



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Diseño e implementación de un sistema de sensorización y
transmisión de datos para un vehículo terrestre no tripulado*

Grado en Ingeniería Mecánica

ALUMNO: Manuel Luaces Cebreiro

DIRECTORES: González Prieto, José Antonio
Ogando Martínez, Ana

CURSO ACADÉMICO: 2025-2026

UniversidadeVigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Diseño e implementación de un sistema de sensorización y
transmisión de datos para un vehículo terrestre no tripulado*

Grado en Ingeniería Mecánica

Intensificación en Tecnología Naval

Infantería de Marina

UniversidadeVigo

RESUMEN

El presente Trabajo Fin de Grado aborda el diseño e implementación de un sistema de sensorización para un vehículo experimental con el objetivo de adquirir datos del entorno y del movimiento del propio vehículo. Para ello, se ha llevado a cabo la selección e integración de distintos sensores, incluyendo un LiDAR, una cámara USB, dos unidades de medición inercial (IMU) y un receptor GNSS, gestionados mediante una Raspberry Pi como unidad de procesamiento.

En primer lugar, se ha realizado el diseño mecánico de soportes específicos mediante modelado CAD en Autodesk Inventor, permitiendo la correcta fijación de los sensores al chasis del vehículo. Dichos soportes han sido fabricados mediante impresión 3D utilizando PLA. Posteriormente, se ha desarrollado la arquitectura del sistema y la implementación del software necesario para la adquisición de datos de los diferentes sensores.

Finalmente, se ha llevado a cabo la integración completa del sistema en el vehículo y la realización de pruebas experimentales para verificar su funcionamiento. Este trabajo demuestra la viabilidad de un sistema de adquisición multisensorial aplicado a plataformas móviles, sentando las bases para futuros desarrollos en el ámbito de la robótica móvil y la conducción autónoma.

PALABRAS CLAVE

Sistema no tripulado, ROS 2, Fusión Sensorial, Adquisición de datos en tiempo real, Sistemas Multisensor.

AGRADECIMIENTOS

En primer lugar, me gustaría expresar mi más sincero agradecimiento a mis compañeros de la promoción 426-156 en especial a mi camarera, la 103, Santiago Cervera Sáez de Santa María, Álvaro Fernández Valverde y Rafael Francisco Liaño Mock, por su apoyo constante a lo largo de todo el año. Su ayuda, tanto en el ámbito académico como en el personal, ha sido fundamental para afrontar y superar las distintas etapas de este trabajo.

Asimismo, quiero agradecer a la promoción 156 del Real y Glorioso Cuerpo de la Infantería de Marina, por el compañerismo, la unión y el apoyo incondicional demostrado en todo momento. Ser parte de esta promoción ha sido, sin duda, una de las experiencias más enriquecedoras de esta etapa y probablemente de toda mi vida.

De manera especial, agradezco a mi familia, y en particular a mis padres, por estos cinco años de apoyo constante. A pesar de la distancia, su cercanía, confianza y respaldo han estado siempre presentes, siendo un pilar fundamental en mi formación tanto personal como profesional.

Quisiera también reconocer la labor de mis tutores, por su comprensión, paciencia, disponibilidad y orientación a lo largo de todo el proceso de desarrollo de este Trabajo Fin de Grado.

Por último, gracias a la Armada y a España, porque aquel yo de 17 años que cruzó por primera vez la puerta de Carlos I no podía imaginar la oportunidad que se le estaba brindando. Gracias por la familia formada en el camino, por los valores inculcados, por las enseñanzas recibidas y por la transformación personal que ha supuesto este recorrido.

