

# **Especificaciones técnicas**

## Suspensión Activa Electromagnética (SAE)

Sebastián Sánchez, Martín Calderón, Juan Felipe Hernández, Diego Ramírez

1 de Diciembre 2025

# Contents

<b>1</b>	<b>Introducción</b>	<b>2</b>
1.1	Propósito . . . . .	2
1.2	Público Objetivo . . . . .	2
1.3	Alcance del Proyecto . . . . .	2
<b>2</b>	<b>Descripción del Producto</b>	<b>3</b>
2.1	Descripción del Hardware: Componentes . . . . .	3
2.1.1	Sensores . . . . .	3
2.1.2	Actuadores . . . . .	4
2.1.3	Potencia . . . . .	5
2.2	Descripción del Hardware: PCB . . . . .	5
2.2.1	Dimensiones . . . . .	6
2.2.2	Capas . . . . .	6
2.2.3	Esquemático de conexiones . . . . .	9
2.2.4	Reglas generales . . . . .	10
2.2.5	Etapas del diseño . . . . .	12
2.3	Descripción del Software . . . . .	15
2.3.1	Arquitectura General . . . . .	15
2.3.2	Componentes Principales del Software . . . . .	16
2.3.3	Interacción entre Paquetes . . . . .	17
2.4	Entorno de Desarrollo y Herramientas de Software . . . . .	17
2.5	Restricciones de diseño e implementación . . . . .	18
2.5.1	Restricciones de Potencia y Alimentación <b>V</b> . . . . .	18
2.5.2	Restricciones de Diseño y Fabricación <b>PCB</b> . . . . .	18
2.5.3	Restricciones de Software y Rendimiento <b>ROS2</b> . . . . .	18
2.5.4	Restricciones Ambientales y de Calidad <b>Δ</b> . . . . .	19
<b>3</b>	<b>External Interface Requirements</b>	<b>20</b>
3.1	Interfaz de usuario GUI . . . . .	20

<b>4 Funcionalidades del sistema</b>	<b>22</b>
4.1 Seguimiento de línea . . . . .	22
4.2 Amortiguamiento electromagnético activo . . . . .	23

# Chapter 1

## Introducción

### 1.1 Propósito

El proyecto tiene como objetivo diseñar un sistema electrónico y su sistema de control para una Suspensión Activa Electromagnética (SAE), la cual disipa vibraciones y movimientos transformando energía mecánica en eléctrica. El desarrollo del proyecto consiste en diseñar el mecanismo para el sensado, procesamiento de señales y el control de los actuadores.

### 1.2 Público Objetivo

Personal técnico con conocimientos de control, diseño de sistemas electrónicos e ingenieros especializados en la amortiguación. La propuesta reducirá el costo de adquirir un sistema de amortiguación activo en vehículos particulares. De esta forma más personas podrían tener acceso a esta tecnología.

### 1.3 Alcance del Proyecto

El desarrollo se dividió en dos partes: diseño de PCB y sensórica y motores. En la primera, se buscó integrar en un espacio compacto todos los módulos, sensores y componentes necesarios para la interacción con el microcontrolador, cumpliendo normas de diseño. En la segunda, se realizaron pruebas individuales de cada componente y posteriormente la integración del sistema completo, incluyendo alimentación, sensores, actuadores y motores. El funcionamiento se basa en medir el desplazamiento vertical de la suspensión de un vehículo, aplicar un sistema de control que determine la fuerza requerida para contrarrestar la compresión de los resortes y accionar los solenoides de manera precisa. Todo el proceso opera a velocidad constante mediante una interfaz que permite monitorear el rendimiento del sistema en tiempo real. El alcance del sistema electrónico es diseñar la PCB y programar los componentes necesarios para correr el sistema de control sobre esta arquitectura electrónica.

## Chapter 2

# Descripción del Producto

## 2.1 Descripción del Hardware: Componentes

El sistema electrónico del proyecto fue diseñado para funcionar sobre un Single Board Computer. Específicamente una Raspberry Pi 5. La Raspberry Pi utiliza una alimentación de 5V a 5A. Equivale al consumo máximo que puede soportar la placa.

### 2.1.1 Sensores

#### TCRT 5000

<b>Especificaciones TRCT 5000 de 5 Canales</b>	
Distancia de sensado	12mm
Tensión de alimentación	5V
Interfaz Digital	I2C
Disipación de potencia	200mW
Dimensiones	95mm x 20mm x 10mm
Peso	9g
Temperatura de funcionamiento	10°C-30°C

Table 2.1: Especificaciones TCRT 5000

#### MPU 6050

<b>Especificaciones MPU 6050</b>	
Grados de Libertad	6
Tensión de alimentación	5V
Interfaz Digital	I2C
Disipación de potencia	15mW
Dimensiones	20mm x 16mm x 3mm
Peso	3g
Temperatura de funcionamiento	-30°C- 80°C

Table 2.2: Especificaciones MPU6050

#### ADS 1115

<b>Especificaciones MPU 6050</b>	
Grados de Libertad	6
Tensión de alimentación	5V
Interfaz Digital	I2C
Disipación de potencia	15mW
Dimensiones	20mm x 16mm x 3mm
Peso	3g
Temperatura de funcionamiento	-30°C- 80°C

Table 2.3: Especificaciones ADS 1115

### Convertidor DC-DC XL4016 Buck

<b>Especificaciones XL4016 Buck</b>	
Tensión de Entrada	DC 7-35V
Tensión de Salida	0.8-28V
Corriente de Salida	8A
Potencia de Salida	300 W
Eficiencia	95%
Frecuencia de operación	300KHz
Dimensiones	65mm x 47mm x 22mm
Peso	50g
Temperatura de funcionamiento	-40°C- 85°C

Table 2.4: Especificaciones Convertidor DC-DC

### Resistencia variable de desplazamiento

<b>Especificaciones resistencia de desplazamiento</b>	
Resistencia	0 – 5kΩ
Temperatura de funcionamiento	-40°C- 85°C

Table 2.5: Especificaciones Resistencia variable

### 2.1.2 Actuadores

#### Solenoide JF-0530B

<b>Especificaciones Solenoide</b>	
Voltaje	12V
Corriente Nominal	300mA
Desplazamiento	2cm
Fuerza	5N
Temperatura de funcionamiento	-10°C- 50°C

Table 2.6: Especificaciones Resistencia variable

#### Motores JGB37-520 178RPM con Encoder

<b>Especificaciones Motor JGB37-520 178RPM</b>	
Potencia	7-15W
Voltaje	12V
Corriente	0.45-1A
Velocidad	6-1590RPM
Torque máximo	2KG.CM
Eje de salida	21.5mm
Dimensiones	70mm x 60mm x 20mm
Peso	196g
Temperatura de funcionamiento	-20°C- 60°C

Table 2.7: Especificaciones Motor JGB37-520

### 2.1.3 Potencia

#### Convertidor DC-DC XL4016 Buck

<b>Especificaciones XL4016 Buck</b>	
Tensión de Entrada	DC 7-35V
Tensión de Salida	0.8-28V
Corriente de Salida	8A
Potencia de Salida	300 W
Eficiencia	95%
Frecuencia de operación	300KHz
Dimensiones	65mm x 47mm x 22mm
Peso	50g
Temperatura de funcionamiento	-40°C- 85°C

