando un total de 12 pines físicos ocupados que se detallan en la **Fig. 4.17**.

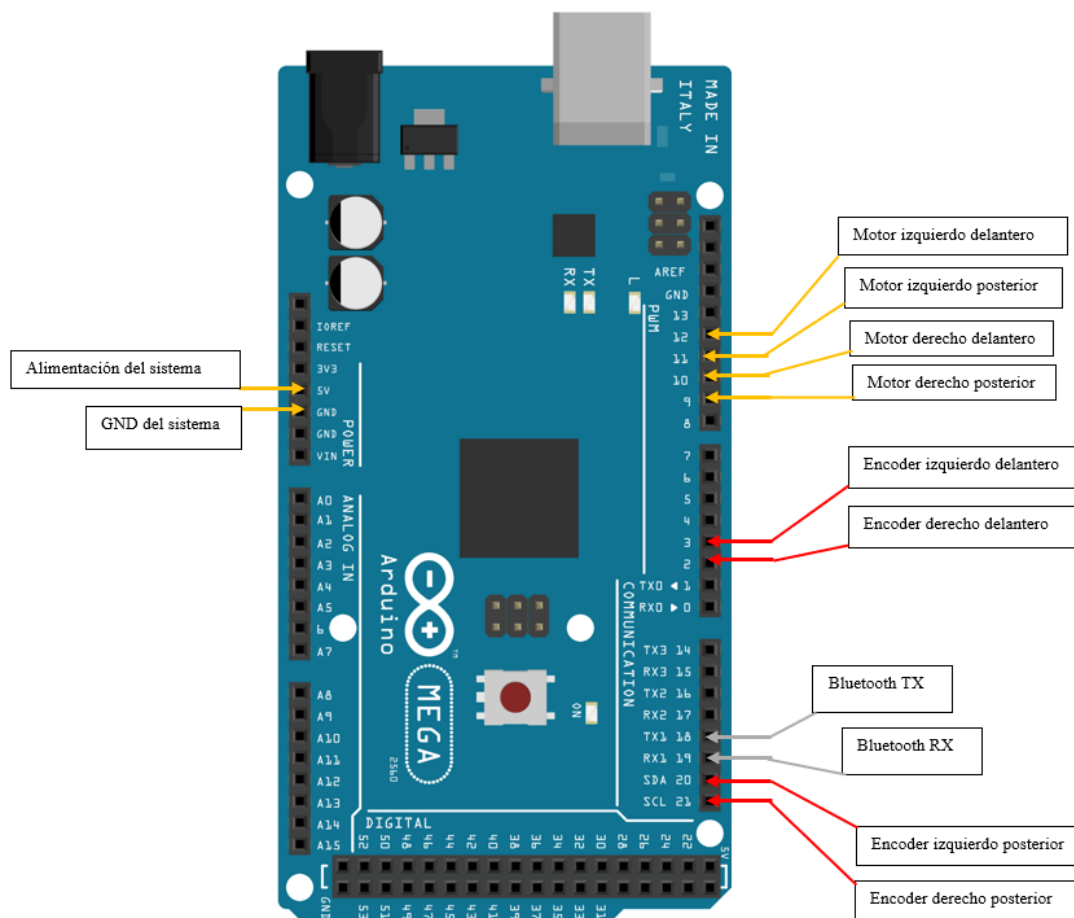


Fig. 4.17 Diagrama de pines empleados del Arduino Mega.

4.3.2 Aplicación móvil

La **Fig. 4.18** ilustra el entorno online de MIT App Inventor, que se emplea en el desarrollo de la aplicación para dispositivos móviles Android con el fin de mover el robot vía bluetooth. Este entorno ofrece herramientas sencillas e intuitivas, lo que facilita la programación. La aplicación descargable se encuentra en el url del anexo A.

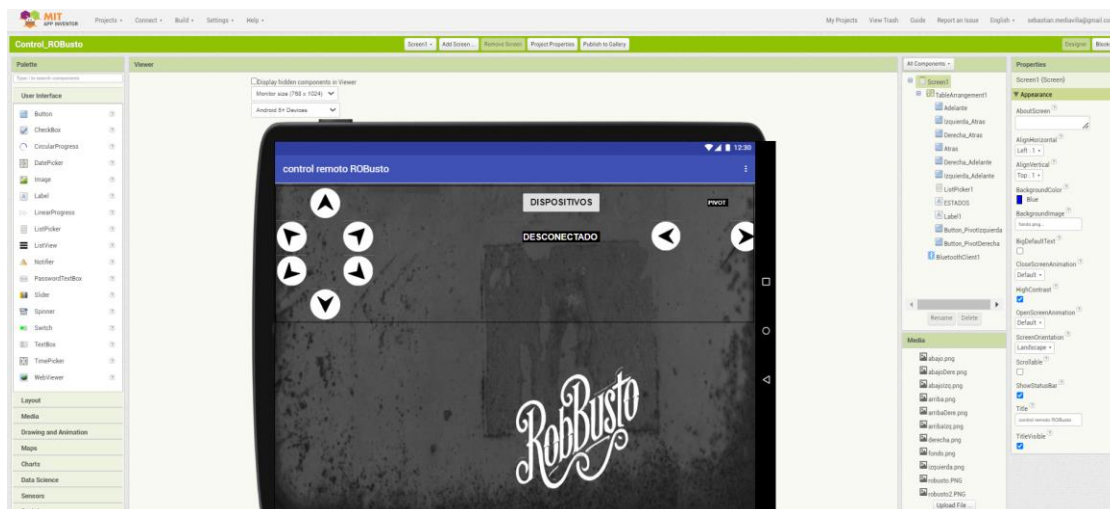


Fig. 4.18 Entorno online de MIT App Inventor.

4.3.2.1 Interfaz gráfica

La aplicación cuenta con dos pantallas. En la pantalla principal se muestra, a la izquierda, las flechas para mover el robot por el entorno, a la derecha, las flechas para girarlo sobre su propio eje, y en el centro, el botón para seleccionar el dispositivo bluetooth como se observa en la **Fig. 4.19**. La **Fig. 4.20** muestra la pantalla secundaria, a la cual se accede mediante el botón de dispositivos, presenta una lista de los dispositivos bluetooth cercanos a los que se puede vincular.



Fig. 4.19 Interfaz principal de control del robot.

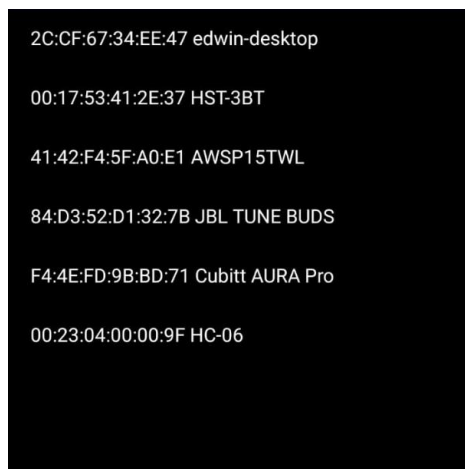


Fig. 4.20 Menú de selección de dispositivos Bluetooth.

