INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE
Departamento de Procesos Tecnológicos e Industriales

Sustentabilidad y tecnología

PROYECTO DE APLICACIÓN PROFESIONAL (PAP)

Programa de desarrollo tecnológico para la sustentabilidad ambiental energética y alimentaria

4D08 Desarrollo tecnológico para la sustentabilidad ambiental, energética y alimentaria

RoboMesha etapa primavera 2025

PRESENTAN
Programas educativos y Estudiantes
Ing. En Mecatrónica. Fernando Vidal Luna Jaime
Ing. En Mecatrónica. Aaron Homero Sepúlveda Salcedo
Ing. En Mecatrónica. María Alhelí Pérez Rosas
Ing. En Mecatrónica. Alexa Torres Rochín

Profesor PAP: Dr. Jorge Alberto Lizárraga Rodríguez
Tlaquepaque, Jalisco, abril de 2025

Contenido

REPORTE PAP

Presentación Institucional de los Proyectos de Aplicación Profesional

Los Proyectos de Aplicación Profesional (PAP) son experiencias socio-profesionales de los alumnos que desde el currículo de su formación universitaria- enfrentan retos, resuelven problemas o innovan una necesidad sociotécnica del entorno, en vinculación (colaboración) (co-participación) con grupos, instituciones, organizaciones o comunidades, en escenarios reales donde comparten saberes.

El PAP, como espacio curricular de formación vinculada, ha logrado integrar el Servicio Social (acorde con las Orientaciones Fundamentales del ITESO), los requisitos de dar cuenta de los saberes y del saber aplicar los mismos al culminar la formación profesional (Opción Terminal), mediante la realización de proyectos profesionales de cara a las necesidades y retos del entorno (Aplicación Profesional).

El PAP es un proceso acotado en el tiempo en que los estudiantes, los beneficiarios externos y los profesores se asocian colaborativamente y en red, en un proyecto, e incursionan en un mundo social, como actores que enfrentan verdaderos problemas y desafíos traducibles en demandas pertinentes y socialmente relevantes. Frente a éstas transfieren experiencia de sus saberes profesionales y demuestran que saben hacer, innovar, co-crear o transformar en distintos campos sociales.

El PAP trata de sembrar en los estudiantes una disposición permanente de encargarse de la realidad con una actitud comprometida y ética frente a las disimetrías sociales. En otras palabras, se trata del reto de "saber y aprender a transformar".

El Reporte PAP consta de tres componentes:

El primer componente refiere al ciclo participativo del PAP, en donde se documentan las diferentes fases del proyecto y las actividades que tuvieron lugar durante el desarrollo de este y la valoración de las incidencias en el entorno.

En caso de requerirse alguna adecuación al nombre de las fases propuestas para este componente, se puede realizar siempre y cuando sea complementario a lo ya establecido.

El segundo componente presenta los productos elaborados de acuerdo con su tipología.

El tercer componente es la reflexión crítica y ética de la experiencia, el reconocimiento de las competencias y los aprendizajes profesionales que el estudiante desarrolló en el transcurso de su labor.

Resumen

El Proyecto de Aplicación Profesional (PAP) «RoboMesha» (P2025-PAP4D08A) tiene como objetivo desarrollar una mesa hexagonal autopropulsada para apoyar la enseñanza práctica de robótica, control y programación en entornos académicos, en colaboración con la Asociación Tecnológica de la IEEE. Durante el semestre Primavera 2025, se dio continuidad a trabajos anteriores, enfocándose en el diseño del sistema electrónico y la integración de funciones clave como tracción omnidireccional, sensores y comunicación remota.

Los objetivos del semestre incluyeron definir una arquitectura híbrida con Raspberry Pi 4 y Arduino Nano; integrar un controlador para cuatro motores DC con encoders de efecto Hall, validando su funcionamiento mediante I²C; habilitar la recepción remota de comandos vía Firebase; y documentar el desarrollo en un manual técnico para asegurar transferencia de conocimiento.

Se empleó la metodología Design Thinking, un enfoque iterativo centrado en el usuario que abarca cinco fases: Empatizar, Definir, Idear, Prototipar y Evaluar. Esta metodología facilitó una adaptación constante del sistema a las necesidades educativas reales, desde la concepción hasta la implementación funcional.

Como resultado, se logró un subsistema de propulsión estable que permite desplazamientos suaves en los ejes Vx, Vy y ω, controlado desde una interfaz gráfica en React conectada a Firebase. También se verificó la lectura continua de encoders y se estableció una fuente energética eficiente mediante un convertidor buck que regula de 11.1 V a 5 V.

Los entregables incluyeron un manual técnico, esquemáticos, código fuente en Python y React, y un repositorio público en GitHub, dejando bases sólidas para futuras expansiones.

1. Ciclo participativo del Proyecto de Aplicación Profesional

El PAP es una experiencia de aprendizaje y de contribución social integrada por estudiantes, profesores, actores sociales y responsables de las organizaciones, que de manera colaborativa construir sus conocimientos para dar respuestas a problemáticas de un contexto específico y en un tiempo delimitado. Por tanto, la experiencia PAP supone un proceso en lógica de proyecto, así como de un estilo de trabajo participativo y recíproco entre los involucrados.

1.1 Entendimiento del ámbito y del contexto

RoboMesha surge en respuesta a la necesidad de acercar herramientas tecnológicas accesibles y funcionales a escuelas de nivel primaria y secundaria que, por limitaciones económicas, no pueden ofrecer clases de robótica, electrónica o automatización. Estas instituciones educativas, ubicadas en contextos de escasos recursos, carecen frecuentemente del equipo y la infraestructura necesarios para brindar experiencias prácticas en estas disciplinas. La propuesta del proyecto responde a este reto, ofreciendo una plataforma didáctica asequible que promueve la enseñanza activa de habilidades STEAM. A nivel teórico, el proyecto se sustenta en el modelo constructivista del aprendizaje y en enfoques de educación inclusiva orientados a reducir la brecha tecnológica.

1.2 Caracterización de la organización o comunidad

Aquí tienes el texto mejor redactado, con mayor coherencia y manteniendo un estilo similar: La comunidad beneficiaria está conformada por instituciones educativas de nivel básico, cuyos estudiantes enfrentan condiciones de vulnerabilidad económica y tienen escasa exposición a tecnologías emergentes. Este proyecto se lleva a cabo en colaboración con la Asociación Tecnológica de la IEEE, permitiendo al ITESO aportar, mediante este PAP, una plataforma estudiantil orientada a generar soluciones tecnológicas de alto impacto social.

En consecuencia, la misión del proyecto es fortalecer el ecosistema educativo y tecnológico mediante iniciativas colaborativas que promuevan la equidad en el acceso a la tecnología. Los integrantes del PAP forman un equipo multidisciplinario, participando activamente en

áreas como el diseño, la programación, la documentación técnica y la validación práctica de prototipos.

1.3 Identificación de la(s) problemática(s)

Se identificaron diversas problemáticas que enfrentan las escuelas públicas de nivel básico: carencia de equipos para prácticas de robótica, falta de docentes capacitados en tecnologías emergentes, y escasez de materiales educativos adaptados a su realidad. Además, muchas iniciativas previas carecen de escalabilidad y soporte técnico, por lo que no trascienden más allá de una demostración puntual. Desde esta perspectiva, el reto fue crear una solución que pudiera integrarse fácilmente a distintos entornos educativos sin requerir una infraestructura costosa ni conocimientos avanzados.

1.4. Planeación de alternativa(s)

Como alternativa, se diseñó RoboMesha: una mesa robótica hexagonal autopropulsada con tracción omnidireccional, sensores básicos y una interfaz gráfica sencilla. Además, se concibió un sistema completo de comunicación electrónica que separa los sensores y actuadores en dos grupos: los destinados al aprendizaje de los estudiantes y los necesarios para el funcionamiento propio del robot.

Para el uso educativo, se propuso un Arduino externo programable desde una IDE instalada directamente en la Raspberry Pi 5 (basada en arquitectura ARM64), lo que permite a los estudiantes interactuar con los componentes sin requerir un equipo adicional. En cuanto al control funcional del robot, se empleó la Raspberry Pi 5 como unidad central de procesamiento, complementada con un microcontrolador auxiliar (Arduino Nano) para solventar la limitación de pines GPIO.

Sin embargo, esta arquitectura híbrida no pudo implementarse completamente debido a retrasos en la adquisición de componentes por parte del área de compras del ITESO. Como consecuencia, el desarrollo se centró en reutilizar materiales disponibles en el laboratorio de mecatrónica, lo que impidió la integración de la parte educativa del sistema. El esfuerzo se

dirigió a establecer una base funcional robusta del sistema de tracción y movimiento omnidireccional del robot, sobre la cual puedan desarrollarse futuras mejoras.

1.5. Desarrollo de la propuesta de mejora

Durante el desarrollo de RoboMesha se siguió un procedimiento estructurado que permitió implementar alternativas técnicas orientadas a mejorar y validar el funcionamiento del sistema electrónico y robótico. A continuación, se detalla el proceso, herramientas y técnicas utilizadas, así como evidencia visual y descriptiva del trabajo realizado.

Actividades Realizadas y Procedimientos

Inicialmente se definieron los componentes electrónicos que serían utilizados, considerando factores académicos y funcionales como facilidad de integración, disponibilidad, compatibilidad y valor educativo. Se identificaron dos enfoques principales: académico (orientado al aprendizaje práctico con componentes sencillos y accesibles) y funcional (dirigido a la operación estable del robot con capacidades de navegación omnidireccional).

Debido a limitaciones logísticas en la adquisición de componentes específicos (como sensores DHT11, LiDAR, ACS712 y ultrasonido), el equipo se adaptó empleando motores JGB37-520 con encoders integrados, que ya estaban disponibles en el laboratorio de mecatrónica. La integración de estos encoders fue esencial para la validación del control omnidireccional.