Table 2.8: Especificaciones Convertidor DC-DC

#### MÓDULO DE CARGA Y PROTECCIÓN BMS HW-296 16.8V 4S Li-Ion18650

<b>Especificaciones BMS HW-296</b>	
Tipo de Batería	Li-Ion 18650
Configuración	4S (4 celdas en serie)
Voltaje Nominal	14.8V
Voltaje máximo	16.8V
Protección	Sobrecarga, sobredescarga, cortocircuito
Corriente de carga	10A(recomendado), 20A(máximo)
Dimensiones	70mm x 60mm x 20mm
Peso	6g
Temperatura de funcionamiento	-20°C- 60°C

Table 2.9: Especificaciones BMS HW-296

## 2.2 Descripción del Hardware: PCB

A rasgos generales lo que busca el diseño del módulo electrónico es condensar todas las conexiones de señales de los distintos sensores, la etapa de potencia para alimentación los periféricos (motores, solenoides, etc), ademas del envío y recepción de señales por medio de la conexión al microcontrolador. Como parte del problema de optimización es el volumen ocupado en el montaje para el vehículo, se implemento un lógica modular en la que el microcontrolador se puede remover fácilmente a la PCB. Estos dos sistemas se sobreapan uno en el otro con el fin de ocupar el menor volumen posible.

A continuación, en las siguientes secciones se encuentran parámetros generales del sistema para su correcta operación.

### 2.2.1 Dimensiones

Todo el sistema electrónico de la PCB está embebido dentro de un espacio cuadrado de 101.5mm X 101.5mm, este es capaz de albergar todas las etapas y componentes.

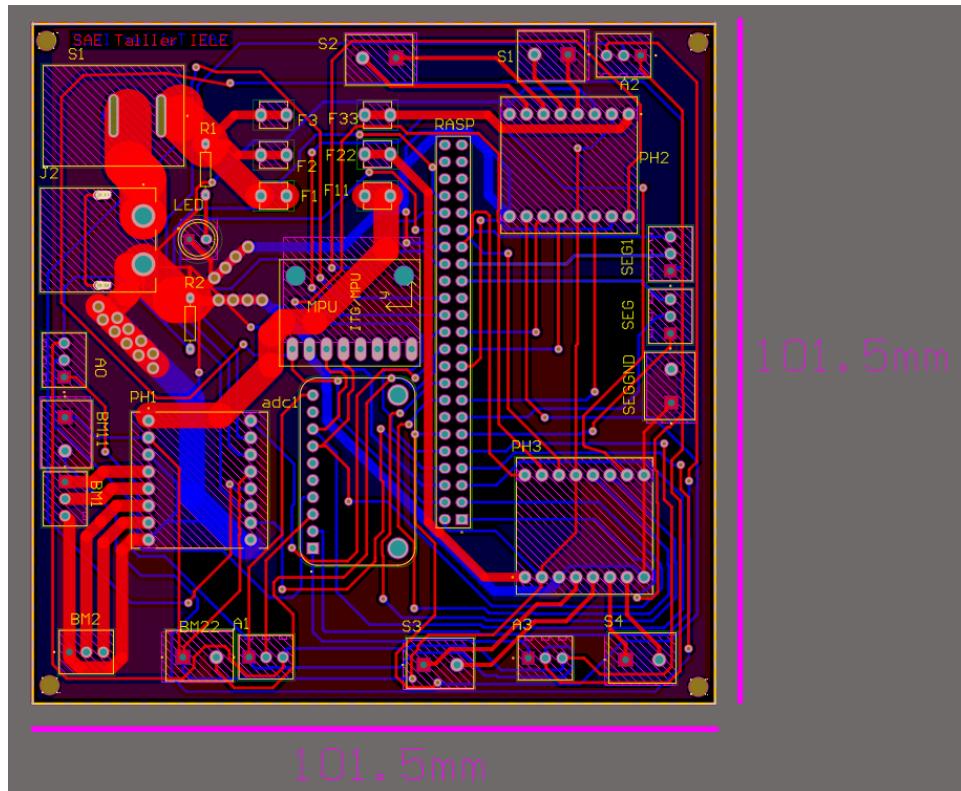


Figure 2.1: Dimensiones PCB

### 2.2.2 Capas

El diseño del circuito impreso se estructuró utilizando una tarjeta de dos capas, en la cual se distribuyeron de manera organizada las pistas, las zonas de blindaje y las áreas destinadas a la colocación de componentes. La correcta definición de cada capa permite garantizar un ruteo eficiente, controlar el comportamiento electromagnético del sistema y asegurar una adecuada manufacturabilidad.

La capa superior incluye la leyenda de componentes (Top Overlay), donde se ubican las referencias serigráficas que permiten identificar la posición y orientación de cada elemento durante el ensamblaje. Adicionalmente, esta capa contiene el plano de cobre correspondiente al Copper Signal Top Layer, destinado principalmente al ruteo de señales críticas y a la distribución de líneas de control.

Encima del plano conductor se encuentra la capa de máscara antisoldante (SolderMask Top Layer), la cual protege el cobre expuesto y evita cortocircuitos durante el proceso de soldadura. La máscara delimita las zonas donde se requiere contacto eléctrico, tales como pads y puntos de test.

De manera complementaria, la parte inferior de la tarjeta está compuesta por la capa de cobre denominada Copper Signal Bottom Layer, utilizada tanto para el retorno de corrientes como para el ruteo de señales secundarias o de menor criticidad. Esta capa permite distribuir el ruteo en dos niveles, reduciendo cruces y optimizando la longitud total de las pistas. Similar a la capa superior, se incluye la respectiva máscara

antisoldante (SolderMask Bottom Layer), que proporciona protección mecánica y aislamiento eléctrico al cobre inferior.

La selección de estas capas y su organización responde a criterios de integridad de señal, disipación térmica y compatibilidad con los procesos de fabricación establecidos. Las figuras correspondientes presentan la disposición visual de cada una de las capas descritas, facilitando su interpretación dentro del diseño general de la PCB.