4.3.2.2 Diagrama de bloques

La **Fig. 4.21** muestra la programación por bloques de la aplicación móvil. La estructura se encuentra dividida en secciones dedicadas a la comunicación Bluetooth y al envío de comandos según el botón que se presiona.

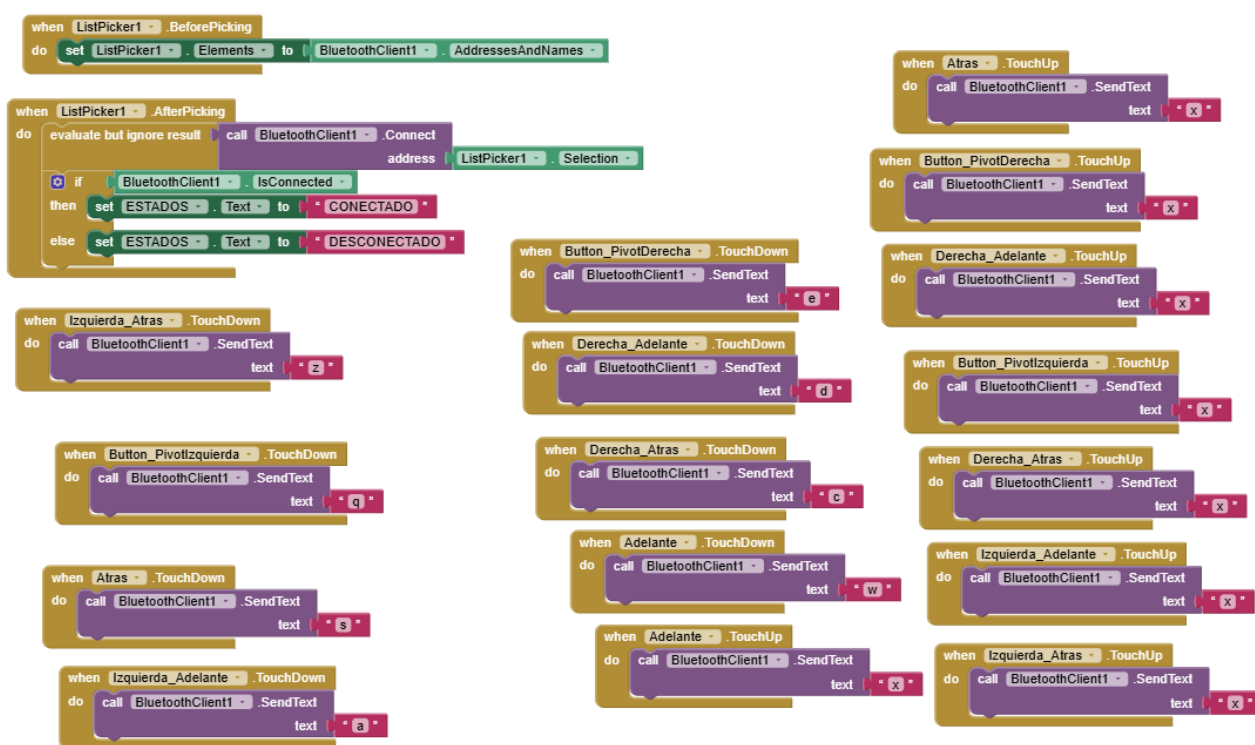


Fig. 4.21 Diagrama de bloques de la aplicación móvil.

4.3.3 Instalación del sistema operativo para la Raspberry Pi 5

En la Raspberry Pi 5 se instala *Ubuntu 24.04.1 LTS*, utilizando el software *Raspberry Pi Imager* para grabar la imagen del sistema operativo en una tarjeta microSD de 128 GB. Se selecciona esta distribución de Linux porque es compatible con ROS 2 facilitando así la configuración de paquetes y dependencias.

4.3.3.1 Comunicación entre Arduino y la Raspberry Pi 5

La comunicación entre el Arduino Mega y la Raspberry Pi 5 se realiza a través de un puerto serial, utilizando un cable que conecta el pin USB del Arduino con uno de los puertos USB de la Raspberry Pi tal como se ve en la **Fig. 4.22**. El Arduino envía cada 2 segundos los valores de RPM de los motores por el puerto serie a 9600 baudios, este envío se lee desde la Raspberry Pi mediante la librería *serial* en Python.

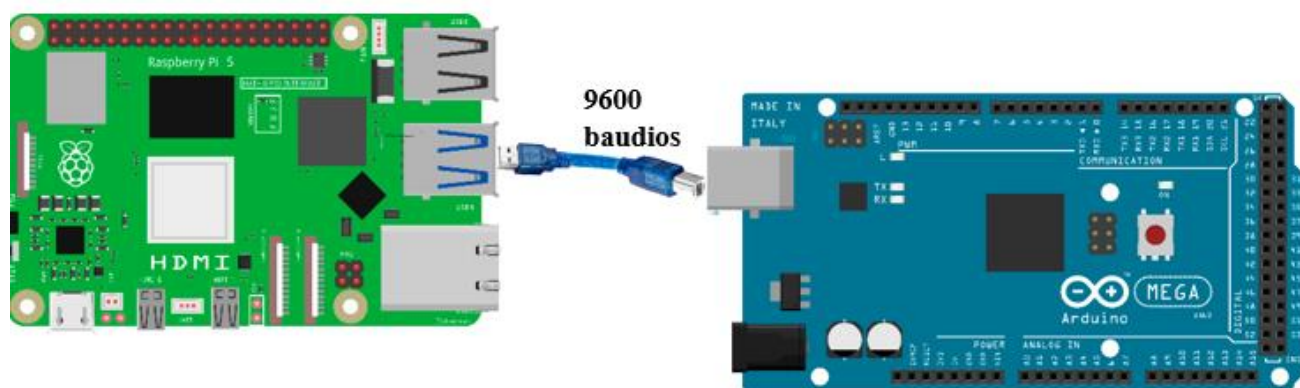


Fig. 4.22 Conexión física entre Raspberry Pi 5 y Arduino Mega.

4.3.4 Montaje de ROS 2

Para instalar la versión *jazzy* de ROS 2 en *Ubuntu 24.04.1 LTS*, se sigue el proceso detallado en la **Fig. 4.23** que incluye la instalación de dependencias y la inicialización de los entornos de desarrollo. Esta instalación se la realiza desde la terminal y es de suma importancia escribir los comandos en el orden estipulado para no tener problemas al momento de ejecutar paquetes y nodos del programa.

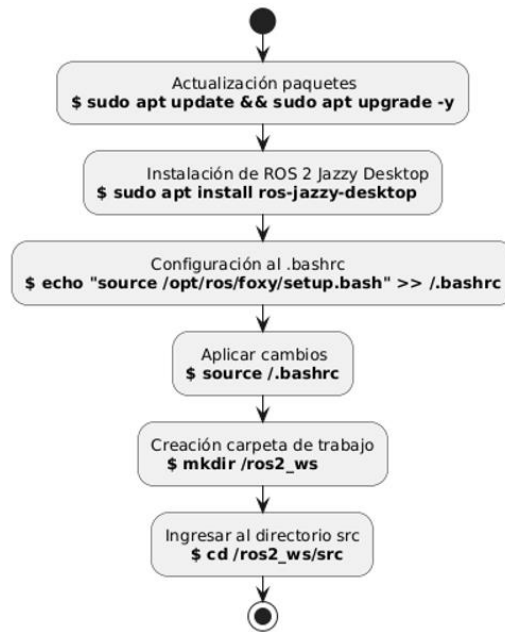


Fig. 4.23 Diagrama de instalación de ROS 2.