Herramientas y Técnicas Utilizadas

Raspberry Pi 4 y Arduino Nano para procesamiento central y periférico, respectivamente.

Comunicación serial UART mediante adaptador USB-Serial para liberación de pines GPIO.

Protocolo de comunicación I2C para controlar motores y leer encoders.

Programación en Python (Raspberry Pi) y C++ (Arduino).

Biblioteca smbus2 para manejo de comunicación I2C.

IDE Arduino instalado directamente en la Raspberry Pi para agilizar la programación y validación local.

Proceso Detallado de Elaboración

Debido a la magnitud del proyecto RoboMesha y su carácter multidisciplinario, el desarrollo del sistema electrónico se abordó en tres fases principales:

1. Electrónica

2. Control

3. Interfaz

Estas fases están estrechamente interrelacionadas, ya que el área electrónica no solo se encarga de la selección y conexión de componentes, sino también de proporcionar datos críticos al área de control, así como de facilitar la interacción con la interfaz de usuario. De igual forma, electrónica depende de los requerimientos definidos por estas otras áreas para realizar una integración funcional completa. Por tanto, el desarrollo electrónico no es un esfuerzo aislado, sino un núcleo integrador dentro del ecosistema RoboMesha.

1. Electrónica

Fundamento Conceptual del Sistema Electrónico

-Enfoque Académico

El sistema electrónico de RoboMesha fue concebido con un fuerte enfoque académico, orientado a brindar a los estudiantes una experiencia de aprendizaje práctica, accesible y efectiva en el campo de la robótica. Para lograrlo, se seleccionaron componentes ampliamente utilizados en el ámbito educativo, priorizando su disponibilidad en el mercado, facilidad de conexión, abundante documentación y compatibilidad con plataformas populares como Arduino y Raspberry Pi.

El uso de LEDs facilita la representación visual de los estados del sistema, haciendo más intuitiva la interpretación de señales digitales. Sensores como el ultrasónico, el seguidor de línea y el sensor de temperatura fueron seleccionados por su bajo nivel de complejidad y su alto valor didáctico. Estos dispositivos permiten enseñar principios de percepción robótica sin requerir conocimientos avanzados de calibración o procesamiento de datos.

Asimismo, se integraron microcontroladores accesibles como el Arduino Nano, reconocido por su bajo costo, facilidad de programación mediante entornos conocidos (IDEs), y una extensa base de proyectos y documentación. Esta combinación permite a los estudiantes desarrollar aplicaciones funcionales con una curva de aprendizaje progresiva.

Componentes seleccionados con fines educativos:
• LEDs indicadores (rojo y verde)
• Buzzer activo 3.3V–5V
• Sensor ultrasónico JSN SR04T
• Sensor de temperatura DHT11
• Seguidor de línea
• Arduino Nano

Estos dispositivos fueron elegidos con base en criterios pedagógicos como: simplicidad de integración, observación directa de resultados, y facilidad de reutilización en múltiples experimentos. Todo ello fomenta el aprendizaje activo y la exploración tecnológica por parte del estudiante.

-Enfoque Funcional

Desde una perspectiva funcional, el objetivo fue lograr un sistema electrónico estable y completo, capaz de operar una plataforma robótica omnidireccional que interactúe en tiempo real con su entorno. Para esto, se integraron sensores, actuadores y controladores capaces de ejecutar tareas clave como la navegación autónoma, la recolección de datos ambientales y la ejecución remota de comandos a través de una interfaz.

La selección de componentes funcionales se hizo bajo criterios técnicos como:
- Compatibilidad con Raspberry Pi 5
- Precisión en mediciones
- Eficiencia energética
- Facilidad de comunicación (I2C, UART, SPI)
- Robustez en entornos educativos

Esta integración garantiza un sistema confiable, adaptable y escalable.

Nombre del componente	Función	Justificación técnica	Estado de uso	Voltaje de operación	Protocolo de comunicación
Raspberry Pi 5	Unidad de procesamiento central	Alta capacidad de cómputo, conectividad avanzada, soporte para Linux y múltiples buses de comunicación.	Probado con Raspberry pi 4 modelo B	5V	I2C
IMU LSM9DS1	Medición inercial	Proporciona datos de orientación y aceleración en 9 ejes; útil para estimación de movimiento.	No probado	3.3V	I2C
Sensor de corriente ACS712	Monitoreo eléctrico	Permite detectar consumo anómalo en los motores; salida análoga compatible con ADC.	No probado	5V	Salida analógica
Sensor de temperatura DHT11	Monitoreo térmico	Bajo costo, fácil integración, útil para validar condiciones térmicas del sistema.	No probado	3.3 – 5V	Comunicación digital (protocolo propietario de un solo hilo)
Sensor ultrasónico JSN SR04T	Detección de obstáculos	Sensor resistente al agua con buena	No probado	5V	Trigger/Echo digital

		precisión y rango adecuado.			
Sensor de batería (divisor de voltaje)	Medición de voltaje	Supervisión del estado de carga; fácil implementación y bajo costo.	No probado	Depende de resistencias	Salida analógica
LiDAR RPLIDAR C1	Escaneo de entorno	Alta resolución angular y velocidad de escaneo, ideal para mapas y navegación.	No probado	5V	UART
Adaptador WiFi RT5370	Comunicación remota	Permite la conexión a Firebase o controladores vía red local o internet.	No probado	5V	USB
Encoder JGB37-520	Retroalimentación de motor	Permite conocer la velocidad y posición de cada rueda con alta precisión.	No probado	5V	Señal digital (cuadratura A/B)
ADC ADS1115	Lectura de sensores analógicos	16 bits de resolución, comunicación I2C, múltiples canales.	No probado	3.3 – 5V	I²C
Buzzer MH-FMG	Alerta sonora	Señales auditivas para advertencias o estados del sistema.	No probado	3.3 – 5V	Señal digital (ON/OFF)
LEDs	Indicadores visuales	Permiten comunicar estados del sistema de forma clara al usuario.	No probado	1.8 – 3.3V	Señal digital (ON/OFF)

Motores JGB37-520	Tracción y movimiento	Motores DC con torque adecuado y baja corriente nominal.	No probado	12V	PWM (mediante puente H)
Pantalla LCD 5” iPistBit	Interfaz visual	Permite mostrar datos y menú interactivo para usuario.	No probado	5V	HDMI o GPIO (según modelo)
Convertidor Buck LM2596	Regulación de voltaje descendente	Provee 5V estables desde 11.1V de batería.	No probado	Entrada 4V–40V / Salida ajustable (típicamente 5V)	N/A
Convertidor Boost 05KJ	Regulación de voltaje ascendente	Eleva voltaje cuando se requiere alimentar módulos de mayor demanda.	No probado	Entrada 2V–24V / Salida ajustable (11.1)	N/A
Optoacopladores PC817	Aislamiento eléctrico	Protege circuitos sensibles en señales de control.	No probado	5V (lado LED)	Aislado (transmisión digital)
Puente H L298N	Control de motores	Control bidireccional de corriente para los 4 motores.	No probado	5V lógica / hasta 35V motores	PWM / ENA-ENB / IN1–IN4
Batería LiPo Zeee 11.1V 6000mAh	Fuente de energía	Alta capacidad y tasa de descarga para mantener estabilidad del sistema.	No probado	11.1V	N/A
Arduino Nano	Control auxiliar	Soporte para módulos secundarios o pruebas didácticas.	No probado	5V	I2C

Tabla 1. Tabla de componentes, su función, justificación, estado de uso, voltaje de operación y protocolo de comunicación.

La integración de estos componentes sienta las bases para un sistema electrónico modular, adaptable y escalable. Su validación progresiva permitirá asegurar un funcionamiento robusto y confiable, alineado con los objetivos de desarrollo tecnológico y formativo del proyecto RoboMesha.

A continuación, mostramos los Planos “maestros”, que se diseñaron como objetivo a futuro para el proyecto:

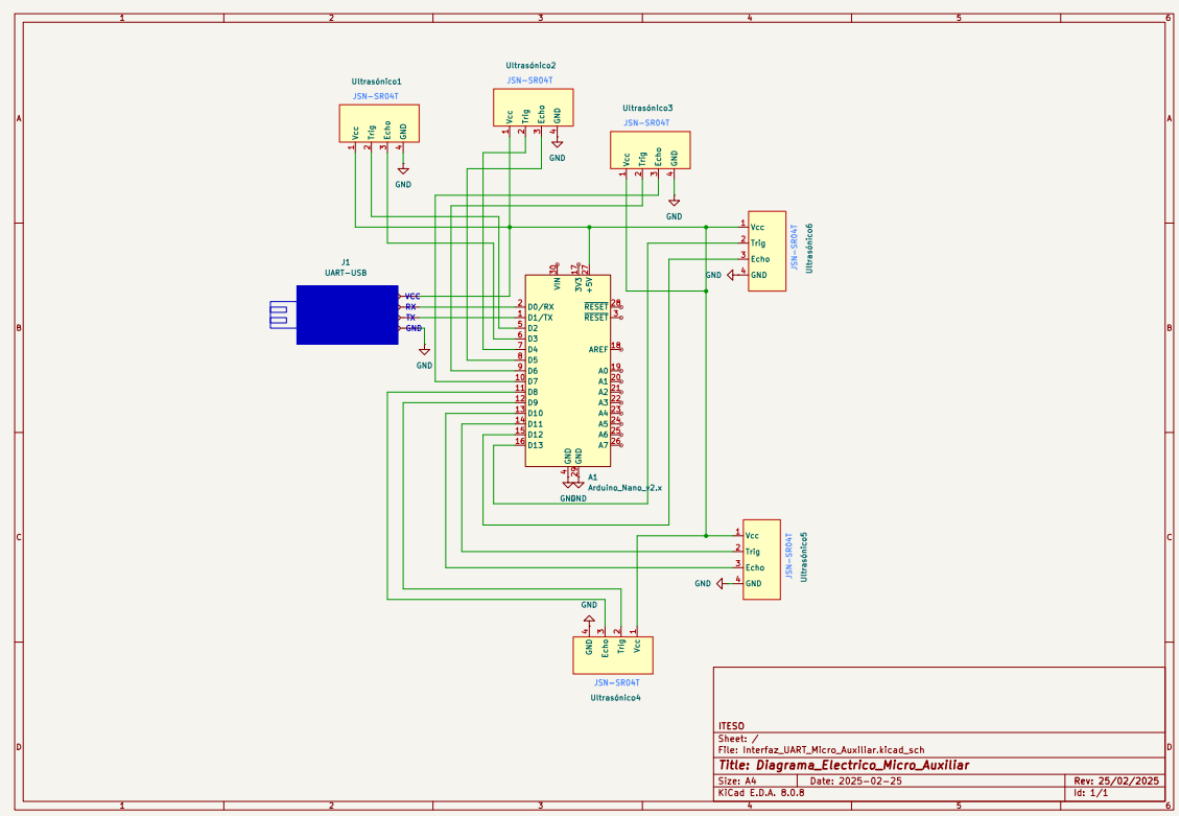

Figura 1. Diagrama eléctrico del microcontrolador auxiliar (Arduino Nano).