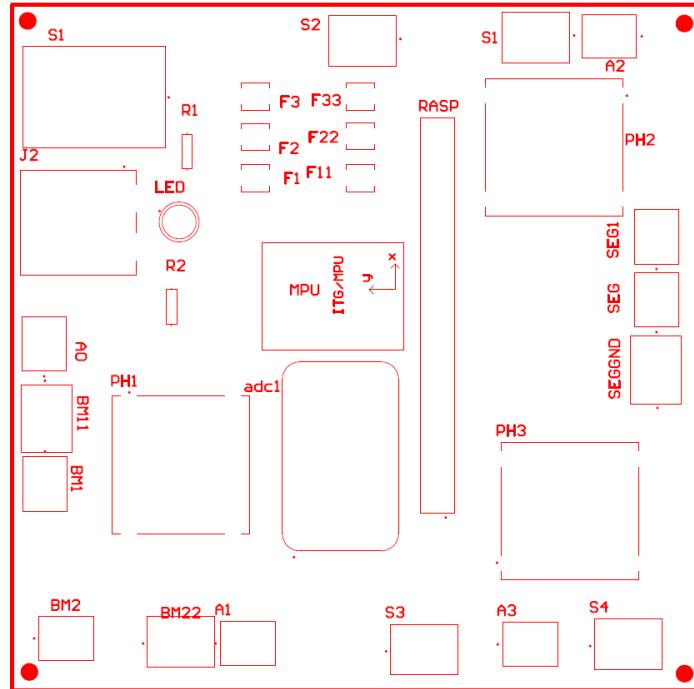


Figure 2.2: Legend Top Overlay

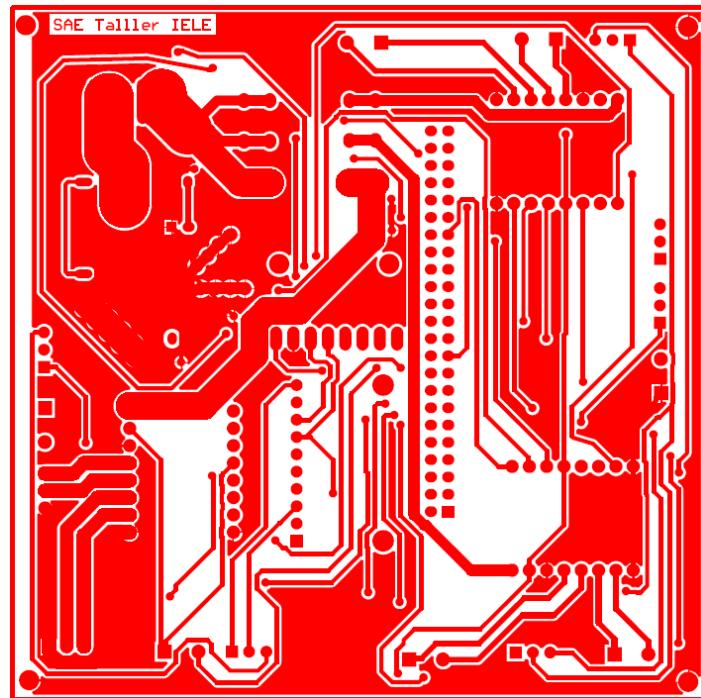


Figure 2.3: Copper Signal Top Layer

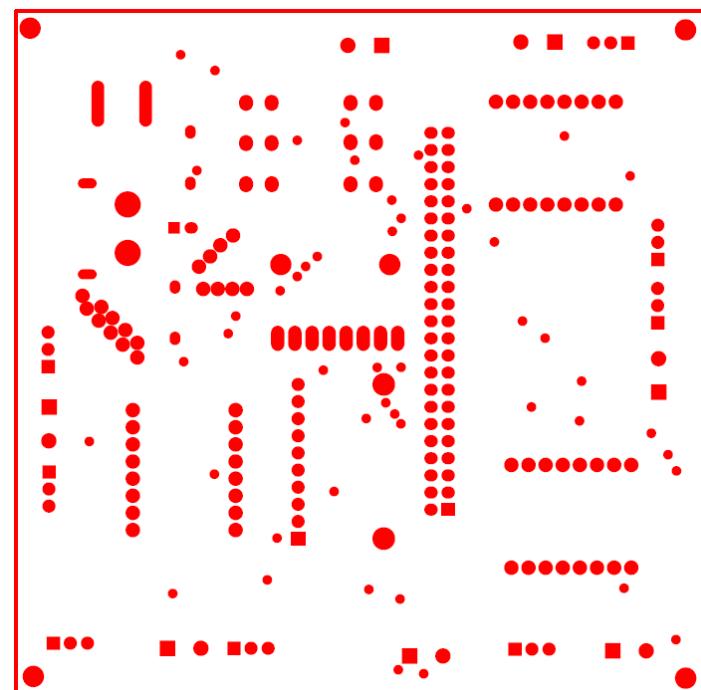


Figure 2.4: SolderMask Top Layer

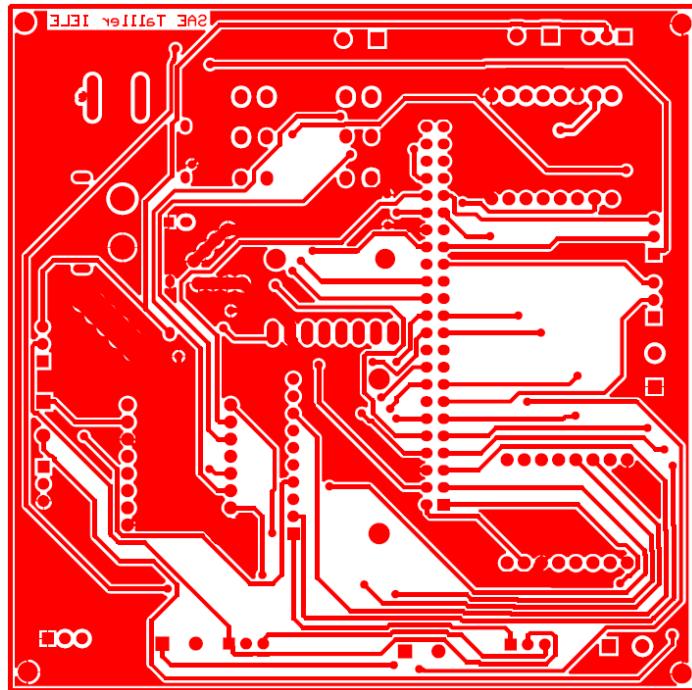


Figure 2.5: Copper Signal Bottom Layer

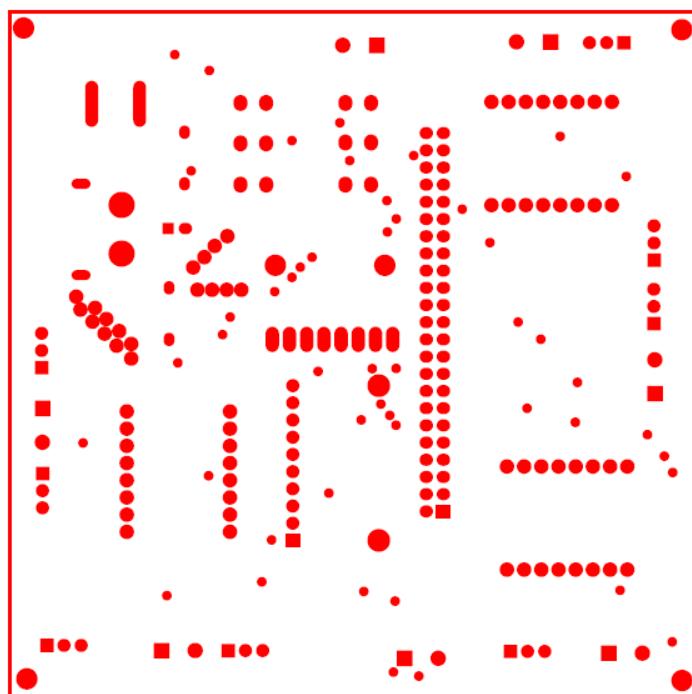


Figure 2.6: SolderMask Bottom Layer

### 2.2.3 Esquemático de conexiones

El diseño de la PCB se basó en el esquemático mostrado en la Figura 2.7, el cual integra todas las etapas funcionales del sistema: potencia, digital y análoga. Este esquemático constituye la representación lógica del