4.3.4.1 Creación del nodo medidor de velocidades

La **Fig. 4.24** ilustra los pasos para crear el paquete y posteriormente el nodo de ROS 2 responsable de establecer la conexión serial con el Arduino, leer los datos de RPM y calcular la velocidad lineal de las ruedas. Estos comandos se ejecutan en la terminal.

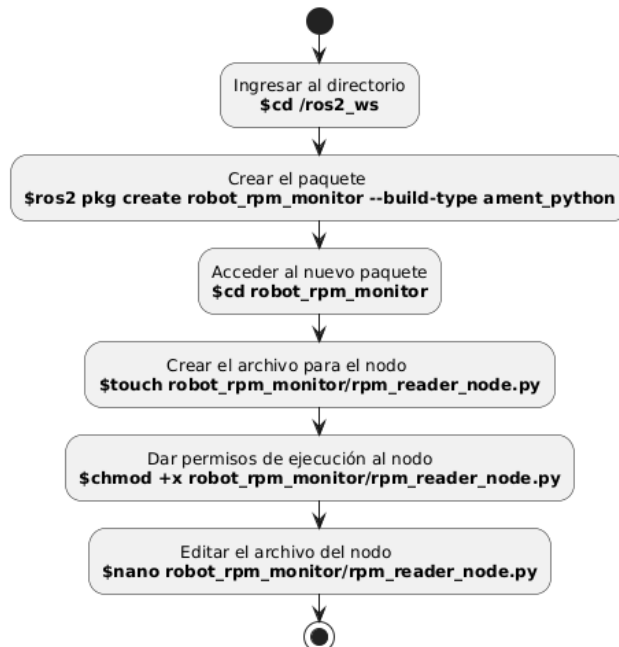


Fig. 4.24 Diagrama de creación de nodo de ROS2.

4.3.4.2 Diagrama de flujo del nodo

El diagrama de flujo de la **Fig. 4.25** describe como el nodo *RPMReaderNode* recibe periódicamente datos por el puerto serial, los separa en RPM de las ruedas del lado derecho y RPM de las ruedas del lado izquierdo, calcula las velocidades lineales de ambas en m/s y finalmente publica estos valores en el tópic *rpm_data*. Este proceso se repite en intervalos de 2 segundos, lo que garantiza la actualización constante de la información. El código de este nodo se encuentra en el url del anexo A.

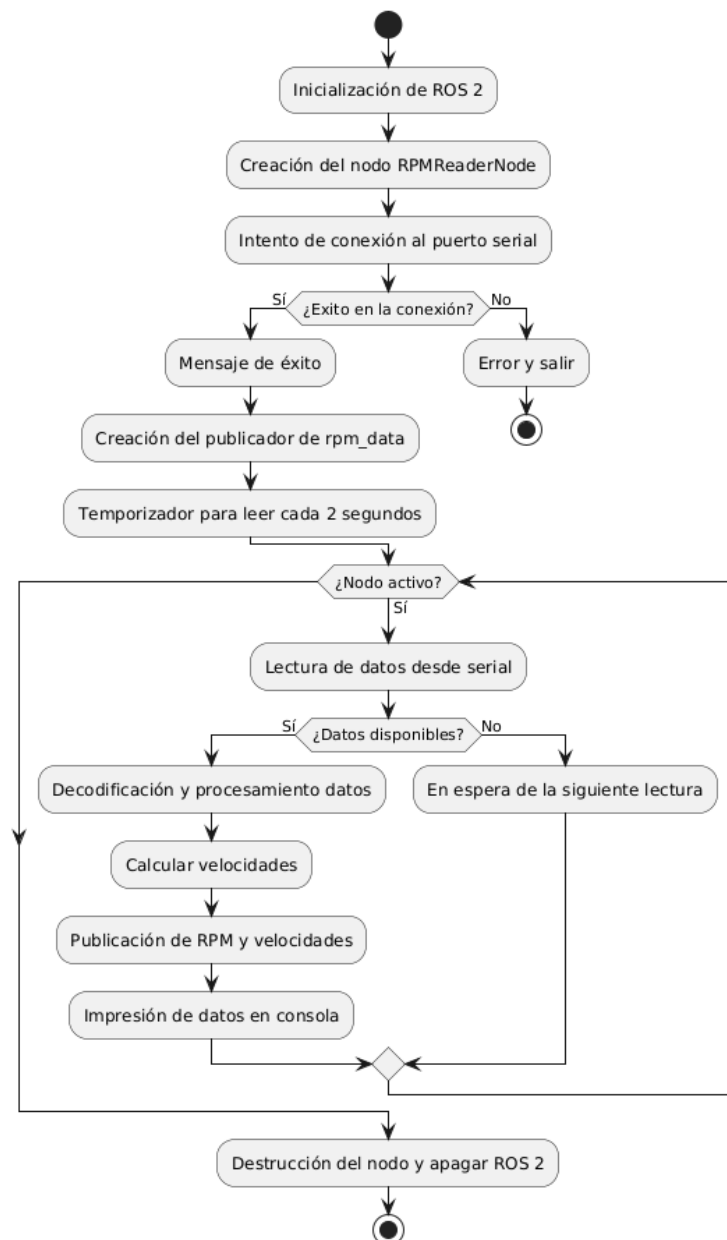


Fig. 4.25 Diagrama de flujo del nodo.

4.3.4.3 Visualización de parámetros

La **Fig. 4.26** muestra el procedimiento necesario para ejecutar el nodo dentro de la terminal y visualizar las velocidades en tiempo real.

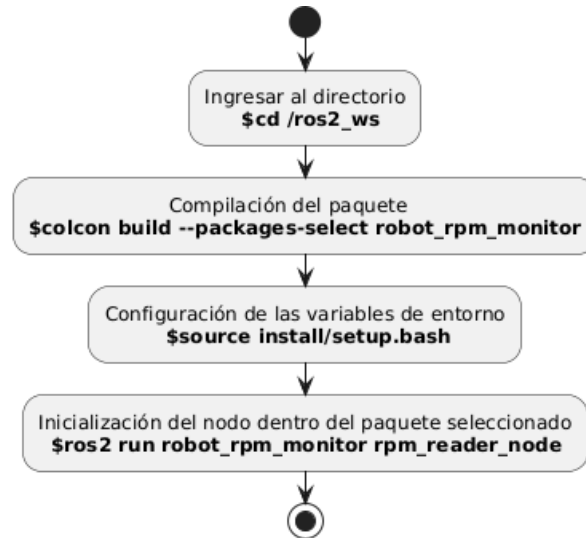


Fig. 4.26 Diagrama de comandos para iniciar el nodo.