En este esquemático se ilustran las conexiones del sistema auxiliar encargado de la lectura de sensores ultrasónicos, el cual cumple una función clave en la percepción del entorno del robot. Se utilizaron seis módulos ultrasónicos JSN-SR04T, distribuidos en distintas

direcciones para proporcionar una cobertura espacial amplia. Estos sensores permiten detectar la proximidad de objetos mediante señales acústicas, generando datos de distancia que son fundamentales para tareas de navegación y evasión de obstáculos.

Los sensores están conectados directamente a un microcontrolador Arduino Nano, que actúa como unidad de procesamiento auxiliar. Este microcontrolador se encarga de gestionar el disparo (Trig) y la recepción (Echo) de cada sensor, calcular las distancias medidas y preparar los datos para su transmisión.

Una vez procesados, los datos son enviados mediante comunicación serial a través del puerto UART del Arduino, el cual está vinculado a un convertidor UART-USB. Este convertidor permite establecer un canal de comunicación entre el microcontrolador auxiliar y la unidad central del sistema: la Raspberry Pi, facilitando así la integración entre el subsistema de percepción y el sistema de control general del robot.

Este diseño modular permite distribuir las tareas de percepción espacial a una unidad secundaria, liberando recursos de procesamiento en la Raspberry Pi y facilitando una arquitectura escalable, eficiente y más fácil de depurar.

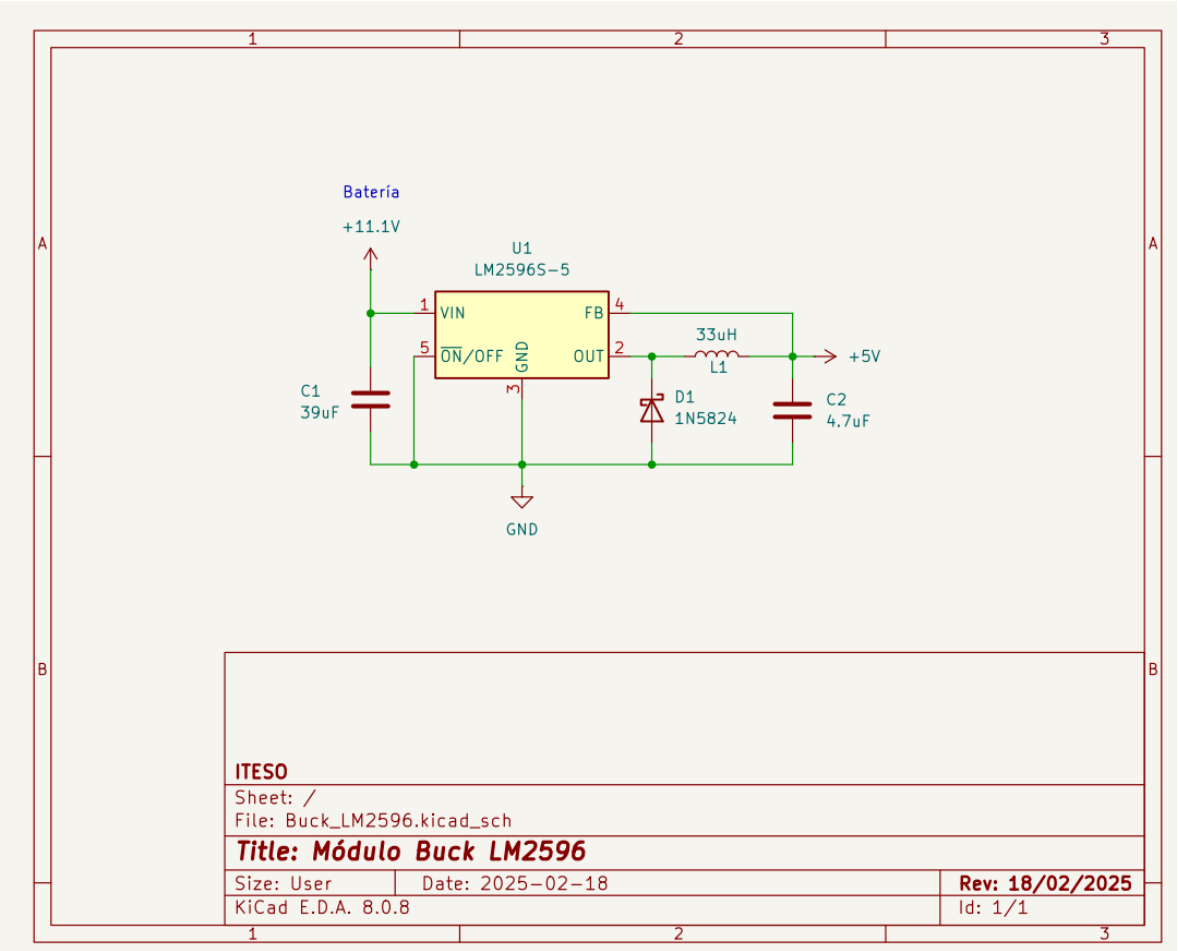

Figura 2. Diagrama eléctrico del Módulo Buck LM2596.

Este diagrama representa el diseño de un módulo reductor de voltaje (buck) basado en el regulador LM2596-5, cuya función principal es suministrar una salida estable de 5V a partir de una fuente de 11.1V proveniente de una batería LiPo, que realmente la batería proporcionaba más voltaje, 16.6 V, que hace que podamos alimentar 3 módulos buck. Esta regulación es esencial para alimentar de forma segura componentes electrónicos sensibles, como microcontroladores y sensores.

El circuito incluye un capacitor de entrada (C1) para filtrar posibles fluctuaciones provenientes de la batería, una bobina de 33 µH (L1) que trabaja junto al diodo Schottky (D1) y al capacitor de salida (C2) para asegurar una conversión eficiente y con baja ondulación de voltaje. El pin ON/OFF del LM2596 también está cableado para permitir su habilitación automática al recibir voltaje de entrada.

Además, se adquirió un segundo módulo Buck del mismo tipo, ajustado para suministrar 12V regulados. Este segundo módulo se destina a alimentar circuitos que requieren mayor voltaje, como sistemas de carga o distribución de potencia a otros subsistemas, manteniendo siempre la estabilidad eléctrica del sistema general.

El uso de estos módulos garantiza un suministro confiable, eficiente y protegido contra sobrecargas, fundamentales en entornos educativos y de prototipado como el de RoboMesha.

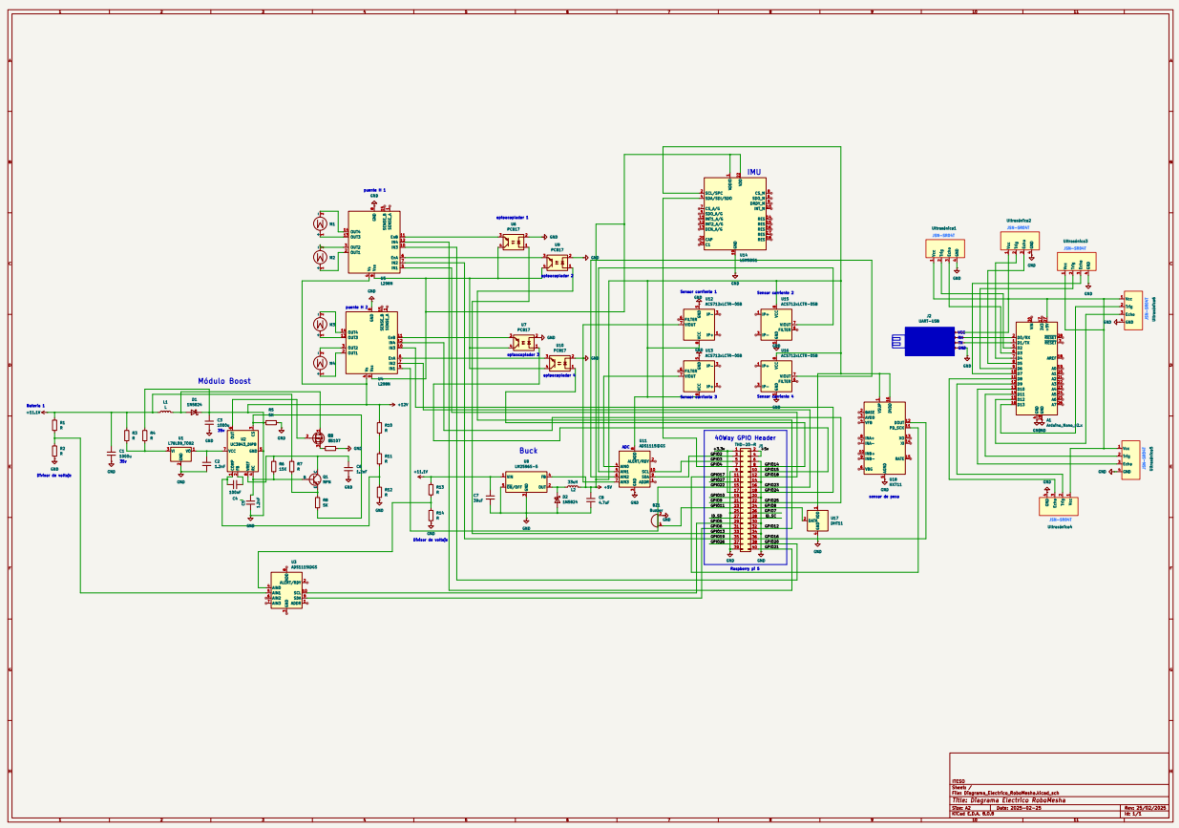

Figura 3. Diagrama eléctrico del sistema electrónico de RoboMesha.