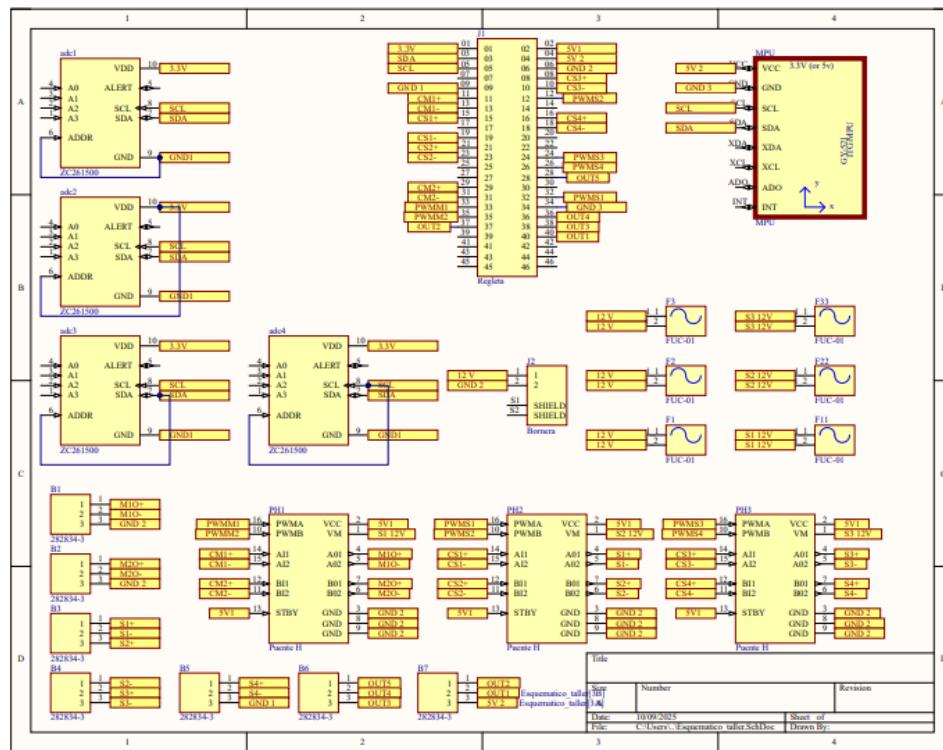
hardware y permite visualizar la interconexión entre los módulos de control, los actuadores y los periféricos de adquisición de datos.

En la parte superior derecha se encuentra la unidad MPU, encargada del procesamiento central del sistema. Esta sección incluye las líneas de alimentación de 3.3 V, las interfaces de comunicación I<sup>2</sup>C y SPI, así como los pines dedicados al control de actuadores. A su vez, en la zona superior izquierda y central se agrupan los dispositivos de sensado y comunicación digital, entre ellos los convertidores de nivel y módulos basados en I<sup>2</sup>C, los cuales permiten la expansión del número de señales disponibles para la MPU.

La sección intermedia del esquemático reúne los sensores que operan con señales análogas, incluyendo aquellos que funcionan bajo el principio de variación resistiva y el convertidor ADC encargado de digitalizar estas entradas. Esta estructura garantiza una adquisición confiable al mantener la separación entre trayectos de señal sensibles y líneas de alta potencia.

En la parte inferior se ubican los módulos de potencia, conformados por los tres puentes H utilizados para controlar los dos motores y los cuatro solenoides. Cada puente dispone de su propio fusible de protección en serie con la línea de alimentación, lo cual permite la detección y aislamiento de fallos. Asimismo, esta área incluye el interruptor maestro y el LED indicador del estado del sistema.

El esquemático permite observar claramente la organización del sistema, la asignación de conexiones y el flujo de señales entre las distintas etapas, constituyendo la base para el posterior diseño físico de la PCB.



ser de 0.8 mm por 1 A de corriente que fluye por esta. De este modo, se tomó el peor de los casos (flujo de pico máximo de corriente) con un margen de seguridad para cada caso con el fin de poder realizar un diseño seguro y funcional, y evitar el daño de la PCB por sobrecorrientes. En el caso de señales generales de control y de adquisición de datos digitales se usó un ancho de pista de 0.6 mm, el cual es suficiente para este tipo de usos. La siguiente tabla muestra para cada etapa el valor de corriente (corriente máxima pico sumando el margen de seguridad) y el ancho de la pista usado teniendo en cuenta lo descrito anteriormente.

Etapa	Ancho [mm]	Corriente [A]
Potencia	7.2	9
Motores (2)	4	5
Solenoides (2)	1.5	2
Señales generales (digitales y análogas)	0.6	0.75

Table 2.10: Reglas de ancho de pistas para las distintas configuraciones

## Clearance

Para el diseño de la PCB se establecieron dos valores principales de clearance. En primer lugar, el espaciado mínimo entre pistas se fijó en 0.3 mm, garantizando una separación adecuada para evitar cortocircuitos y minimizar la interferencia entre señales adyacentes. En segundo lugar, y conforme a las reglas de fabricación asociadas al blindaje, se definió una distancia mínima de 0.45 mm entre cualquier elemento de la PCB y el polígono de cobre conectado a GND, con el fin de asegurar un aislamiento apropiado y prevenir posibles arcos eléctricos o fallos por proximidad.

La selección de estos valores de clearance es fundamental para garantizar la confiabilidad eléctrica y mecánica del sistema. Un espaciado adecuado reduce el riesgo de fallas durante la operación, facilita el proceso de manufactura y contribuye a la integridad del señalamiento, especialmente en regiones donde se concentran pistas críticas o áreas sujetas a disipación térmica.

## Vías

En el proceso de diseño del circuito impreso se establecieron criterios diferenciados para la selección de vías, considerando tanto las características eléctricas de las señales como los requerimientos de manejo de corriente. Para las pistas correspondientes a señales digitales y analógicas se emplearon vías de 1.27 mm de diámetro externo con una perforación de 0.7 mm, dado que estos niveles de señal no implican flujos de corriente elevados y permiten mantener un tamaño compacto sin comprometer la integridad eléctrica.

Para la etapa de potencia se implementó una estrategia distinta, orientada a garantizar la transferencia adecuada de corriente entre capas. Se utilizaron vías de 2 mm de diámetro externo con perforación de 1.2 mm, asignando una vía por cada 0.5 A que circulara por la pista durante el cambio de capa. En aquellos trazos que superaban 1 A, se incorporaron múltiples vías en paralelo con el fin de distribuir el flujo de corriente y evitar puntos de concentración térmica. Esta práctica corresponde al denominado “efecto colador”, en el cual varias vías permiten que la corriente atraviese la PCB de forma uniforme, reduciendo pérdidas resistivas, mitigando el recalentamiento y preservando la integridad del material dieléctrico.

Como caso representativo, para pistas diseñadas para transportar 2 A se dispusieron cuatro vías ubicadas contiguamente, manteniendo únicamente la condición de que todas coincidieran con el punto de transición entre capas. Esta metodología se replicó proporcionalmente para los demás caminos de potencia, asegurando que la densidad de vías fuera coherente con la corriente máxima prevista en cada traza.

La siguiente tabla 2.11 muestra de forma resumida las dimensiones de las vías para cada tipo de etapa de conexión y el número de vías empleadas.

Etapa	Diámetro de vía [mm]	Perforación [mm]	Número de vias
Señales digitales y análogas	1.27	0.7	No críticas; baja corriente
Etapa de potencia	2.0	1.2	1 vía por cada 0.5 A
Motores	2.0	1.2	10 vías en paralelo para 5 [A]
Solenoides	2.0	1.2	4 vías en paralelo para 2 [A]

Table 2.11: Reglas de diseño y especificaciones de vías utilizadas en la PCB.