CONTENIDO

Contenido	1
Índice de Figuras	5
Índice de Tablas.....	9
1 Introducción y objetivos	11
1.1 Antecedentes	11
1.2 Motivación	11
1.3 Objetivos	12
1.4 Estructura del trabajo	12
2 Estado del arte	14
2.1 Introducción	14
2.2 Vehículos terrestres no tripulados (UGV)	14
2.3 Sensores empleados en sistemas no tripulados	15
2.3.1 Sensores LiDAR	15
2.3.2 Sensores Radar.....	17
2.3.3 Cámaras RGB	18
2.3.4 Cámaras Infrarrojas/Térmicas	19
2.3.5 Sensores Sonar.....	21
2.3.6 Sistemas de posicionamiento global (GPS/GNSS).....	22
2.3.7 Unidades de medida inercial (IMU)	24
2.3.8 Sistema AIS	25
2.3.9 Sistema DVL	26
2.3.10 Síntesis de los diferentes sensores empleados en sistemas de percepción	26
2.4 Fusión Sensorial	27
2.5 Arquitecturas Sensoriales Avanzadas	30
2.6 Aplicaciones reales de arquitecturas sensoriales en sistemas no tripulados	32
2.6.1 Vehículos Aéreos No Tripulados (UAV)	32
2.6.2 Vehículos Terrestres No Tripulados (UGV).....	33
2.6.3 Vehículos de Superficie No Tripulados (USV)	34
2.6.4 Vehículos Submarinos No Tripulados (AUV).....	35
2.7 Tecnologías emergentes en sensorización y percepción avanzada	36
2.7.1 Tecnologías emergentes en sensores	37
2.7.2 Tecnologías emergentes en procesamiento	37
2.7.3 Impacto potencial en sistemas no tripulados	37

2.8 Modelos de Error, Ruido y Calibración	38
2.8.1 Errores y ruido en sensores	38
2.8.2 Modelo conceptual de error en las mediciones	39
2.8.3 Calibración de sensores	39
2.8.4 Impacto del error y relación con la fusión sensorial	39
2.9 SLAM específico por sensor	39
2.9.1 SLAM basado en visión (Visual SLAM)	40
2.9.2 SLAM basado en LiDAR (LiDAR SLAM)	40
2.9.3 SLAM basado en radar (Radar SLAM)	41
2.9.4 SLAM basado en sonar (Sonar SLAM)	41
2.9.5 Comparación y complementariedad entre enfoques	41
2.10 Inteligencia artificial en percepción cognitiva	42
2.10.1 IA en sensores y procesamiento de datos	42
2.10.2 IA en fusión sensorial	42
2.10.3 IA en percepción cognitiva	43
2.10.4 Beneficios y limitaciones	43
2.10.5 Perspectiva futura	43
2.11 Aplicaciones en la Armada y ejemplos reales	43
2.11.1 Aplicaciones en el dominio marítimo	44
2.11.2 Aplicaciones en el dominio aéreo	44
2.11.3 Aplicaciones en el dominio terrestre	45
2.11.4 Integración de sistemas tripulados y no tripulados	45
2.12 Normativa y Seguridad en Sensores	45
2.13 Conclusiones Generales	47
3 Desarrollo del TFG	48
3.1 Plataformas y Sensores del Sistema	48
3.1.1 Plataforma Móvil	48
3.1.2 Unidad de Procesamiento	50
3.1.3 Sensor LiDAR	52
3.1.4 Cámara USB	55
3.1.5 Sensores Inerciales IMU	57
3.1.6 Sistema de Posicionamiento Global (GNSS)	60
3.1.7 Resumen del Conjunto de Plataformas y Sensores	64
3.2 Diseño e integración de soportes para sensores	65
3.3 Desarrollo del sistema embarcado	69
3.3.1 Preparación del sistema operativo	70

3.3.2 Configuración de red y acceso remoto	71
3.3.3 Instalación de Paquetes Básicos y Herramientas del Sistema	75
3.3.4 Configuración de permisos y gestión de paquetes Python	75
3.3.5 Instalación y Configuración de la IMU Phidget	76
3.3.6 Instalación y comprobación del receptor GNSS	77
3.3.7 Instalación y comprobación de la Cámara USB	77
3.3.8 Instalación y comprobación del LIDAR	78
3.3.9 Resumen Librerías Instaladas	78
3.3.10 Instalación y Configuración de ROS 2 Jazzy	79
3.3.11 Integración del sistema de visualización mediante Foxglove Bridge	80
3.4 Arquitectura del sistema	82
3.4.1 Bloques funcionales del sistema	82
3.4.2 Arquitectura Hardware	82
3.4.3 Arquitectura Software	83
3.4.4 Flujo de datos	84
3.4.5 Justificación de la Arquitectura Adoptada	85
3.4.6 Consideraciones y limitaciones del sistema	86
3.5 Implementación Software del Sistema Multisensor	87
3.5.1 Diseño del nodo ROS	87
3.5.2 Gestión Temporal y ejecución del sistema	87
3.5.3 Adquisición y procesamiento de datos	88
3.5.5 Sistema de referencia y transformaciones	90
3.5.6 Gestión de frecuencias y sincronización	92
3.5.7 Diseño modular del sistema	93
3.5.8 Integración con Foxglove	94
3.5.9 Síntesis del desarrollo software	96
3.5.10 Ventajas de la implementación adoptada	97
3.5.11 Consideraciones finales	97
4 Resultados y validación experimental	99
4.1 Entorno Experimental	99
4.2 Metodología de pruebas	100
4.3 Validación de sensores	101
4.3.1 Validación de la IMU	101
4.3.2 Validación del GNSS	103
4.3.3 Validación de la Cámara	104