Circuito General del Sistema Electrónico de RoboMesha

La imagen presenta el circuito completo del sistema electrónico de RoboMesha, donde se integran todos los módulos funcionales clave para el control, percepción, alimentación y comunicación del robot. Este diseño centraliza tanto la distribución energética como la gestión de señales provenientes de sensores y actuadores.

A la izquierda se encuentran los módulos de regulación de voltaje:

- Un módulo Boost, encargado de elevar el voltaje de la batería en caso de requerirse tensiones mayores para módulos de alta demanda.
- Un módulo Buck, que proporciona una salida estable de 5V para alimentar microcontroladores y sensores de baja tensión.

En el centro del diagrama se ubica la unidad principal de procesamiento, representada por el conector GPIO de la Raspberry Pi, el cual se interconecta con diversos subsistemas mediante buses de comunicación como I2C, UART y señales digitales. Desde este punto se distribuyen las conexiones hacia los sensores inerciales (IMU), sensores analógicos a través de un ADC ADS1115, así como el control de motores a través de un puente H L298N.

También se incluyen elementos de seguridad y aislamiento eléctrico, como optoacopladores, y múltiples sensores que permiten la percepción ambiental: ultrasónicos, de corriente, temperatura y seguimiento de línea.

A la derecha del esquema se encuentra el subsistema de percepción auxiliar basado en Arduino Nano, previamente detallado, que gestiona múltiples sensores ultrasónicos y transmite los datos a la Raspberry Pi por medio de un módulo UART-USB.

Este diseño busca mantener una arquitectura modular, organizada en bloques funcionales bien diferenciados, que favorezca tanto el diagnóstico como futuras expansiones. El uso de componentes estándar y ampliamente documentados permite una mayor accesibilidad para estudiantes y facilita el mantenimiento del sistema. La combinación de regulación energética, control distribuido y comunicación entre módulos asegura una operación estable y eficiente del robot en condiciones educativas y de prototipado.

2.-Control

Arquitectura de Control

La arquitectura propuesta de RoboMesha contempla una integración entre la Raspberry Pi y un microcontrolador auxiliar (Arduino Nano), diseñada como solución a la limitación de pines GPIO disponibles. Aunque esta arquitectura no pudo implementarse físicamente debido a retrasos logísticos en la adquisición de componentes, el diseño busca distribuir de forma eficiente la gestión de periféricos entre ambos controladores.

El Arduino Nano estaría encargado de módulos secundarios como sensores de ambiente, mientras que la Raspberry Pi se reservaría para procesamiento central y conectividad con la interfaz web. La comunicación entre ambas placas se planteó mediante UART, utilizando un adaptador USB-Serial, lo que evita el uso de pines adicionales en la Raspberry Pi.

Comunicación Serial

El canal UART permite una comunicación sencilla entre el Arduino y la Raspberry Pi a través del puerto USB. Esto expone una interfaz tipo /dev/ttyACM0 en la Raspberry Pi, que puede consultarse con:

```
1. ls /dev/ttyACM*
```

Esta estrategia libera los pines GPIO y simplifica la integración de múltiples sensores al redistribuir las responsabilidades entre ambas placas.

Código de Comunicación Serial

Código en Arduino Nano (C++):

```
1. void setup() {
2.   Serial.begin(9600); // Inicializa el puerto serie
3. }
4.
5. void loop() {
6.   Serial.println("Hola Rasp!"); // Envía mensaje cada segundo
7.   delay(1000);
8. }
9.
```

Código en Raspberry Pi (Python):

```python
import serial

arduino = serial.Serial('/dev/ttyACM0', 9600, timeout=1)
print("Esperando datos del Arduino...")

while True:
    if arduino.in_waiting > 0:
        linea = arduino.readline().decode('utf-8').strip()
        print(f'Recibido: {linea}')

```

Formato y flujo de datos

El Arduino envía mensajes codificados en UTF-8, típicamente en formato de cadena de texto (string). En este caso, los datos fluyen de forma unidireccional (Arduino → Raspberry Pi), con una frecuencia de 1 Hz, pero el canal puede ampliarse para comunicación bidireccional si se requiere enviar comandos desde la Raspberry.

Soporte IDE y Programación

Instalación del IDE Arduino en Raspberry Pi

Para facilitar la programación sin depender de equipos externos, se instaló el IDE de Arduino (v1.8.19) directamente en la Raspberry Pi. Esta versión es compatible con la arquitectura ARM y permite programar microcontroladores desde el mismo entorno embebido.

Pasos de instalación:

1. Crear el directorio de trabajo:

```
mkdir -p ~/ArduinoPgm
cd ~/ArduinoPgm
```

2. Descargar el paquete oficial:

```
wget https://downloads.arduino.cc/arduino-1.8.19-linuxarm.tar.xz
```

3. Descomprimir el archivo:

```
tar -xf arduino-1.8.19-linuxarm.tar.xz
```

4. Ejecutar el instalador:

```
cd arduino-1.8.19
sudo ./install.sh
```

Compatibilidad y sugerencias:

• Funciona correctamente en Raspberry Pi 4 con Raspberry Pi OS (basado en Debian).
• No se recomienda el IDE 2.x debido a incompatibilidades con arquitectura ARM.
• Se recomienda cerrar otras aplicaciones para evitar lentitud al compilar.

Ventajas del enfoque:

Esta instalación convierte a la Raspberry Pi en una estación de desarrollo embebida, ideal para contextos educativos, prototipos portátiles o laboratoros. Permite programar, cargar y probar código directamente desde el dispositivo que controla el sistema físico, ofreciendo una experiencia integral.

Sensores Integrados

Durante el período de desarrollo no se logró integrar nuevos sensores debido a problemas logísticos relacionados con el suministro de componentes. La falta de entrega oportuna por parte del área de compras del ITESO impidió la incorporación de sensores previamente seleccionados como el DHT11 (temperatura y humedad), el ACS712 (sensor de corriente), sensores ultrasónicos de proximidad y el sensor LiDAR.

Como alternativa, se utilizaron únicamente los motores con encoders magnéticos integrados en el chasis del prototipo. Estos encoders, de tipo Hall digital, permitieron medir la velocidad y dirección de giro de cada rueda, aportando información esencial para el control de movimiento omnidireccional.

• Tipo de sensor: Encoder magnético Hall en el motor JGB37-520.
• Protocolo de conexión: Digital, con salida cuadrada conectada a pines GPIO.
• Alimentación: 5V DC.
• Lectura: A través de los pines GPIO de la Raspberry Pi 4.

Lectura básica de encoder en Raspberry Pi:

```
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)
encoder_pin = 9
GPIO.setup(encoder_pin, GPIO.IN, pull_up_down=GPIO.PUD_UP)

def encoder_callback(channel):
    print("Pulso detectado en el encoder.")

GPIO.add_event_detect(encoder_pin, GPIO.RISING, callback=encoder_callback)

try:
    while True:
        pass
except KeyboardInterrupt:
    GPIO.cleanup()

```

Este código permitió detectar los pulsos generados por el encoder durante el movimiento, proporcionando retroalimentación útil para la validación del sistema de tracción.

Driver de Motor Preinstalado

El prototipo de RoboMesha incluye un controlador de motores de la marca HIWONDER, diseñado para manejar hasta cuatro motores DC con encoders integrados. Este driver incorpora control en lazo cerrado, lectura del voltaje de batería y configuración de parámetros como el tipo de motor o polaridad del encoder.

Investigación del Funcionamiento

Ante la falta de documentación oficial sobre el protocolo I²C del driver, se estableció contacto con el proveedor HIWONDER a través de WhatsApp. Se obtuvo acceso a una carpeta de Google Drive con documentación técnica relevante.

Esta información permitió desarrollar un script en Python utilizando la librería smbus2 para controlar los motores desde la Raspberry Pi.

Estructura de Comandos

• 0x34: Dirección I²C del driver.

• 0x14: Selección del tipo de motor.

• 0x15: configuración de la polaridad del encoder

• 0x33: Envío de velocidades a los motores

• 0x3C: Lectura de los valores de encoder acumulados

Configuración y control del driver de motor:

```python
# Configuración de tipo de motor y polaridad
bus.write_byte_data(0x34, 0x14, 3)  # Motor JGB37-520
bus.write_byte_data(0x34, 0x15, 0)  # Polaridad estándar

# Enviar velocidad de avance
velocidad_adelante = [50, -50, -50, 50]
bus.write_i2c_block_data(0x34, 0x33, velocidad_adelante)

# Leer voltaje de batería
bateria = bus.read_i2c_block_data(0x34, 0x00, 2)
voltaje = bateria[1] << 8

# Leer valores de encoder
datos_encoders = bus.read_i2c_block_data(0x34, 0x3C, 16)
encoders = struct.unpack('iiii', bytes(datos_encoders))

```

Control de Movimiento

El control de movimiento omnidireccional de RoboMesha se logró mediante una arquitectura de control colaborativa entre hardware, software y modelado matemático.