## Blindaje

Para el blindaje del sistema se implementaron polígonos de cobre en ambas capas del circuito impreso, los cuales fueron conectados al plano de tierra (GND) con el fin de mejorar el desempeño térmico y favorecer la estabilidad eléctrica del diseño. Estos polígonos fueron configurados con un clearance mínimo de 0.45 mm respecto a los demás elementos de la PCB, garantizando un aislamiento adecuado frente a pistas y componentes sensibles.

Las áreas ocupadas por los polígonos en cada capa se presentan a continuación:

Capa	Área [mm <sup>2</sup> ]
Top	3294.234
Bottom	4182.012

Table 2.12: Área

### 2.2.5 Etapas del diseño

El diseño del circuito impreso se estructuró en tres etapas funcionales principales: potencia, digital y análoga. Esta división permite organizar el ruteo, gestionar adecuadamente los retornos de corriente y garantizar la integridad de señal en cada sección del sistema. A continuación se describen las características generales de cada etapa y se presentan las figuras correspondientes a la distribución física dentro de la PCB.

#### Etapa de Potencia

La etapa de potencia comprende todo el sistema encargado de suministrar la energía necesaria para los actuadores de alto consumo. En esta zona se encuentran los tres puentes H: uno dedicado al control simultáneo de los dos motores de tracción, y dos adicionales destinados al accionamiento independiente de los cuatro solenoides (dos por puente). Cada puente H dispone de un fusible de protección en serie con su línea de alimentación, lo que permite aislar fallos y evitar sobrecorrientes que puedan comprometer la integridad de la PCB.

Adicionalmente, esta etapa incluye el interruptor general de encendido y el indicador luminoso tipo LED, el cual permite verificar visualmente el estado de alimentación del sistema. El ruteo de esta sección fue diseñado con pistas de gran ancho, múltiples vías de transición y zonas reforzadas de cobre para soportar adecuadamente las corrientes involucradas. La Figura 2.8 muestra el área (encerrada en los polígonos de color verde) correspondiente a la etapa de potencia dentro del diseño del circuito.

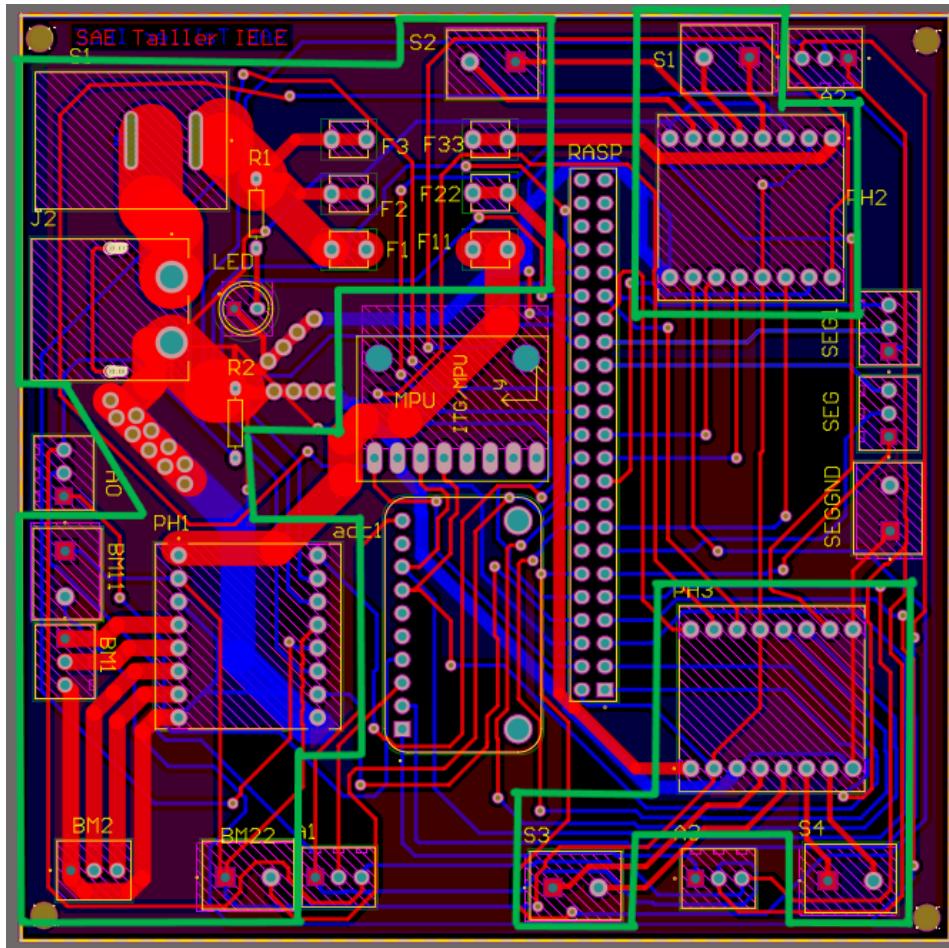


Figure 2.8: Distribución de la etapa de potencia, incluyendo los tres puentes H, fusibles de protección y elementos asociados al suministro energético.

### Etapa Digital

La etapa digital está conformada por los elementos responsables del procesamiento lógico y el control del sistema. En esta área se encuentra la unidad MPU encargada de la gestión de instrucciones, así como el módulo seguidor de línea que proporciona señales discretas utilizadas para la navegación del vehículo.

Las conexiones ubicadas en esta sección fueron ruteadas con pistas de menor ancho, manteniendo niveles de interferencia reducidos y separaciones adecuadas respecto a las zonas de potencia. Se emplearon vías de pequeño diámetro y planos de referencia para asegurar la integridad de las señales digitales de alta sensibilidad. La Figura 2.9 muestra la zona (área encerrada en los polígonos de color verde) correspondiente a la etapa digital.