De la **Fig. 4.27** se aprecia que, al iniciar este nodo, la terminal empieza a reportar las RPM del lado derecho y del lado izquierdo, así como su conversión a velocidades lineales expresadas en metros por segundo.

```
colcon build [1/1 done] [0 ongoing] 105x25
edwin@edwin-desktop:~$ cd ~/ros2_ws
colcon build --packages-select robot_rpm_monitor
source install/setup.bash
ros2 run robot_rpm_monitor rpm_reader_node
Starting >>> robot_rpm_monitor
Finished <<< robot_rpm_monitor [2.58s]

Summary: 1 package finished [2.90s]
[INFO] [1740371915.178822295] [rpm_reader_node]: Nodo de lectura de RPM iniciado
[INFO] [1740371917.181950759] [rpm_reader_node]: Conexión serial establecida
[INFO] [1740371919.201516205] [rpm_reader_node]: Datos recibidos: Sistema iniciado: Control de Motores y
RPM para 4 encoders
[INFO] [1740371921.200821391] [rpm_reader_node]: Datos recibidos: 0.00,0.00
[INFO] [1740371921.202025319] [rpm_reader_node]: RPM Derecha: 0.0, RPM Izquierda: 0.0
[INFO] [1740371921.203126545] [rpm_reader_node]: Velocidad Derecha: 0.00 m/s, Velocidad Izquierda: 0.00 m
/s
[INFO] [1740371923.200786219] [rpm_reader_node]: Datos recibidos: 0.00,0.00
[INFO] [1740371923.201882667] [rpm_reader_node]: RPM Derecha: 0.0, RPM Izquierda: 0.0
[INFO] [1740371923.202924820] [rpm_reader_node]: Velocidad Derecha: 0.00 m/s, Velocidad Izquierda: 0.00 m
/s
[INFO] [1740371925.200763919] [rpm_reader_node]: Datos recibidos: 0.00,0.00
[INFO] [1740371925.201723962] [rpm_reader_node]: RPM Derecha: 0.0, RPM Izquierda: 0.0
[INFO] [1740371925.202789910] [rpm_reader_node]: Velocidad Derecha: 0.00 m/s, Velocidad Izquierda: 0.00 m
/s
[INFO] [1740371927.200781733] [rpm_reader_node]: Datos recibidos: 0.00,0.00
[INFO] [1740371927.201723962] [rpm_reader_node]: RPM Derecha: 0.0, RPM Izquierda: 0.0
```

Fig. 4.27 Nodo activado y enviando datos.

A partir de estos datos, en una nueva terminal se usa el comando *rqt* para generar gráficas que solo muestren las velocidades en metros por segundo variando en el tiempo, tal como se ilustra en la **Fig. 4.28**. Este proceso de graficar las velocidades se realiza en tiempo real y permite la identificación de cambios en el comportamiento de las ruedas del robot.

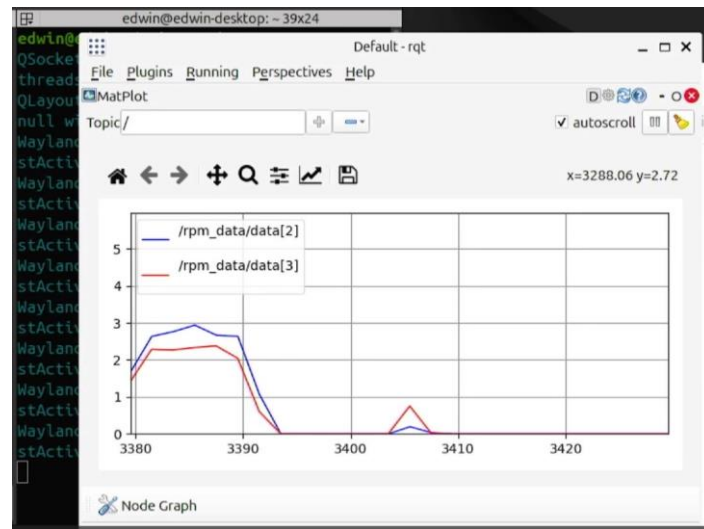


Fig. 4.28 Gráfica generada por el comando *rqt* de las velocidades.

4.4 Ensamblaje del sistema

En esta etapa, se procede al ensamblaje físico de todo el sistema dentro de la estructura construida en [25]. En la **Fig. 4.29** se aprecia la estructura mecánica terminada, lista para albergar los elementos de control, potencia y sensórica que componen el sistema.



Fig. 4.29 Estructura mecánica del robot móvil de alto torque.

4.4.1 Montaje de elementos electrónicos

Se inicia colocando los motores con sus respectivos encoders en la sección inferior del interior del robot. En la **Fig. 4.30** se observa cómo cada motor queda sujeto al chasis y se le conecta su encoder, asegurando un acoplamiento estable para la medición de RPM.

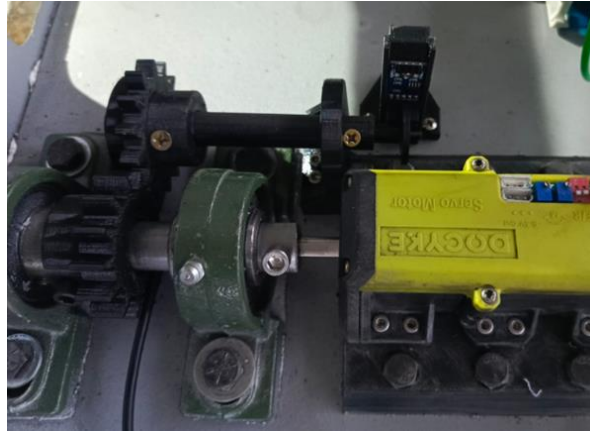


Fig. 4.30 Acople del motor con encoder.

A continuación, se ubica el banco de baterías en la sección superior del interior del robot. Para la distribución del peso, se coloca un bloque de dos baterías a cada lado, mientras que, el power bank se coloca en la parte delantera de la sección inferior como se puede apreciar en la **Fig. 4.31**. El interruptor de dos posiciones con el que se energiza al sistema se instala en el costado izquierdo exterior del robot.

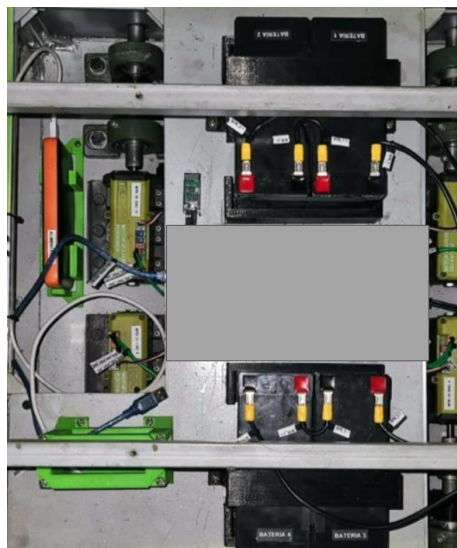


Fig. 4.31 Ubicación de baterías y power bank.

Finalmente, se ubican el Arduino Mega y la Raspberry Pi 5 en sus carcasas correspondientes, también en la sección superior. Se observa mediante la **Fig. 4.32** como cada placa de desarrollo queda protegida y organizada, permitiendo un acceso sencillo a los pines de conexión y a los puertos de comunicación.