Movimientos Habilitados

Se implementaron movimientos hacia las siguientes direcciones: adelante, atrás, izquierda, derecha y diagonales. Esto fue posible gracias al cálculo cinemático inverso aplicado a un conjunto de cuatro ruedas omnidireccionales.

Modelo cinemático inverso:

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} = \frac{1}{R}\begin{bmatrix}1 & 1 & -(l_1+l_2)\\ 1 & 1 & (l_1+l_2)\\ 1 & -1 & (l_1+l_2)\\ 1 & -1 & -(l_1+l_2)\end{bmatrix}\begin{bmatrix}V_x\\ V_y\\ W_z\end{bmatrix}$$

Modelo cinemático directo:

$$\begin{bmatrix}V_x\\ V_y\\ W_z\end{bmatrix} = \frac{R}{4}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & -1 & 1\\ \frac{1}{l_1+l_2} & \frac{1}{l_1+l_2} & \frac{-1}{l_1+l_2} & \frac{-1}{l_1+l_2}\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix}$$

Ejemplo de función para mover el robot hacia adelante:

```
1. def mover_adelante():
2.     velocidades = [50, -50, -50, 50]
3.     bus.write_i2c_block_data(0x34, 0x33, velocidades)
```

Desde una perspectiva pedagógica, esta estrategia favorece la comprensión del ciclo completo de desarrollo —programación, carga, ejecución y validación— en un solo entorno. Esto fortalece el aprendizaje práctico en proyectos de robótica y automatización, al enseñar simultáneamente conceptos de software y hardware.

Diagrama Lógico de Flujo de Comandos

• Recibir comando desde Firebase.
• Interpretar dirección del movimiento.
• Calcular velocidades con cinemática inversa.
• Enviar velocidades al driver vía I^2C.
• Leer encoders para validar el movimiento.
• Comparar posición actual con objetivo.
• Repetir hasta alcanzar destino.

Este control distribuido permitió realizar desplazamientos suaves, precisos y repetibles, asegurando una respuesta robusta frente a comandos de navegación en tiempo real.

Distribución de Energía y Regulación

El sistema RoboMesha fue diseñado con una arquitectura energética eficiente, capaz de alimentar de forma confiable tanto los actuadores de potencia como los módulos de control y sensado. La fuente principal del sistema es una batería de polímero de litio (Li-Po) de 11.1 V, seleccionada por su alta capacidad de descarga, densidad energética y portabilidad, ideal para aplicaciones robóticas móviles.

Dado que múltiples componentes del sistema requieren niveles de tensión inferiores, se integró un convertidor buck (reductor de voltaje) que permite transformar eficientemente la

salida de 11.1 V a 5 V. Este voltaje estabilizado se utiliza para alimentar sensores, microcontroladores y módulos de comunicación sin riesgo de sobrealimentación.

Componentes de alimentación

• Batería Li-Po 11.1V: Fuente principal de energía que alimenta los motores y actúa como base de la red de distribución energética del sistema.
• Módulo Buck Step-Down: Convertidor de voltaje que regula los 11.1 V a 5 V para dispositivos electrónicos de bajo consumo.
• Convertidor Boost (05KJ): Aunque no implementado en esta versión, se contempla el uso de un módulo elevador para alimentar ciertos periféricos con demandas de tensión superiores.

Esquema de protección del sistema

Actualmente, el sistema no cuenta con mecanismos de protección electrónica activos. No obstante, se contempla la integración de medidas que aumenten la confiabilidad y seguridad del sistema en futuras versiones.

Entre las protecciones planificadas se encuentran:

• Optoacopladores: Aislamiento galvánico entre las señales de control y los circuitos de potencia, protegiendo microcontroladores frente a ruidos eléctricos.
• Diodos flyback: Protección contra corrientes inversas generadas por inductancias de motores.
• Fusibles electrónicos o térmicos: Prevención de daños ante sobrecorrientes o cortocircuitos accidentales.
• TVS diodos: Protección contra sobretensiones transitorias que pudieran dañar componentes sensibles.
• Monitores de voltaje y corriente: Supervisión en tiempo real de las condiciones de operación, con posibilidad de ejecutar acciones de protección.

Estas medidas están orientadas a incrementar la durabilidad del sistema, mejorar la seguridad durante su operación y asegurar su funcionamiento en contextos educativos, donde la protección ante errores de conexión o sobrecarga es fundamental.

Flujograma lógico

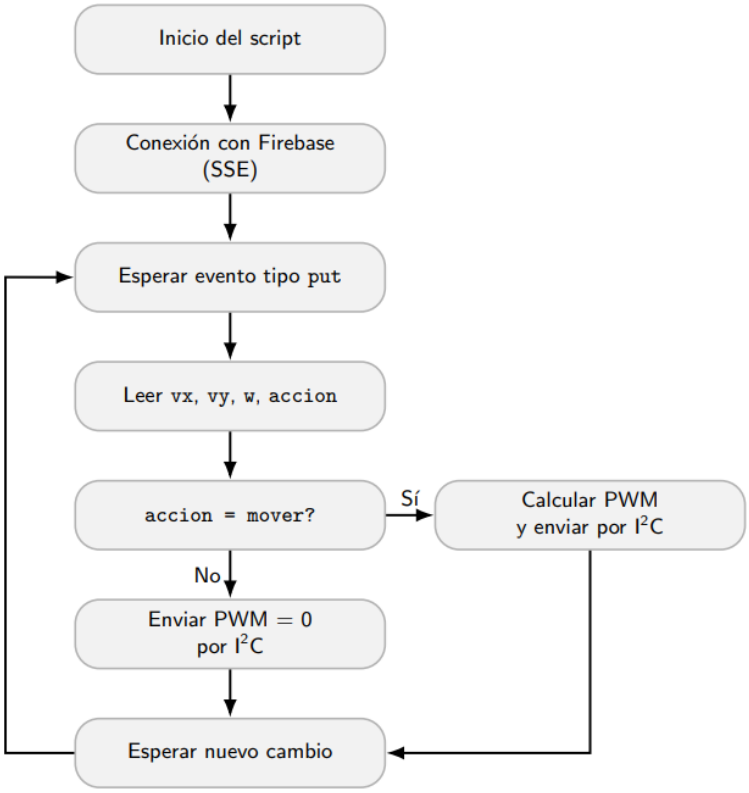

Figura 4. Flujograma lógico del calculo del PWM y envío por I2C.

3.-Interfaz:

La interfaz de RoboMesha fue diseñada para ofrecer un control remoto intuitivo y visualmente atractivo del robot, permitiendo a los usuarios interactuar con el sistema desde una aplicación de escritorio o navegador.

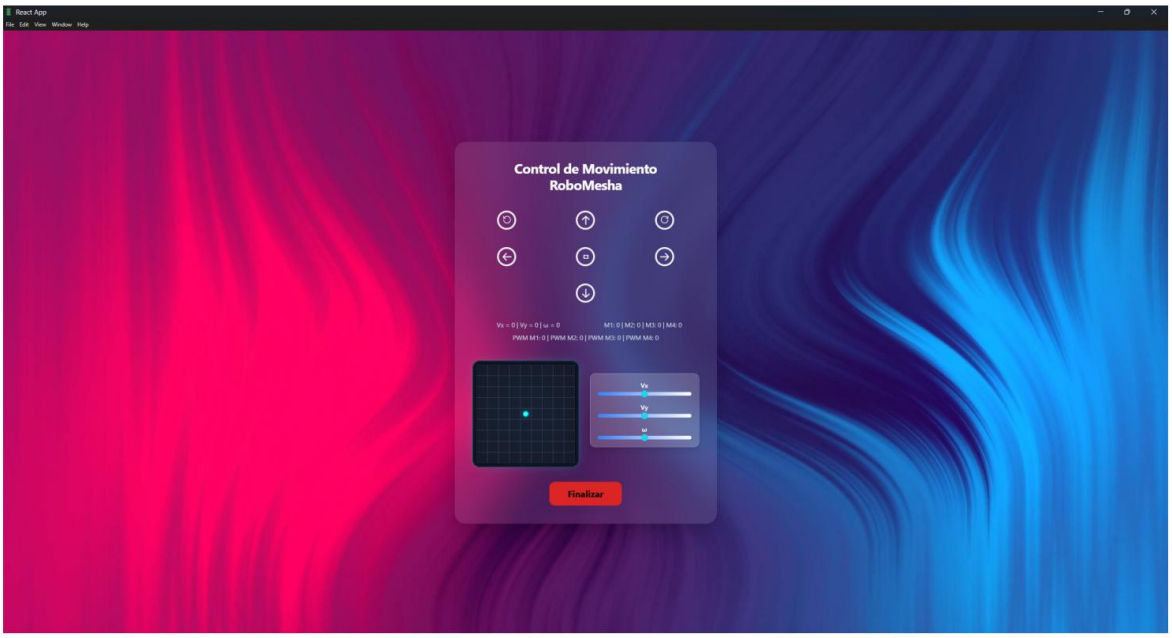

Figura 5. Captura de pantalla de la interfaz de RoboMesha.

La interfaz incluye los siguientes elementos:

- Botonera de control direccional (parte superior): permite enviar comandos de movimiento básicos (adelante, atrás, izquierda, derecha, rotación).
- Lectura de comandos enviados (centro): muestra en tiempo real los valores de velocidad lineal y angular: *vx*, *vy* y ω.
- Estado de los motores (M1–M4): presenta el valor PWM actual de cada motor, útil para diagnóstico y verificación.
- Vector gráfico (parte inferior izquierda): permite controlar el robot de manera más precisa usando el mouse o touchpad.
- Sliders de velocidad (parte inferior derecha): ofrecen control manual y separado de *vx*, *vy* y ω.
- Botón ”Finalizar”: detiene la ejecución del sistema de control.

Esta interfaz se conecta con Firebase para enviar los valores de movimiento, que son recibidos por la Raspberry Pi y aplicados en tiempo real al sistema de motores.