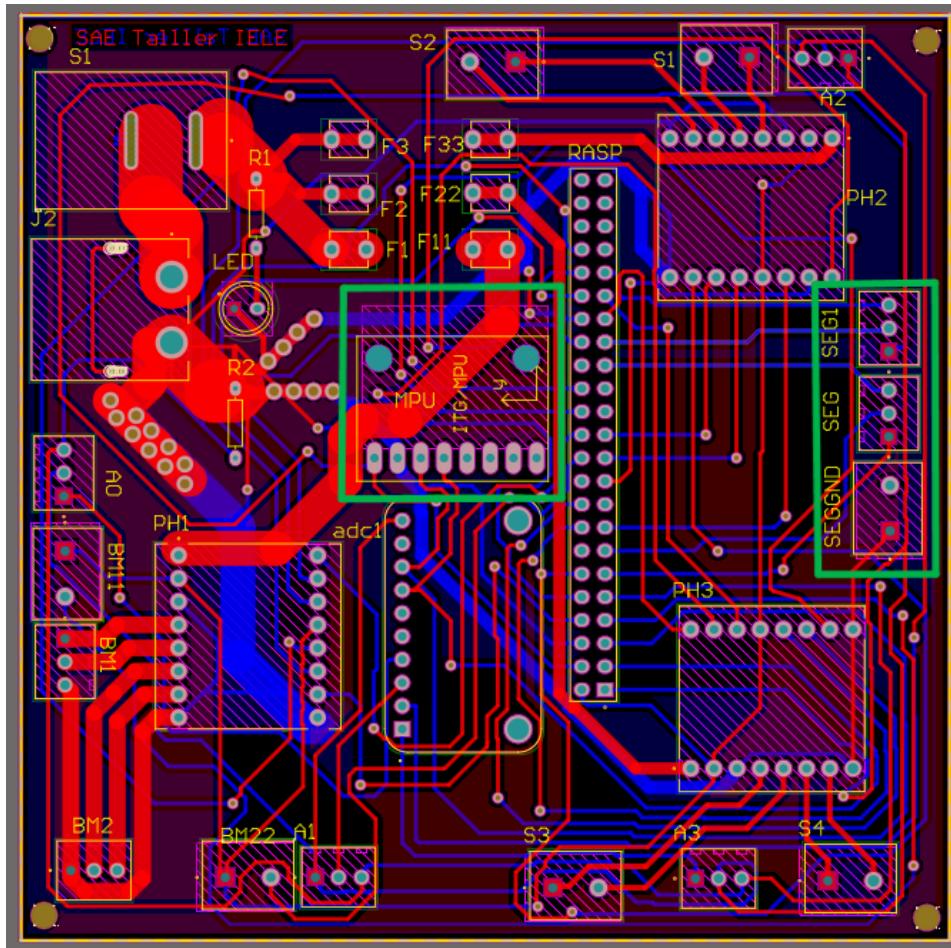


Figure 2.9: Área correspondiente a la etapa digital, conformada por la MPU y el módulo seguidor de línea.

### Etapa Análoga

La etapa análoga está conformada por los sensores encargados de medir variables físicas mediante señales continuas. En esta sección se incluyen sensores con comportamiento equivalente a potenciómetros variables, así como el convertidor ADC encargado de transformar dichas señales en información digital utilizable por la MPU.

Debido a la naturaleza sensible de las señales análogas, esta área fue ubicada lejos de las regiones de alta potencia y se diseñó con cuidado en relación al clearance, planos de referencia y longitud de pistas. El objetivo es minimizar el ruido eléctrico y garantizar la estabilidad durante la conversión analógico-digital. La Figura 2.10 presenta la distribución física (área encerrada en los polígonos de color verde) de la etapa análoga en la PCB.

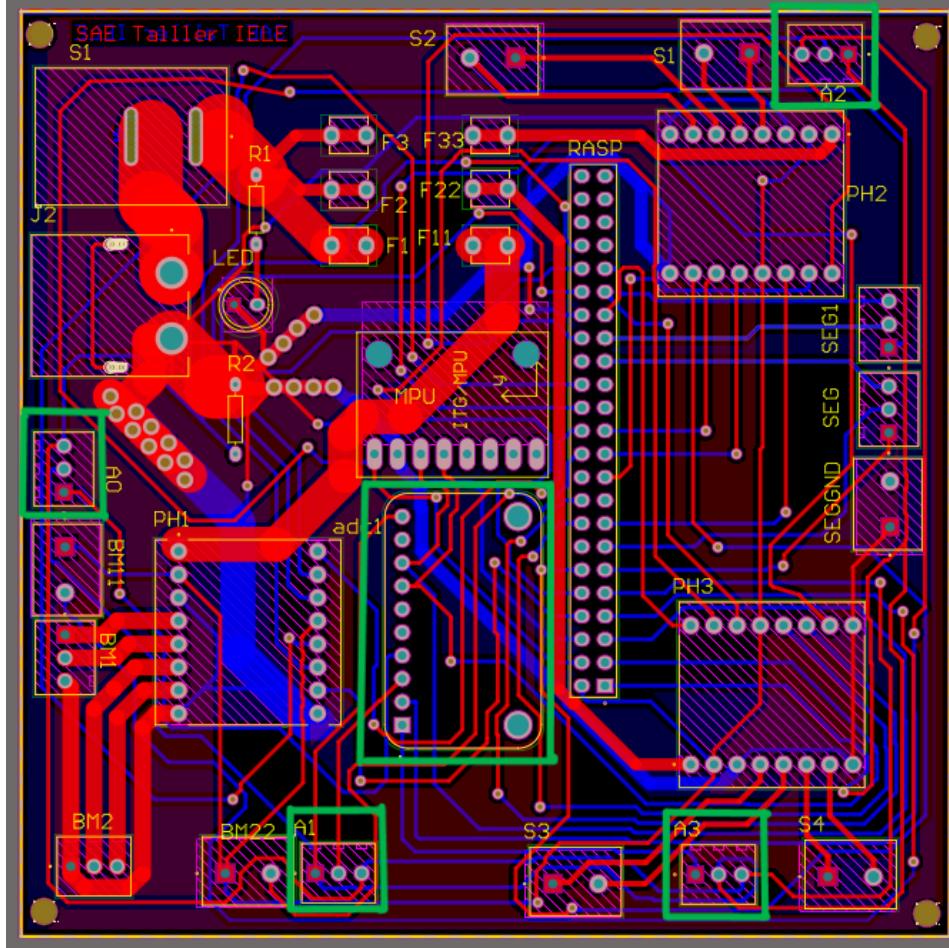


Figure 2.10: Distribución de la etapa analógica, que incluye los sensores de entrada y el convertidor ADC para el procesamiento de señales continuas.

## 2.3 Descripción del Software

El sistema de software del proyecto SAE está construido sobre una arquitectura modular basada en ROS2 (Robot Operating System 2), lo que permite desacoplar el procesamiento, la adquisición de datos, el control y la actuación física del sistema. La estructura del software garantiza escalabilidad, mantenibilidad y facilidad de integración con nuevas funcionalidades.

A continuación se describe la arquitectura lógica, los componentes principales y las interacciones entre paquetes ROS2.

### 2.3.1 Arquitectura General

El software del sistema SAE está organizado en un *workspace* ROS2 compuesto por varios paquetes que agrupan nodos según su funcionalidad. Cada módulo se comunica mediante *topics*, *services* y *parameters*, siguiendo la arquitectura:

**Sensores → Procesamiento → Control Central → Actuadores → GUI**

A continuación se presenta un resumen detallado de cada paquete y de los nodos implementados.

### 2.3.2 Componentes Principales del Software

#### 1. Paquete `main_package`

Este es el paquete más grande del sistema y contiene la lógica central del vehículo. Está organizado en subcarpetas:

- `launch/`: incluye los archivos `sae.launch.py` y `sensors.launch.py`.
  - `sae.launch.py`: lanza todos los nodos del sistema (sensores, controladores, actuadores y GUI).
  - `sensors.launch.py`: lanza únicamente nodos de sensado, utilizado para visualización y pruebas en la GUI.
- `actuadores/`: contiene los nodos encargados de enviar señales a los actuadores físicos.
  - `m1_actuator.py`: nodo para representar los motores DC que generan el movimiento de las ruedas.
  - `m2_actuator.py`: nodo para representar los solenoides utilizados en el módulo de amortiguación electromagnética.

**Nota:** Estos nodos aún no han sido probados sobre hardware real; su validación está pendiente.

- `controladores/`: módulos de control lógico asociados a los actuadores.
  - `m1_controller.py`: controlador para el actuador m1.
  - `m2_controller.py`: controlador para el actuador m2.

Actualmente, estos controladores incluyen únicamente una **implementación de referencia**, por lo que requieren una versión final de la estrategia de control.