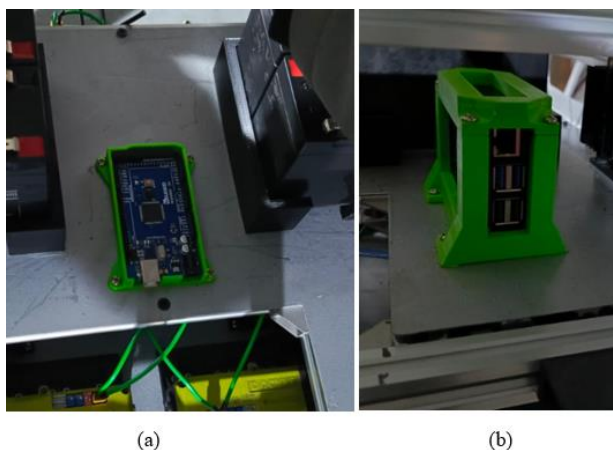


Fig. 4.32 Ubicación del microcontrolador y el minicomputador. (a) Arduino Mega. (b) Raspberry Pi 5.

4.4.2 Cableado del sistema

Como lo muestra la **Fig. 4.33** para el cableado de potencia, se conectan los motores a través de la bornera de distribución que enlaza directamente con el banco de baterías, asegurando una alimentación confiable para cada motor.

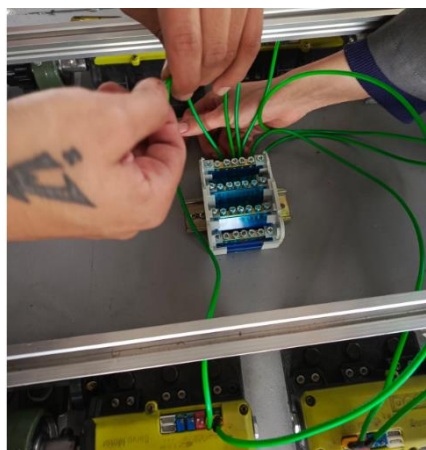


Fig. 4.33 Cableado para alimentación de motores.

Seguidamente, se realizan las conexiones del sistema de control mediante cables Dupont, que vinculan los encoders, el módulo Bluetooth y el manejo de motores con el Arduino Mega.

4.4.3 Fabricación de PCB

Todos los elementos de control requieren 5 voltios y una conexión a tierra o GND, por lo que se diseña y fabrica una PCB que agrupa varias borneras, unificadas en un solo punto de alimentación y un solo punto de tierra. En la **Fig. 4.34** se aprecia el diseño realizado en Fritzing, y en la **Fig. 4.35** se muestra la PCB ya construida, con sus respectivas pistas y las borneras soldadas a la placa.

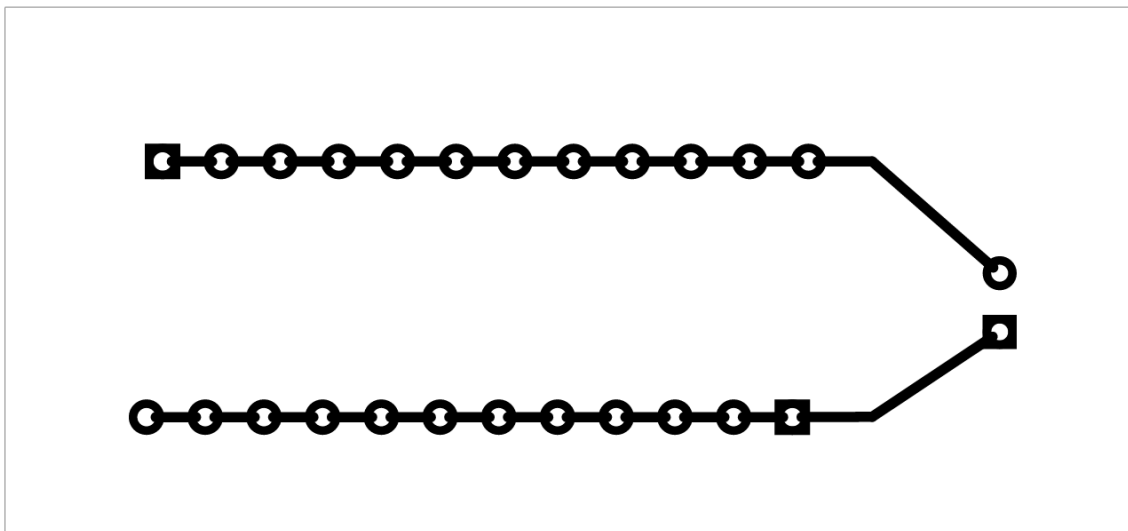


Fig. 4.34 Diseño de PCB en Fritzing.

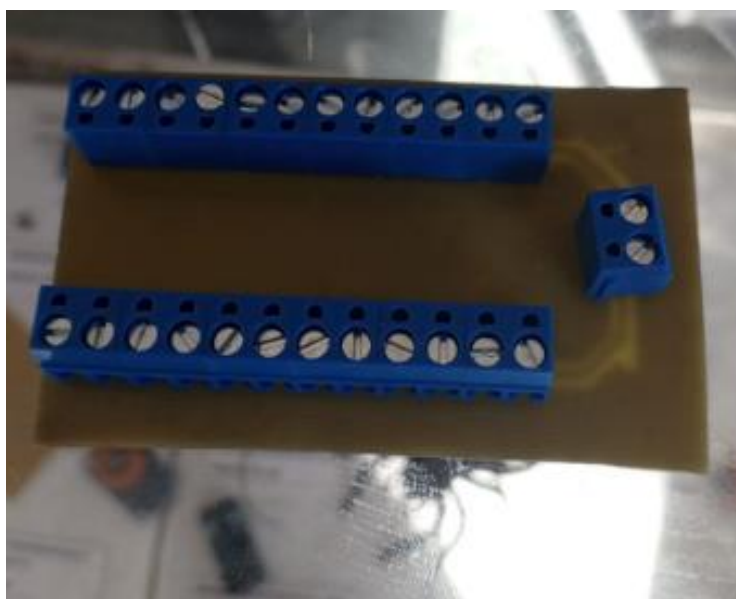


Fig. 4.35 PCB manufacturada.

4.4.4 Etiquetado

Una vez finalizado el cableado, se procede a etiquetar todos los cables con su respectiva nomenclatura para lograr un mayor orden y facilitar futuras revisiones o modificaciones. En la **Fig. 4.36** se observa como cada cable lleva una etiqueta que identifica su función o punto de conexión. Además, se colocan etiquetas en los elementos del sistema, lo que contribuye a una rápida ubicación de cada componente del robot.

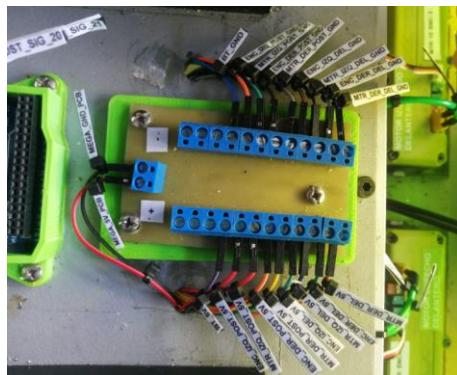


Fig. 4.36 Cables etiquetados.