Comunicación con Firebase

Para establecer comunicación entre la Raspberry Pi y Firebase en el entorno de pruebas, se desarrolló el script firebaseconnect3.py. Este archivo permite la lectura continua de comandos desde la nube y la actualización de las velocidades de los motores en tiempo real mediante comunicación I2C.

El script utiliza la API REST de Firebase con eventos Server-Sent Events (SSE) para detectar cualquier cambio en la ruta /comandos/robomesha de la base de datos en tiempo real.

Información recibida desde Firebase:

- vx: velocidad lineal en eje X
- vy: velocidad lineal en eje Y
- w: velocidad angular (giro)
- accion: puede ser "mover" o "detener"

Comportamiento del sistema:

- Si accion = "mover", se calcula el PWM para cada rueda en función de *vx*, *vy* y ω usando la matriz de control omnidireccional W.
- Si accion = "detener", se envía un vector de ceros por I2C para detener los motores.
- El sistema evita redundancia comparando con el último comando ejecutado.

Control remoto por Firebase

El control remoto del robot RoboMesha se realiza mediante la lectura continua de datos almacenados en Firebase Realtime Database. A través de esta nube, el sistema recibe comandos que modifican la velocidad del robot en tiempo real. Esta estrategia permite separar la interfaz gráfica del hardware embebido, habilitando el control desde una aplicación web o móvil.

Comandos ejecutados remotamente:

- Modificación de valores de velocidad: *vx*, *vy*, *w*
- Definición de la acción: "mover" o "detener"

Esta interfaz permitió validar desde la etapa de pruebas que los comandos enviados desde la base de datos en la nube eran correctamente interpretados y aplicados al sistema de movimiento del robot mediante I2C.

Ejemplo de estructura de datos en Firebase:

```
{
  "comandos": {
    "robomesha": {
      "accion": "detener",
      "timestamp": 1745179144694,
      "vx": 0,
      "vy": 0,
      "w": 0
    }
  }
}
```

La estructura de datos utilizada para el control remoto se encuentra disponible públicamente en [4].

Fragmento de código en Python (uso en Raspberry Pi):

```
response = session.get(FIREBASE_STREAM_URL, stream=True,
    headers={"Accept": "text/event-stream"})
client = sseclient.SSEClient(response)

for event in client.events():
    if event.event == "put":
        try:
            payload = json.loads(event.data)
            path = payload.get("path", "")
            data = payload.get("data", {})

            if path == "/" and isinstance(data, dict):
                estado_actual.update(data)

```

Este código escucha en tiempo real los cambios en Firebase usando eventos tipo SSE. Cuando se detecta un cambio, actualiza los valores del diccionario estado_actual, que luego se convierte en PWM para el motor.

Limitaciones actuales:

- Pequeño retardo en la transmisión, típico del servicio gratuito de Firebase.
- No hay autenticación implementada (la base está abierta por simplicidad de pruebas).
- No se manejan múltiples robots simultáneamente ni historial de comandos.

Posibles mejoras:

- Implementar autenticación con claves de acceso por usuario.
- Añadir nodos independientes para múltiples unidades de RoboMesha.
- Integrar buffer de comandos con timestamps para evitar pérdida de paquetes.

Este control remoto ha permitido validar el sistema de navegación del robot sin requerir interacción física directa, siendo ideal para entornos educativos y pruebas remotas.

Repositorio y flujo del sistema

El repositorio oficial del proyecto RoboMesha está disponible públicamente en GitHub y puede consultarse en [5]. En él se integran todos los componentes esenciales para operar el sistema tanto desde el entorno web como desde la Raspberry Pi.

Contenido principal del repositorio:

- Interfaz web – desarrollada en React, permite el control remoto del robot desde un navegador.
- Scripts de conexión con Firebase – sincronizan los comandos enviados desde la nube.
- Lógica de control en Raspberry Pi – ejecuta las acciones físicas mediante motores y sensores.

Estructura del código fuente:

- public/ – Archivos estáticos de la interfaz web.
- src/ – Código fuente principal de la interfaz React.
- firebaseconnect3.py – Script que corre en la Raspberry Pi y ejecuta los comandos recibidos desde Firebase.

- requirements-pi.txt – Lista de dependencias necesarias para ejecutar el script en la Raspberry Pi.
- README.md – Documentación principal del repositorio: descripción del proyecto, instalación y uso.

Control PWM de Motores

Identificación de terminales

Modulación por ancho de pulso (PWM)

En el sistema RoboMesha, las señales de control para los motores no se generan directamente desde pines PWM de la Raspberry Pi. En su lugar, la Raspberry Pi calcula los valores de modulación por ancho de pulso (PWM) correspondientes a las velocidades deseadas (*vx*, *vy*, ω), y los transmite mediante el bus I2C hacia un controlador de motores dedicado.

La relación entre la velocidad deseada y el valor de PWM enviado se obtiene utilizando una matriz de transformación para plataformas omnidireccionales, que traduce los vectores de velocidad lineal y angular a velocidades individuales por rueda.
El resultado se normaliza y escala en el rango permitido de PWM.

Parámetros clave utilizados:

- Velocidad máxima asumida: 250 (unidad arbitraria)
- PWM máximo: 100
- Salida: lista de 4 valores PWM (uno por rueda), en el rango -100 a 100

Fragmento representativo del código:

```
 1. def calcular_pwm(vx, vy, omega):
 2.     v = np.array([vx, vy, omega])
 3.     velocidades = np.dot(W, v)
 4.     factor_escala = np.max(np.abs(velocidades)) / 250 if np.max(np.abs(velocidades)) > 250
else 1
 5.     if factor_escala > 1:
 6.         velocidades /= factor_escala
 7.     velocidades[1] *= -1
 8.     velocidades[2] *= -1
 9.     pwm_values = np.clip((velocidades / V_MAX) * PWM_MAX, -PWM_MAX, PWM_MAX)
10.     return [int(p) for p in pwm_values]
11.
12. bus.write_i2c_block_data(DIRECCION_MOTORES, REG_VELOCIDAD_FIJA, pwm_values)
13.
```

Relación con la velocidad real:

Los valores PWM determinan la potencia aplicada a cada motor. Aunque no hay una medición directa de RPM o velocidad lineal, se ha observado que los motores responden proporcionalmente en velocidad al valor PWM, especialmente en el rango de 30–100. En valores bajos (debajo de 20), el par no es suficiente para vencer la fricción estática, por lo que se considera un umbral mínimo para el movimiento efectivo.

Esta estrategia permite un control ágil sin requerir generación de señal PWM por hardware, delegando la modulación al controlador de motores físico.

Para validar el movimiento, se realizó el cálculo cinemático inverso utilizando matrices de transformación, permitiendo desplazamientos precisos hacia adelante, atrás y diagonales.

Limitaciones, Retos y Adaptaciones

Durante el desarrollo del proyecto RoboMesha, surgieron diversos obstáculos que impactaron tanto el diseño como la implementación del sistema. No obstante, el equipo adoptó estrategias adaptativas que permitieron continuar con el avance del proyecto sin comprometer sus objetivos esenciales.

Adquisición de componentes

Uno de los retos principales fue la dificultad para adquirir ciertos componentes clave. La combinación de largos tiempos de entrega, falta de disponibilidad local y complicaciones logísticas derivó en la imposibilidad de integrar algunos módulos electrónicos originalmente planeados.

Ante esta situación, se optó por reutilizar componentes previamente disponibles, tales como las llantas, el puente H, una unidad Raspberry Pi y fuentes de alimentación auxiliares. Esta estrategia de contingencia permitió continuar con el armado del prototipo y validar su funcionalidad sin interrupciones críticas.

Interacción con otras áreas

La colaboración con otras áreas del proyecto fue fundamental, en particular con el área de control, debido a la estrecha relación funcional entre ambas. Se estableció una comunicación continua que facilitó la integración efectiva entre el sistema electrónico y la lógica de control del robot.

Esta sinergia permitió resolver problemas técnicos de forma conjunta, alinear decisiones de diseño y asegurar que la arquitectura electrónica respondiera correctamente a los requerimientos operativos definidos por las demás áreas. Esta interacción fue clave para consolidar la etapa funcional del proyecto y sentar las bases para futuras iteraciones con mayor integración y robustez.

Recomendaciones y Continuidad

Componentes sugeridos para futuras iteraciones

Durante el desarrollo del sistema, uno de los principales retos fue la falta de entrega oportuna de componentes electrónicos por parte del área de adquisiciones, lo cual afectó el cumplimiento de los objetivos establecidos. Se recomienda mejorar la comunicación y la coordinación con esta área, así como establecer procesos más ágiles para la gestión y autorización de insumos críticos.

Para futuras versiones del sistema, se sugiere considerar el uso de una Raspberry Pi 5, la cual ofrece notables mejoras en potencia de procesamiento, eficiencia térmica y disponibilidad de pines GPIO, en comparación con la versión 4 utilizada en este proyecto. Estas capacidades facilitarían la integración de sensores avanzados, como módulos LiDAR, y permitirían el despliegue de algoritmos de inteligencia artificial para navegación o detección de entornos.

Asimismo, se recomienda retomar la integración de los módulos originalmente planeados, entre los que destacan:

- Sensores de temperatura y humedad.
- Sensores de corriente y voltaje.
- Sensores ultrasónicos y de peso.

- Optoacopladores para aislamiento eléctrico.
- Pantalla táctil para interacción local con el usuario.

La incorporación de estos módulos incrementaría significativamente la funcionalidad, robustez y valor pedagógico del sistema RoboMesha.