- `sensores/`: contiene los nodos responsables de capturar datos del entorno y del chasis.
  - `distanceSensor_node.py`: nodo para el ADC ADS1115. Publica distancias estimadas a partir del voltaje medido en el canal configurado.
  - `mpu6050_node.py`: nodo para la IMU MPU6050. Publica datos crudos (*raw*) en formato JSON y aceleración Z como `Float32`.
  - `tcrt5000_node.py`: nodo de los sensores de línea TCRT5000. Publica un vector de cinco valores digitales.

#### 2. Paquete `service_package`

- `servicios/`: contiene servicios utilizados por la GUI o por nodos externos.
  - `web_gui_client.py`: nodo cliente que conecta ROS2 con una interfaz web Flask. Permite visualizar sensados y podría extenderse para enviar comandos de control.
- `templates/`: contiene los archivos HTML de la GUI integrada.
  - `index.html`: plantilla principal para visualización en tiempo real.
- `utils/`: funciones auxiliares utilizadas por distintos módulos.

En conjunto, este paquete constituye el núcleo de operación del vehículo, integrando sensores, bucles de control, actuadores y la interfaz gráfica.

### 2.3.3 Interacción entre Paquetes

El flujo de datos entre los módulos del sistema sigue la siguiente estructura lógica:

1. Los sensores generan datos crudos.
2. `sensores_package` publica la información en los topics correspondientes.
3. `main_package` recibe la información, la procesa y determina acciones en los nodos controladores.
4. los actuadores ejecutan las señales físicas sobre el hardware.
5. Paralelamente, `gui_package` puede enviar comandos mediante servicios ROS2 y mostrar información al usuario.

## 2.4 Entorno de Desarrollo y Herramientas de Software

El proyecto SAE requirió el uso de diversas herramientas de software para el diseño electrónico, el desarrollo del sistema embebido, la simulación, la comunicación distribuida y la implementación de la interfaz gráfica. A continuación se enumeran las versiones utilizadas y el propósito de cada una dentro del proyecto.

### Software para diseño electrónico

- **Altium Designer 25.8.1:** empleado para el diseño esquemático y de PCB del sistema. Incluyó la generación de librerías, reglas de diseño, rutado y preparación de archivos de fabricación.

### Software para el sistema embebido y comunicación

- **Ubuntu 24.04 LTS (64-bit):** sistema operativo instalado en la Raspberry Pi 5.
- **ROS2 Jazzy:** middleware utilizado para la comunicación entre nodos mediante *topics, services y parameters*.
- **Python 3.10+:** lenguaje principal para el desarrollo de nodos ROS2, controladores, actuadores y la interfaz gráfica.
- **Librerías especializadas:**
  - `rclpy`: API de ROS2 para nodos escritos en Python.
  - `adafruit_ads1x15`: para la lectura del ADC ADS1115.
  - `mpu6050`: para adquisición de datos de la IMU.
  - `RPi.GPIO`: control de pines digitales en la Raspberry Pi.
  - `busio, board`: interfaces para comunicación I2C.

### Software para la interfaz gráfica y visualización

- **Flask 3.x:** framework web utilizado para la GUI integrada con ROS2.
- **HTML:** empleados en el diseño de la interfaz alojada en `templates/index.html`.

En conjunto, este entorno de desarrollo permitió integrar diseño electrónico, programación embebida, simulación, comunicación distribuida mediante ROS2 y visualización interactiva del sistema.

## 2.5 Restricciones de diseño e implementación

Las siguientes restricciones establecen los límites y requisitos obligatorios que deben cumplirse durante el diseño de hardware, la implementación de software y la integración del sistema para garantizar la funcionalidad, confiabilidad y seguridad del proyecto, complementando las especificaciones detalladas en el Capítulo 2.

### 2.5.1 Restricciones de Potencia y Alimentación V

Las operaciones del sistema están limitadas por los rangos de corriente máxima y nominal de los componentes de potencia y la lógica digital.

Variable	Valor MIN	Valor NOM	Valor MAX
Tensión de Potencia Principal	14.8 V	16.8 V	16.8 V
Tensión Lógica de Control	3.3 V DC	5 V DC	5 V DC
Corriente de Alimentación Total	8.5 A	—	10 A
Corriente de Lógica Digital	0.2 A	—	0.3 A
Corriente de Componente Análoga	1 mA	—	3 mA

Table 2.13: Restricciones de Tensión y Corriente del Sistema.

### 2.5.2 Restricciones de Diseño y Fabricación PCB

Estas reglas son obligatorias para la integridad física y eléctrica del circuito impreso.

- **Restricción de Dimensiones Físicas:** El sistema electrónico debe estar estrictamente limitado al espacio cuadrado máximo de **101.5 mm × 101.5 mm** para respetar la optimización de volumen del vehículo.
- **Restricción de Densidad de Vías (Disipación):** Se debe implementar la técnica de **Efecto Colador** en las líneas de alta corriente y planos de tierra, utilizando una densidad de **1 vía de 1.2 mm** de perforación por cada **0.5 A** que fluya entre capas, con el fin de garantizar la disipación térmica y la baja resistencia.
- **Restricción de Aislamiento Crítico (Clearance):** El espacio mínimo de **0.45 mm** entre cualquier elemento conductor (pista, via o pad) y el polígono de cobre **GND** debe ser mantenido para prevenir arcos eléctricos y garantizar el aislamiento.
- **Restricción de Interfaz Unificada:** Todos los sensores principales deben utilizar el bus **I2C** para la adquisición de datos, obligando al microcontrolador a gestionar la carga capacitiva y la estabilidad del bus a una velocidad mínima de 400 kHz.

### 2.5.3 Restricciones de Software y Rendimiento ROS2

La arquitectura de software y el rendimiento deben cumplir con los siguientes requisitos funcionales.

- **Restricción de Arquitectura (Modularidad):** La implementación debe mantener una estricta separación de responsabilidades: la lógica de control reside exclusivamente en el **main\_package**, la lectura en **sensores\_package** y la ejecución en **actuadores\_package**. La comunicación es obligatoria a través de **Topics, Services o Actions** de ROS2.
- **Restricción de Latencia Crítica:** El bucle de control que gestiona el **Seguimiento de Línea** y el **Amortiguamiento Electromagnético Activo** debe operar a una **frecuencia mínima de 100 Hz**. El tiempo de latencia desde la lectura del sensor IMU/Línea hasta la actuación física debe ser inferior a **10 ms**.

- **Restricción de Robustez y Fail-Safe:** El `main_package` debe implementar una función de **seguridad activa (\*fail-safe\*)** que detenga los actuadores ante la pérdida de datos vitales de sensores o la falla en la comunicación con el `actuadores_package`.
- **Restricción de Versión Fija:** Para asegurar la reproducibilidad, el sistema debe ser desarrollado y desplegado sobre una **versión específica y congelada de ROS2** y del sistema operativo base (ej. Ubuntu/Linux), prohibiendo la actualización de paquetes que puedan introducir inestabilidad.