4.5 Pruebas de rendimiento

En esta fase se somete al robot a distintas evaluaciones para recopilar datos y validar el correcto funcionamiento del sistema. En la **Fig. 4.37** se visualiza como se conecta una pantalla y demás periféricos a las Raspberry Pi 5 para reflejar en tiempo real los cambios de velocidad. Además, se emplea un multímetro para medir la corriente suministrada por el banco de baterías en diferentes escenarios, como se muestra en la **Fig. 4.38**.



Fig. 4.37 Conexión de periféricos a la microcomputadora.



Fig. 4.38 Conexión del multímetro en serie al banco de baterías.

Todas las pruebas realizadas inician sin carga externa, solo con el propio peso del robot que es de 50 kg, y se aumenta el peso gradualmente hasta alcanzar los 185 kg adicionales, tal como se aprecia en la **Fig. 4.39**, donde se observa el robot cargando la máxima masa establecida.



Fig. 4.39 Robot soportando 185 kg adicionales.

4.5.1 Pruebas de consumo de corriente en línea recta en superficie plana

Se evalúa el consumo de corriente cuando el robot se desplaza en línea recta sobre una superficie plana, sometiéndolo a un aumento de cargas externas. El banco de baterías se conecta al multímetro para registrar la corriente en cada escenario. En la **TABLA 3** se muestran los valores de carga externa y la corriente consumida.

TABLA 3: Consumo de corriente en línea recta.

Carga Externa (kg)	Corriente (A)
0	2.50
20	2.60
40	2.70
60	2.90
80	3.00
100	3.10
120	3.20
140	3.40
160	3.70
185	3.90

A partir de estos datos, se genera la gráfica de carga contra corriente de la **Fig. 4.40**, donde se aprecia que, incluso con el incremento progresivo de peso, el consumo de corriente aumenta de manera moderada. Esta tendencia indica que el robot se desempeña de manera regular al desplazarse en línea recta.

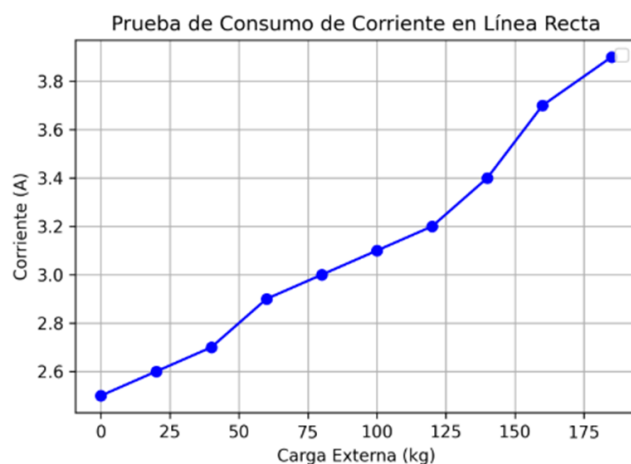


Fig. 4.40 Gráfica carga (kg) vs corriente (A) en línea recta.

4.5.2 Pruebas de consumo de corriente en rotación sobre el propio eje en superficie plana

Se mide el consumo de corriente cuando el robot gira sobre su propio eje, aumentando la carga sobre el de manera progresiva. Se registra la corriente en cada escenario. En la **TABLA 4** se muestran los valores de carga externa y la corriente consumida.

TABLA 4: Consumo de corriente en rotación sobre el propio eje.

Carga Externa (kg)	Corriente (A)
0	6.00
20	6.50
40	7.00
60	8.00
80	9.00
100	10.00
120	12.00
140	14.00
160	16.00
185	18.00

La gráfica de carga contra corriente de la **Fig. 4.41** revela un incremento notable a medida que se añade más peso, evidenciando que los motores se esfuerzan considerablemente al girar sobre su propio eje. Los picos de corriente son sustancialmente mayores que en la prueba de línea recta, aunque se mantienen dentro de los rangos previstos para el banco de baterías. Esta característica confirma que el diseño de alimentación resulta adecuado para soportar maniobras de alta demanda, como las rotaciones y giros con cargas elevadas.

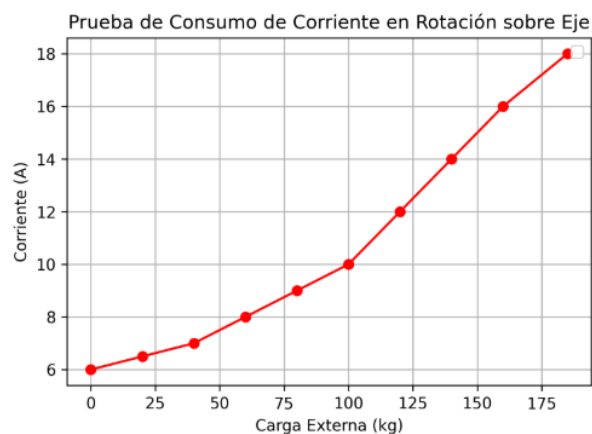


Fig. 4.41 Gráfica carga (kg) vs corriente (A) en rotación sobre el eje.

4.5.3 Pruebas de consistencia de velocidad en línea recta en superficie plana

En esta sección se evalúan las velocidades de las ruedas del lado derecho y de las ruedas del lado izquierdo mientras el robot avanza en línea recta, incrementando la carga externa desde 0 hasta 185 kg. A partir de las gráficas generadas en el nodo de ROS2, se observa cómo varían ambas velocidades conforme aumenta la masa transportada. En la **Fig. 4.42** se aprecia la gráfica de velocidades cuando la carga alcanza los 185 kg, donde se advierte una ligera diferencia entre el lado derecho y el izquierdo.

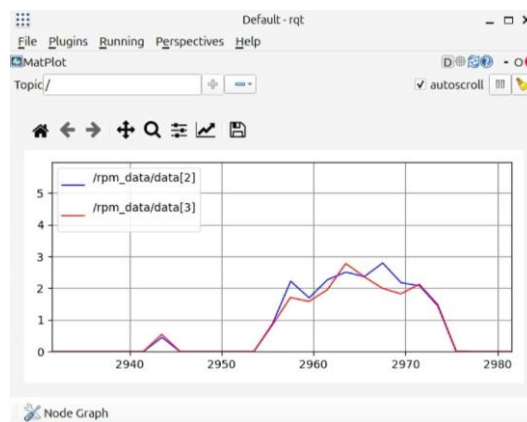


Fig. 4.42 Gráfica de variación de velocidades con 185 kg de carga en línea recta

En la **TABLA 5** se muestran las mediciones obtenidas, con una columna de carga externa y dos columnas que indican las velocidades lineales de cada lado del robot en metros por segundo.

TABLA 5: Consistencia de velocidad en línea recta.

Carga Externa (kg)	Vel. Derecha (m/s)	Vel. Izquierda (m/s)
0	2.75	2.70
20	2.72	2.68
40	2.60	2.58
60	2.50	2.48
80	2.40	2.38
100	2.35	2.33
120	2.30	2.28
140	2.25	2.23
160	2.20	2.18
185	2.10	2.05

En la **Fig. 4.43** resultante, se aprecia que, a medida que aumenta la carga, las velocidades disminuyen progresivamente. Sin embargo, la diferencia entre el lado derecho y el izquierdo no es muy pronunciada, lo cual es favorable porque el robot mantiene un desplazamiento relativamente estable y se dirige hacia su objetivo sin desviaciones significativas.