Escalabilidad del sistema

Para asegurar la adaptabilidad del sistema a nuevos entornos o requerimientos, se propone una estrategia de escalabilidad modular, basada en la expansión controlada y documentada de los periféricos. Entre las recomendaciones se incluyen:

- Utilizar expansores de pines GPIO (como el MCP23017) o multiplexores I²C, permitiendo conectar múltiples dispositivos sin saturar el bus principal.
- Establecer una arquitectura basada en buses de comunicación estándar (UART, SPI, I²C), con direccionamiento bien definido para evitar conflictos entre módulos.
- Diseñar y utilizar placas de expansión o PCBs personalizadas que permitan ordenar mejor los componentes, reducir el cableado y facilitar el mantenimiento.
- Incorporar soluciones de comunicación inalámbrica segura, como módulos WiFi o Bluetooth configurados independientemente de la red institucional, especialmente útil en entornos con restricciones de acceso como el ITESO.

Recomendaciones prácticas

Durante la implementación y pruebas se identificaron varias buenas prácticas que pueden mejorar la estabilidad y confiabilidad del sistema. A continuación, se enlistan recomendaciones clave:

- Calidad de soldaduras: Verificar visual y eléctricamente las conexiones, en especial en pines de comunicación y alimentación. Una soldadura fría o deficiente puede provocar errores intermitentes difíciles de diagnosticar.
- Disipación térmica: Instalar disipadores o ventiladores sobre la Raspberry Pi y otros componentes que generen calor, especialmente si el sistema opera por tiempos prolongados o ejecuta tareas de computación intensiva.

- Conectores de batería: Usar conectores firmes y de buena calidad, asegurando la correcta polaridad y sujeción. Desconexiones accidentales pueden dañar circuitos o corromper la tarjeta SD.
- Conectividad en red institucional: Se detectaron problemas de conexión en la red WiFi del ITESO debido a restricciones como portales cautivos. Se recomienda usar una red privada o conexión Ethernet directa durante el arranque inicial para asegurar conectividad estable con Firebase u otros servicios.

Estas acciones contribuirán a mejorar la experiencia de desarrollo, facilitar la integración de futuras funciones y garantizar un funcionamiento seguro en contextos educativos y de prototipado.

Documentación Técnica Complementaria

Enlace a repositorio

El proyecto cuenta con un repositorio de RoboMesha disponible en el siguiente enlace [5], el cual incluye tanto la interfaz gráfica como los scripts de control necesarios para la operación del robot. Este enfoque permite separar claramente el entorno de control (ejecutado en la laptop o computadora del usuario) y el entorno embebido (ejecutado en la Raspberry Pi).

Entorno de control (laptop / computadora del usuario)

La interfaz web, desarrollada en React, actúa como panel de control visual del robot. Esta aplicación puede ejecutarse en modo de desarrollo o compilarse como ejecutable para su instalación local en Windows.

Ejecución en modo desarrollo:

```
1. git clone https://github.com/Aaronsep/RoboMesha.git
2. cd RoboMesha
3. npm install
4. npm run dev
```

Esto inicia la aplicación web en modo local, permitiendo interactuar con la base de datos de Firebase y enviar comandos de movimiento (vx, vy, ω) al robot.

También es posible empaquetarla como una aplicación de escritorio para su ejecución sin necesidad de terminal.

Entorno embebido (Raspberry Pi)

En la Raspberry Pi no es necesario instalar la interfaz. Solo se ejecuta un script llamado firebaseconnect3.py, que escucha los comandos en tiempo real desde Firebase y aplica las señales de control a los motores mediante I2C.

Ejecución del script:

```
1. python3 firebaseconnect3.py
2.
```

Instalación de dependencias:

```
1. pip install -r requirements_pi.txt
```

Este enfoque modular permite que la interfaz gráfica evolucione de forma independiente al sistema embebido, y facilita la depuración, mantenimiento y escalabilidad del sistema.

Evidencias de los Procesos, Resultados y Servicios Alcanzados

Se efectuaron diversas pruebas funcionales con el prototipo físico del robot, validando la lectura digital de los encoders y la comunicación estable con el driver vía I2C. Los desplazamientos realizados confirmaron el comportamiento omnidireccional del sistema.

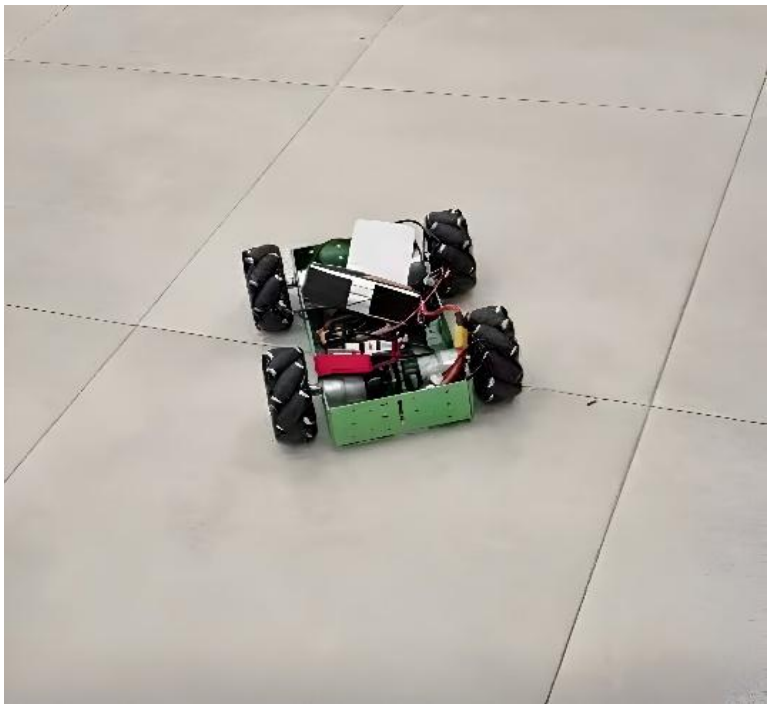

Figura 6. Prototipo físico del robot en movimiento.

Se creó una interfaz gráfica conectada a Firebase para el control remoto del robot. Esta interfaz visual facilita el envío de comandos específicos, mostrando en tiempo real las velocidades asignadas a cada motor.

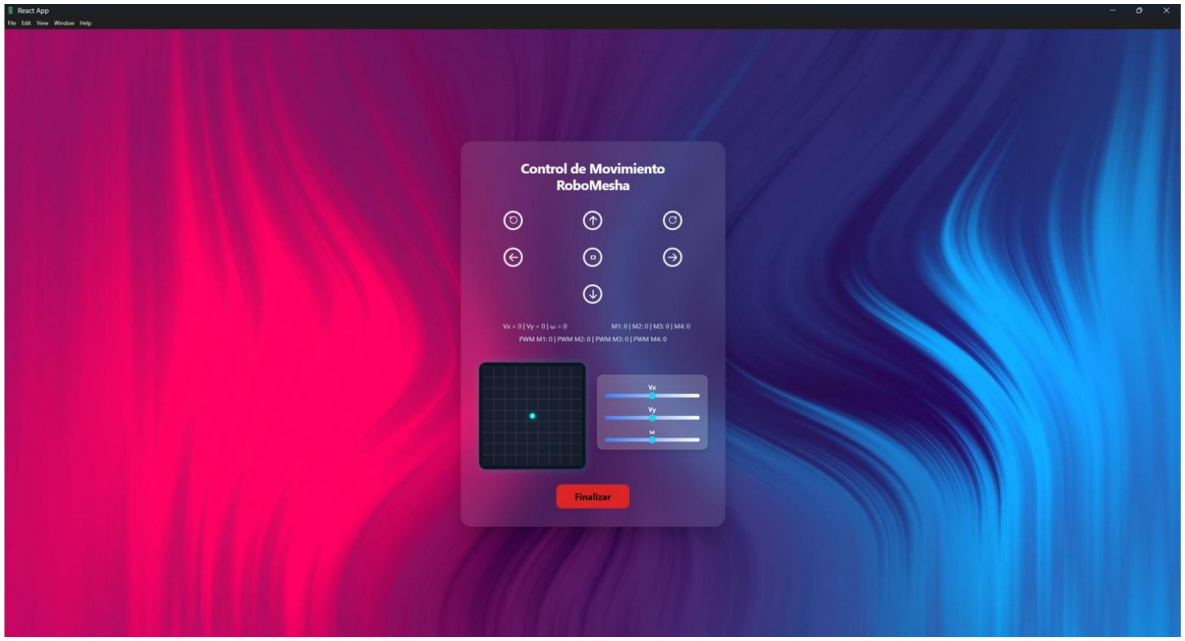

Figura 7. Segunda captura de pantalla de la interfaz de RoboMesha.

Además, agregamos una línea de tiempo de implementación para mostrar la evolución del desarrollo electrónico a lo largo del semestre:

Componente	Función	Voltaje de Operación	Protocolo de Comunicación	Justificación
Raspberry Pi 4	Unidad de procesamiento	5V	UART / I2C	Plataforma abierta y versátil
Arduino Nano	Microcontrolador auxiliar	5V	I2C	Control de periféricos y actuadores
Motores JGB37-520	Tracción del robot	12V	PWM	Disponibilidad local y compatibilidad
Encoders de efecto Hall	Lectura de velocidad	5V	Digital	Medición precisa de RPM

Tabla 2. Línea de tiempo de la evolución del desarrollo tecnológico.

1.6. Valoración de productos, resultados e impactos

La experiencia PAP con RoboMesha permitió desarrollar un sistema electrónico funcional enfocado en sentar las bases de un robot omnidireccional orientado a contextos educativos vulnerables. Aunque la visión original contemplaba una integración total de sensores y actuadores con fines académicos, esta parte del proyecto no pudo implementarse por limitaciones logística, las cuales lamentamos que sucedan, por el tipo de institución que es el ITESO. Aun así, se logró establecer una plataforma BASE robusta sobre la cual se podrá desarrollar la dimensión educativa en futuros semestres.

El sistema de tracción fue validado con éxito mediante pruebas funcionales que demostraron su capacidad para ejecutar movimientos suaves y precisos. La lectura continua de encoders y el control por I2C ofrecieron retroalimentación en tiempo real, mientras que la interfaz gráfica conectada a Firebase permitió controlar el robot remotamente desde cualquier dispositivo.