#### 2.5.4 Restricciones Ambientales y de Calidad $\Delta$

- **Restricción de Temperatura Operacional:** El sistema debe funcionar de manera confiable en el rango de temperatura de **10°C a 30°C**, debido a la limitación del sensor **TCRT 5000**.
- **Restricción de Mantenibilidad Modular:** La PCB debe usar conectores que permitan el reemplazo rápido y modular del microcontrolador y los periféricos de potencia (drivers) sin necesidad de soldadura, facilitando el diagnóstico y la reparación en campo.
- **Restricción de Pruebas Unitarias:** Se exige que los paquetes de software críticos cuenten con **cobertura de pruebas unitarias** para validar la lógica algorítmica y las transformaciones de datos antes de la integración final.

## Chapter 3

# External Interface Requirements

### 3.1 Interfaz de usuario GUI

La interfaz de usuario del sistema SAE corresponde a una aplicación web ligera desarrollada en Flask e integrada directamente con ROS2 mediante el nodo `SpeedWebClient`. Actualmente, la GUI se levanta automáticamente al ejecutar el archivo de lanzamiento `sensors.launch.py`, lo que permite visualizar en tiempo real los datos provenientes de los nodos de sensado sin necesidad de activar los controladores o actuadores.

#### Nodo `SpeedWebClient`

El nodo `SpeedWebClient` es un nodo ROS2 escrito en Python que cumple dos funciones simultáneas:

1. Actúa como cliente de ROS2, suscribiéndose a los *topics* del sistema para obtener información sensorial.
2. Ejecuta un servidor web Flask, permitiendo que un navegador externo acceda a la interfaz gráfica.

El nodo se inicializa dentro del archivo `main()`, donde se construye el objeto `SpeedWebClient` y se lanza un hilo separado encargado de ejecutar el bucle ROS2. De esta manera, Flask y ROS pueden operar en paralelo sin bloquearse mutuamente.

Las fuentes de datos que la interfaz recibe incluyen:

- `tcrt5000_topic`: vector de cinco sensores de línea.
- `distanceSensor_topic`: lecturas del ADS1115.
- `mpu6050_accel_z`: aceleración vertical proveniente de la IMU.
- `max_speed_pct` y `status_text`: variables internas del sistema principal.

Además, el nodo implementa un cliente del servicio `SetSpeed.srv`, permitiendo que el usuario ajuste manualmente la velocidad del vehículo desde la interfaz web.

#### Integración con Flask

La GUI se implementa mediante un servidor Flask inicializado dentro del mismo módulo. Se utiliza la plantilla HTML principal `index.html`, ubicada en el directorio `templates/` del paquete `main_package`. La HTML se sirve desde los siguientes endpoints:

- `/` : carga la interfaz gráfica principal.
- `/status` : devuelve datos actualizados de los sensores en formato JSON.
- `/set_speed` : envía una solicitud al servicio ROS2 para ajustar la velocidad.

El servidor web se ejecuta en:

`http://0.0.0.0:5000`

lo que expone la GUI a cualquier dispositivo conectado a la misma red local.

## Acceso desde la red LAN

Dado que la Raspberry Pi ejecuta el servidor Flask escuchando en todas las interfaces (`0.0.0.0`), la GUI puede ser accedida desde computadores, tablets o teléfonos conectados a la misma red LAN del laboratorio. En la práctica, el acceso se realiza mediante:

`http://raspberrypi.local:5000`

o alternativamente, utilizando la dirección IP asignada a la Raspberry Pi:

`http://<IP-raspberry>:5000`

Esto permite monitorear el sistema de forma inalámbrica sin interferir con los procesos ROS2 que corren en la Raspberry.

## Estado del sistema visible en la GUI

La interfaz muestra en tiempo real:

- Las lecturas de los sensores TCRT5000.
- Las mediciones de distancia provenientes del ADS1115.
- La aceleración en el eje Z de la IMU.
- El estado general del sistema y la velocidad actual.

Adicionalmente, la arquitectura permite extender la GUI para enviar comandos más avanzados al sistema, integrar nuevas métricas o habilitar paneles de control para actuadores.

## Chapter 4

# Funcionalidades del sistema

El sistema SAE integra dos funcionalidades principales: el seguimiento autónomo de línea y el amortiguamiento electromagnético activo. Ambas se implementan mediante una arquitectura distribuida basada en ROS2, donde distintos nodos cooperan para procesar información sensorial, ejecutar algoritmos de control y accionar los actuadores correspondientes.

## 4.1 Seguimiento de línea

### Descripción general

El seguimiento de línea consiste en permitir que el vehículo identifique y mantenga su trayectoria sobre una línea marcada en el piso. Para ello se utilizan cinco sensores TCRT5000 ubicados en la parte inferior frontal del vehículo, capaces de detectar contrastes entre superficie oscura (línea) y clara (fondo).

El objetivo del sistema es ajustar dinámicamente la velocidad relativa de los motores izquierda/derecha para mantener el centro geométrico del robot alineado con la línea detectada.

En futuras versiones, se planea integrar encoders en el actuador `m1_actuator` para mejorar la precisión del control de velocidad y permitir estimación de posición longitudinal.

### Flujo esperado entre nodos

El flujo operativo para esta funcionalidad es el siguiente:

1. El usuario ingresa un porcentaje de velocidad deseada mediante la GUI o un servicio ROS2.
2. El nodo `tcrt5000_node` publica un vector de cinco estados digitales correspondientes a la detección de línea.
3. (Futuro) Los encoders en `m1_actuator` publicarán velocidad real de los motores.
4. El nodo `m1_controller` recibe:
  - El valor de velocidad deseada.
  - El estado de los sensores TCRT5000.
  - (Futuro) La velocidad real medida.
5. El controlador calcula correcciones de trayectoria ajustando el balance entre los motores izquierdo y derecho.

6. El nodo `m1_actuator` ejecuta los PWM y direccionales necesarios para producir el movimiento corregido.

En forma compacta:

Usuario (velocidad deseada) + TCRT5000 + (Encoder) → `m1_controller` → `m1_actuator`

## 4.2 Amortiguamiento electromagnético activo

### Descripción general

El sistema de amortiguamiento electromagnético activo tiene como objetivo mejorar la estabilidad del vehículo compensando perturbaciones verticales mediante solenoides actuados electrónicamente. Cada una de las cuatro ruedas posee un módulo de suspensión independiente, lo que permite ajustar la fuerza electromagnética de forma localizada.

Para la estimación del estado dinámico vertical, se utilizan dos tipos de sensores:

- **Acelerómetro MPU6050:** mide aceleración en el eje Z, útil para detectar impactos, vibraciones y oscilaciones.
- **Sensores de distancia:** miden la separación entre el chasis y el suelo, permitiendo estimar la compresión o extensión de la suspensión.

La combinación de estas señales permite implementar controladores de tipo PD, LQR o incluso controladores adaptativos en futuras versiones.

### Flujo esperado entre nodos

Para cada una de las cuatro llantas, el flujo operativo se describe como:

1. El nodo `mpu6050_node` publica aceleración vertical (`mpu6050_accel_z`).
2. Los nodos de `distanceSensor_node` publican distancias entre el chasis y el punto de referencia.
3. El nodo `m2_controller` recibe:
  - Aceleración en el eje Z.
  - Distancia actual de la suspensión.
4. El controlador determina la corriente o nivel de activación requerido para los solenoides.
5. El nodo `m2_actuator` genera la señal correspondiente para actuar sobre el módulo electromagnético.

En forma resumida:

`MPU6050 + Sensores de distancia → m2_controller → m2_actuator`

Cada llanta opera de manera independiente, lo que permite un amortiguamiento adaptativo en tiempo real.