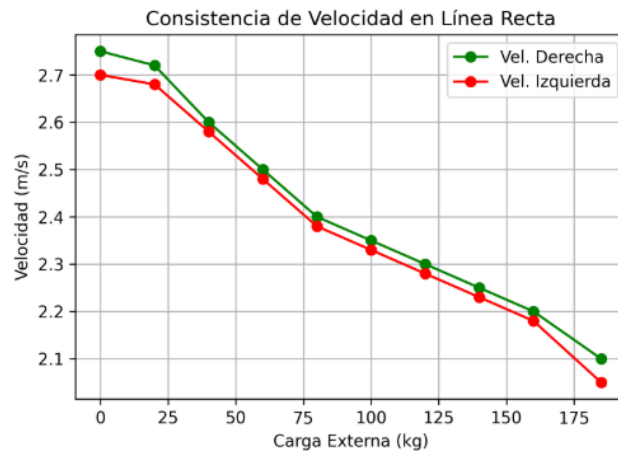


Fig. 4.43 Gráfica carga (kg) vs velocidades (m/s) en línea recta.

4.5.4 Pruebas de consistencia de velocidad en rotación sobre el propio eje en superficie plana

Se analizan las velocidades de cada lado del robot al girar sobre su propio eje, incrementando la carga externa desde 0 hasta 185 kg. Con los datos proporcionados por el nodo de ROS2, se registra un descenso más pronunciado en las velocidades y una mayor diferencia entre los lados. En la **Fig. 4.44** se observa la gráfica correspondiente a la máxima carga de 185 kg, donde se evidencia que el robot no gira de forma perfectamente simétrica.

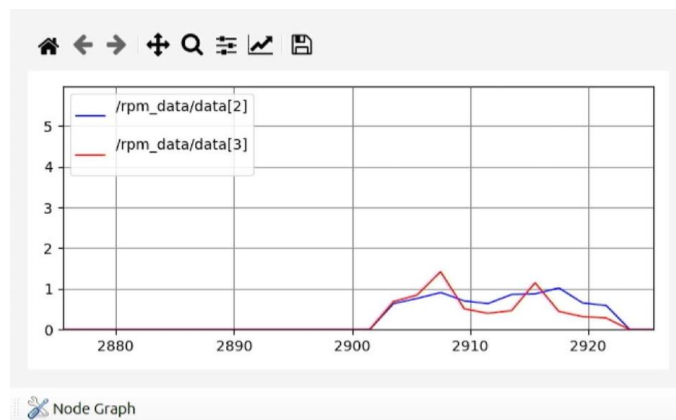


Fig. 4.44 Gráfica de variación de velocidades con 185 kg de carga giro en su propio eje.

La **TABLA 6** presenta los valores de carga y las velocidades de cada rueda en metros por segundo.

TABLA 6: Consistencia de velocidad en rotación sobre el propio eje.

Carga Externa (kg)	Vel. Derecha (m/s)	Vel. Izquierda (m/s)
0	2.65	2.60
20	2.55	2.60
40	2.35	2.15
60	2.15	2.10
80	1.95	1.80
100	1.75	1.85
120	1.55	1.50
140	1.45	1.40
160	1.25	1.18
185	1.05	0.75

De estos valores sale la gráfica de la **Fig. 4.45**, donde se aprecia que, al incrementar la carga, la velocidad de rotación descende de forma considerable y las diferencias entre ambos lados son más notorias que en la prueba de línea recta. Esta variación provoca que el robot no gire en un círculo perfecto y tienda a desplazarse ligeramente durante la maniobra, aunque no se considera alarmante. El sistema se mantiene dentro de los rangos de diseño previstos, demostrando la capacidad del robot para operar con cargas elevadas incluso en movimientos de alta demanda como la rotación sobre su propio eje.

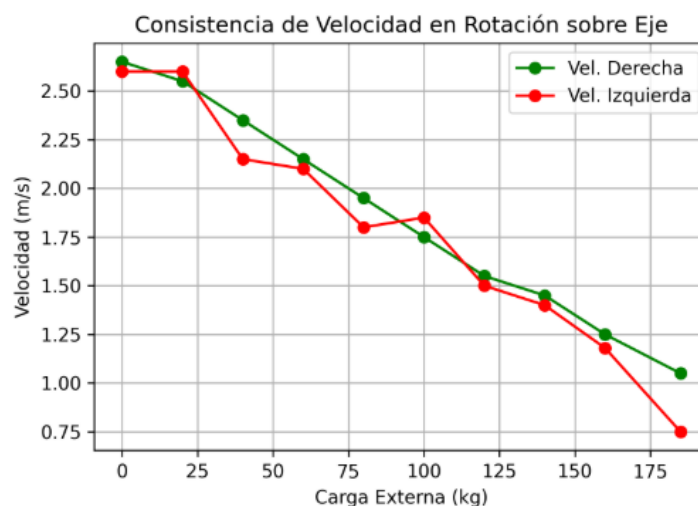


Fig. 4.45 Gráfica carga (kg) vs velocidades (m/s) rotación sobre el eje.

Capítulo V

Conclusiones y recomendaciones

El presente trabajo, centrado en el desarrollo del sistema electrónico de un robot móvil de alto torque, se llevó a cabo en conjunto con el diseño mecánico [25] y el sistema de visión artificial [26], aportando así una solución conjunta que optimizó el rendimiento general del robot.

5.1 Conclusiones

- El diseño del sistema electrónico del robot móvil de alto torque ha demostrado ser eficiente al integrar todos los componentes, garantizando un funcionamiento óptimo.
- La implementación de todos los subsistemas que conforman al robot permitió el equilibrio entre la capacidad energética y la autonomía del robot, lo que mejoró la eficiencia del sistema sin afectar negativamente su funcionalidad.
- Se validó el desempeño del robot a través de distintas pruebas, observando una respuesta estable ante diferentes condiciones de carga. Los resultados demostraron que el sistema es el adecuado para el propósito planteado.

5.2 Recomendaciones

- Se sugiere profundizar en la implementación de algoritmos de odometría para lograr mediciones más precisas de la posición y orientación del robot. Esto facilitará, en el futuro, la aplicación de técnicas de control más avanzadas, como el PID adaptativo o métodos de aprendizaje automático, mejorando así el rendimiento en diversos entornos.

- Se recomienda evaluar la posibilidad de utilizar otro tipo de baterías con mayor densidad energética, mejorando la autonomía del robot sin un aumento significativo en su peso total.
- Para futuros trabajos se propone el desarrollo de nodos y paquetes adicionales que desempeñen funciones específicas, para generar mejoras continuas en el robot y su software.