Entre los productos generados se encuentran: un subsistema estable de propulsión con control PID, código fuente en Python y C++, documentación técnica en formato manual, diagramas eléctricos, una interfaz gráfica en React, y un repositorio público en GitHub. Estos recursos

aportan valor al ecosistema educativo, al ser fácilmente replicables, escalables y adaptables a distintos entornos académicos.

Como impacto educativo, el proyecto contribuye a democratizar el acceso a la robótica educativa en escuelas de escasos recursos, brindando una alternativa práctica y accesible para fomentar habilidades STEAM. Se recomienda que futuras etapas del PAP retomen la implementación de la parte educativa, aprovechando la infraestructura ya construida y evaluando nuevas posibilidades de financiamiento y gestión logística.

Limitaciones técnicas y estrategias de mitigación

Durante la ejecución del proyecto se presentaron diversas limitaciones técnicas, principalmente asociadas a la adquisición tardía o inexistente de componentes electrónicos solicitados a través del área de compras institucional. Esta situación obligó al equipo a replantear parte de la arquitectura electrónica, priorizando el uso de materiales previamente disponibles en el laboratorio de mecatrónica del ITESO.

Lecciones aprendidas:

- La planificación de adquisiciones debe realizarse con mayor antelación y con rutas alternativas en caso de retrasos logísticos.
- Es importante documentar continuamente el estado de los materiales y validar opciones de sustitución técnica con equivalencias funcionales.

Estrategias de mitigación implementadas:

- Reutilización de motores, placas y sensores del inventario institucional, validando su funcionamiento antes de integrarlos.
- Priorización de pruebas funcionales por encima de integración estética o modular para asegurar validación de concepto.
- Flexibilización en el diseño del sistema educativo para que su integración pueda ser completada en futuras fases del proyecto.

1.7. Bibliografía y otros recursos

HIWONDER. (2024). *Documentación técnica del driver de motores*. Recuperado en 2024, de https://drive.google.com/drive/folders/1ZIbMQo2R2YOgqYN3d9nTgIqJnxlGbR8m

Mercado Libre. (2025). *Convertidor DC-DC step-down buck LM2596*. Recuperado el 10 de febrero de 2025, de https://articulo.mercadolibre.com.mx/MLM-1479860262-convertidor-dc-dc-step-down-buck-JM

Talos Electronics. (2025). *Convertidor boost 05kj - Fuente de voltaje ajustable DC-DC*. Recuperado el 10 de febrero de 2025, de https://www.taloselectronics.com/products/fuente-de-voltaje-ajustable-dc-dc-de-10-32v-a-12-35v-vcd-6a-150w

Sepúlveda, A. (2025). *Comandos remotos de RoboMesha* [Base de datos pública]. Firebase. Recuperado en 2025, de https://robomesha-default-rtdb.firebaseio.com/comandos/robomesha.json

Sepúlveda, A. (2025). *RoboMesha – Plataforma robótica educativa*. GitHub. Recuperado en abril de 2025, de https://github.com/Aaronsep/RoboMesha

1.8. Anexos generales

Se invita a revisar el producto del Manual de Área: Electrónica RoboMesha, adjunto como anexo, el cual documenta de manera detallada el funcionamiento del sistema electrónico del prototipo. Este manual es clave para la comprensión integral del proyecto, ya que recopila información técnica, diagramas, justificaciones de diseño y procedimientos que respaldan el desarrollo del sistema propuesto.

2. Productos

Nombre y código del PAP:	4D08 Desarrollo tecnológico para la sustentabilidad ambiental, energética y alimentaria
Nombre del subproyecto:	RoboMesha etapa primavera 2025
Nombre del producto:	Manual de área: Electrónica
Descripción (qué es, para quién se realizó y para qué es):	Este documento funciona como manual técnico para los integrantes actuales y futuros del Proyecto de Aplicación Profesional (PAP) relacionado con el desarrollo del sistema robótico RoboMesha. Su objetivo es documentar de manera clara y estructurada los avances alcanzados hasta el momento, así como proporcionar las bases necesarias para continuar con el desarrollo e implementación de las funcionalidades pendientes. El documento incluye detalles técnicos sobre sensores, actuadores, arquitectura de control, modelos de movimiento y ejemplos de código utilizados, con la finalidad de facilitar la comprensión y continuidad del proyecto en ciclos posteriores.
Autores:	Ing. En Mecatrónica. Fernando Vidal Luna Jaime Ing. En Mecatrónica. Aaron Homero Sepúlveda Salcedo Ing. En Mecatrónica. María Alhelí Pérez Rosas Ing. En Mecatrónica. Alexa Torres Rochín
Estado:	Final.

Tabla 3. Tabla del producto del Manual de área: Electrónica.

Nombre y código del PAP:	4D08 Desarrollo tecnológico para la sustentabilidad ambiental, energética y alimentaria
Nombre del subproyecto:	RoboMesha etapa primavera 2025
Nombre del producto:	Cartel Robomesha área: Electrónica
Descripción (qué es, para quién se realizó y para qué es):	Como complemento visual, se diseñó un cartel explicativo del proyecto que será utilizado durante su exposición en ferias y eventos académicos. Este cartel sintetiza gráficamente los objetivos, componentes, resultados y el impacto social de RoboMesha, facilitando la comunicación efectiva del proyecto a públicos diversos. Su propósito es generar interés, fomentar el diálogo con visitantes y posibles aliados, y destacar el valor educativo de la plataforma en contextos escolares de alta vulnerabilidad.
Autores:	Ing. En Mecatrónica. Fernando Vidal Luna Jaime Ing. En Mecatrónica. Aaron Homero Sepúlveda Salcedo Ing. En Mecatrónica. María Alhelí Pérez Rosas Ing. En Mecatrónica. Alexa Torres Rochín
Estado:	Final.

Tabla 4. Tabla del producto del Cartel RoboMesha área: Electrónica.

Producto	Función principal	Autor responsable(s)	Estado
Manual de entrega de turno	Documentar la arquitectura electrónica y el uso del sistema	María Alhelí Pérez Rosas	Final
RPAP Electrónica	Documentar el proceso de actividades realizadas durante el semestre	María Alhelí Pérez Rosas, Fernando Luna	Final
Código fuente en Python y C++	Control de motores, lectura de encoders y comunicación	Aaron Sepúlveda, Fernando Luna	Final
Interfaz gráfica en React	Control remoto del robot vía Firebase	Aarón Sepúlveda	Prototipo
Diagrama esquemático electrónico	Representación técnica de conexiones y arquitectura	Fernando Luna	Final
Repositorio público en GitHub	Consolidación y acceso abierto al proyecto	Aaron Sepúlveda	Final
Cartel académico del proyecto	Difusión visual para ferias y eventos académicos	Aarón Sepúlveda, Alexa Torres	Final

Tabla 5. Tabla del resumen de productos entregados y trazabilidad.

3. Reflexión crítica y ética de la experiencia

El RPAP tiene también como propósito documentar la reflexión sobre los aprendizajes en sus múltiples dimensiones, las implicaciones éticas y los aportes sociales del proyecto para compartir una comprensión crítica y amplia de las problemáticas en las que se intervino.

3.1 Sensibilización ante las realidades

Durante el desarrollo del proyecto, fue evidente para nosotros el contraste entre los recursos tecnológicos disponibles en instituciones como el ITESO y la falta de acceso a herramientas básicas en muchas escuelas públicas del país. Esta diferencia nos llevó a reflexionar

profundamente sobre las desigualdades estructurales que afectan a miles de estudiantes y reforzó nuestro compromiso por desarrollar soluciones tecnológicas con un propósito social.

A lo largo del proceso, nos posicionamos frente a esta realidad con una actitud de empatía y responsabilidad. Conectar nuestras habilidades técnicas con una necesidad real nos permitió resignificar el papel del ingeniero en la sociedad. Ver la posibilidad de que nuestro trabajo facilite el aprendizaje de niñas y niños que, como muchos de nosotros, no tuvieron acceso temprano a la tecnología, nos motivó a seguir adelante pese a las dificultades logísticas.

Esta experiencia reforzó nuestra convicción de que la ingeniería debe estar al servicio de la sociedad. Reconocimos que nuestras decisiones como futuros profesionales tienen implicaciones éticas importantes y que incluso un proyecto académico puede convertirse en un vehículo de transformación. El PAP nos enseñó que no basta con saber hacer, sino que debemos aprender a hacerlo con propósito y conciencia social.

3.2 Aprendizajes logrados

El Proyecto de Aplicación Profesional RoboMesha no solo permitió desarrollar competencias técnicas, sino que también impulsó una reflexión profunda sobre el papel social de la ingeniería. Las decisiones técnicas tomadas durante el desarrollo estuvieron guiadas por principios de accesibilidad, equidad tecnológica y sostenibilidad a largo plazo.

Un ejemplo concreto fue la elección deliberada de utilizar componentes de bajo costo y fácil acceso, como la Raspberry Pi y el Arduino Nano, priorizando plataformas abiertas y documentadas ampliamente. Esta decisión no solo respondió a la disponibilidad institucional, sino que permitió garantizar que el sistema pueda ser replicado en otras instituciones con recursos limitados.Asimismo, se diseñó una arquitectura modular que permite escalar o simplificar el sistema según el contexto educativo donde se implemente. Esta flexibilidad busca democratizar el acceso a la tecnología sin comprometer la funcionalidad del prototipo.

La integración con Firebase para el control remoto también se fundamentó en su gratuidad para pequeños volúmenes de datos, su accesibilidad desde múltiples dispositivos y su bajo requerimiento de infraestructura adicional. De esta forma, se fomentó una solución digital inclusiva, adaptable y sostenible para ambientes escolares diversos.

En conjunto, estas decisiones reflejan una postura ética clara: el desarrollo tecnológico debe responder a las necesidades sociales, y los ingenieros tenemos la responsabilidad de construir soluciones que no solo funcionen, sino que también sean justas, replicables y transformadoras.