Bibliografía

- [1] D. Restrepo-Echeverri, J. A. Jiménez-Builes, y J. W. Branch-Bedoya, “Educación 4.0: integración de robótica educativa y dispositivos móviles inteligentes como estrategia didáctica para la formación de ingenieros en STEM”, *Dyna (Medellin)*, vol. 89, núm. 222, pp. 124–135, jul. 2022, doi: 10.15446/dyna.v89n222.100232.
- [2] J. G. Guarnizo Marin, D. Bautista Díaz, y J. S. Sierra Torres, *Una revisión sobre la evolución de la robótica móvil*. Universidad Santo Tomás, 2021. doi: 10.15332/dt.inv.2021.02848.
- [3] M. O. Gonzalez-Fernández, Y. A. González-Flores, y C. Muñoz-López, “Panorama de la robótica educativa a favor del aprendizaje STEAM”, *Revista Eureka sobre Enseñanza y Divulgación de las Ciencias*, vol. 18, núm. 2, pp. 1–19, 2021, doi: 10.25267/Rev_Eureka_ensen_divulg_cienc.2021.v18.i2.2301.
- [4] N. Kerimbayev, N. Beisov, A. Kovtun, N. Nurym, y A. Akramova, “Robotics in the international educational space: Integration and the experience”, *Educ Inf Technol (Dordr)*, vol. 25, núm. 6, pp. 5835–5851, nov. 2020, doi: 10.1007/s10639-020-10257-6.
- [5] L. Nwankwo, C. Fritze, K. Bartsch, y E. Rueckert, “ROMR: A ROS-based open-source mobile robot”, *HardwareX*, vol. 14, p. e00426, jun. 2023, doi: 10.1016/j.ohx.2023.e00426.
- [6] S.-E. Oltean, “Mobile Robot Platform with Arduino Uno and Raspberry Pi for Autonomous Navigation”, *Procedia Manuf*, vol. 32, pp. 572–577, 2019, doi: 10.1016/j.promfg.2019.02.254.

- [7] M. Dejanov, “Application of PID algorithms to control robotic platform using Simulink”, en *2023 18th Conference on Electrical Machines, Drives and Power Systems (ELMA)*, IEEE, jun. 2023, pp. 1–4. doi: 10.1109/ELMA58392.2023.10202414.
- [8] A. H. ISIK y Ö. ÇETİN, “Multifunctional and Low Cost Autonomous Mobile Robot”, *Gazi Journal of Engineering Sciences*, vol. 6, núm. 2, pp. 105–110, ago. 2020, doi: 10.30855/gmbd.2020.02.02.
- [9] A. Neaz, S. Lee, y K. Nam, “Design and Implementation of an Integrated Control System for Omnidirectional Mobile Robots in Industrial Logistics”, *Sensors*, vol. 23, núm. 6, p. 3184, mar. 2023, doi: 10.3390/s23063184.
- [10] M. F. F. Adnan, K. A. Ahmad, R. Boudville, Z. Hussain, y N. I. Husin, “Modelling and Simulation of Omnidirectional Mobile Robot with Line Follower and Obstacles Avoidance”, en *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, IEEE, ago. 2020, pp. 174–179. doi: 10.1109/ICCSCE50387.2020.9204922.
- [11] K. V. Vasireddy, M. Rose G, S. Suresh, R. M, S. Sharma, y S. Mary Joy, “Structural Design and Implementation of Omni-Directional Robot Based on Swerve Drive”, en *2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, IEEE, jul. 2022, pp. 1–7. doi: 10.1109/ICSES55317.2022.9914146.
- [12] L. Hou, L. Zhang, y J. Kim, “Energy Modeling and Power Measurement for Three-Wheeled Omnidirectional Mobile Robots for Path Planning”, *Electronics (Basel)*, vol. 8, núm. 8, p. 843, jul. 2019, doi: 10.3390/electronics8080843.
- [13] J. Armas y A. Cerda, “Implementación de un robot móvil para desplazamientos en ambientes no estructurados empleando visión artificial”, 2020. Consultado: el 19 de enero de 2025. [En línea]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/16378>

- [14] D. Quiroga y B. Rubio, “Robot móvil autónomo para la siembra de semillas en el camp”, 2023. Consultado: el 19 de enero de 2025. [En línea]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/24491>
- [15] J. Aguirre, ““Diseño e implementación de un robot móvil autónomo tipo rover para exploración de terrenos poco estructurados”, 2021. Consultado: el 19 de enero de 2025. [En línea]. Disponible en: <http://tesis.ipn.mx/xmlui/handle/123456789/29391>
- [16] J. Ruiz, “Diseño e implementación de un prototipo robótico móvil todo terreno con un mecanismo de suspensión tipo rocker-bogie para la exploración de entornos de alto riesgo en aplicaciones militares”, Quito, 2022. Consultado: el 19 de enero de 2025. [En línea]. Disponible en: <http://repositorio.espe.edu.ec/jspui/handle/21000/32301>
- [17] “Vista de Etapas en la construcción de un robot móvil con sistema de sensado y comunicación”, 2024. Consultado: el 19 de enero de 2025. [En línea]. Disponible en: <https://repository.uaeh.edu.mx/revistas/index.php/icbi/article/view/11357/10687>
- [18] Harmonic Drive SE, “Motor DC: el pionero de los electromotores”. Consultado: el 19 de enero de 2025. [En línea]. Disponible en: <https://harmonicdrive.de/es/glosario/motor-dc>
- [19] Solectroshop, “¿Qué es PWM y cómo usarlo?” Consultado: el 19 de enero de 2025. [En línea]. Disponible en: https://solectroshop.com/es/blog/que-es-pwm-y-como-usarlo--n38?srsId=AfmBOopK8IeTZFrTI_OaBRA6i3pBvryEuS8iNuJEBY38LYnd79JWNngK
- [20] Derichebourg España, “Batería de plomo ácido: qué es, ventajas y desventajas”. Consultado: el 19 de enero de 2025. [En línea]. Disponible en: <https://www.derichebourgespana.com/bateria-de-plomo-acido/>
- [21] M. Fezari y A. Al-Dahoud, “Raspberry Pi 5: The new Raspberry Pi family with more computation power and AI integration”, 2023. Consultado: el 19 de enero de 2025. [En línea]. Disponible en: https://www.researchgate.net/publication/375552555_Raspberry_Pi_5_The_new_Raspberry_Pi_family_with_more_computation_power_and_AI_integration

- [22] ARDUINO, “Arduino Mega 2560 Rev3”. Consultado: el 19 de enero de 2025. [En línea]. Disponible en: https://store.arduino.cc/products/arduino-mega-2560-rev3?srltid=AfmBOoqZVq1tFMojxiG-u6WNo_aRt2iKrjsYN286lR5lUlmM__9gyBWb
- [23] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, y W. Woodall, “Robot Operating System 2: Design, architecture, and uses in the wild”, *Sci Robot*, vol. 7, núm. 66, may 2022, doi: 10.1126/scirobotics.abm6074.
- [24] Articulated Robotics, “ROS Overview”, Articulated Robotics. Consultado: el 19 de enero de 2025. [En línea]. Disponible en: <https://articulatedrobotics.xyz/tutorials/ready-for-ros/ros-overview>
- [25] A. Guerrero, “ROBOT MÓVIL DE ALTO TORQUE: DESARROLLO MECÁNICO”, Ibarra, 2025.
- [26] P. Hinojosa, “ROBOT MÓVIL DE ALTO TORQUE: VISIÓN ARTIFICIAL”, Ibarra, 2025.

Anexos

Anexo A

Códigos y scripts disponibles en: <https://github.com/pericao320/robot-core>