4.3.4 Validación del LiDAR.....	105
4.4 Análisis del rendimiento del Sistema.....	106
4.5 Problemas encontrados y soluciones.....	107
4.6 Conclusiones de Validación.....	109
5 Conclusiones y líneas futuras	111
5.1 Consecución de los objetivos	111
5.2 Conclusión final	111
5.3 Aplicación en el ámbito militar.....	112
5.4 Líneas Futuras	113
6 Bibliografía.....	114
Anexo I: Implicaciones Sociales, y/o Económicas, y/o Ambientales	118
Anexo II: Reflexiones Éticas y Sociales	120
Anexo III: Declaración sobre el uso de herramientas de inteligencia artificial	121
Anexo IV: Planimetría.....	122

ÍNDICE DE FIGURAS

Figura 2-1 El funcionamiento básico de un sensor LiDAR. Fuente: Elaboración asistida por herramientas de inteligencia artificial	16
Figura 2-2 Esquema de funcionamiento de una cámara RGB digital. Fuente: Elaboración asistida por herramientas de inteligencia artificial	19
Figura 2-3 Esquema conceptual del funcionamiento de un sistema GNSS basado en la recepción de señales de varios satélites y el cálculo de la posición del receptor mediante el principio de trilateración. Fuente: Elaboración asistida por herramientas de inteligencia artificial	23
Figura 2-4 Ejemplo conceptual de una arquitectura sensorial multimodal basada en la fusión de información procedente de sensores heterogéneos para tareas de percepción y planificación en sistemas no tripulados. Fuente: Elaboración asistida por herramientas de inteligencia artificial	32
Figura 2-5 Arquitectura sensorial en vehículos aéreos no tripulados (UAV) y principales aplicaciones asociadas. Fuente: Elaboración asistida por herramientas de inteligencia artificial	33
Figura 2-6 Arquitectura sensorial en vehículos terrestres no tripulados (UGV) y principales aplicaciones asociadas. Fuente: Elaboración asistida por herramientas de inteligencia artificial	34
Figura 2-7 Arquitectura sensorial en vehículos de superficie no tripulados (USV), incluyendo sensores principales y aplicaciones en vigilancia marítima y monitorización oceanográfica. Fuente: Elaboración asistida por herramientas de inteligencia artificial	35
Figura 2-8 Arquitectura sensorial en vehículos submarinos no tripulados (AUV) y principales aplicaciones asociadas. Fuente: Elaboración asistida por herramientas de inteligencia artificial	36
Figura 2-9 Comparativa de los principales sistemas no tripulados (UAV, UGV, USV y AUV) y de los sensores habitualmente integrados en cada dominio operativo. Fuente: Elaboración asistida por herramientas de inteligencia artificial	36
Figura 2-10 Proceso de localización y mapeado simultáneos (SLAM), incluyendo las etapas de percepción sensorial, procesamiento de datos, estimación de trayectoria y generación del mapa del entorno. Fuente: Elaboración asistida por herramientas de inteligencia artificial	40
Figura 2-11 Empleo de un vehículo aéreo no tripulado (UAV) en un entorno operativo para tareas de reconocimiento, vigilancia y adquisición de información sensorial. Fuente: [25]	44
Figura 2-12 Integración conceptual de plataformas tripuladas y no tripuladas de la Armada en los dominios marítimo, aéreo y terrestre mediante el uso transversal de sensores avanzados y la compartición de información sensorial. Fuente: Elaboración asistida por herramientas de inteligencia artificial	45
Figura 3-1 Plataforma móvil empleada en el proyecto, basada en un vehículo teledirigido Land Rover Defender TRX4, que sirve como base para la integración de los distintos sensores y sistemas de procesamiento. Fuente: [44]	48
Figura 3-2 Raspberry Pi 4 Model B empleada como unidad central de procesamiento para la adquisición, sincronización y gestión de los datos procedentes de los distintos sensores del sistema ...	50
Figura 3-3 Ubicación de la Raspberry Pi dentro del montaje en el UGV	52
Figura 3-4 Principio de funcionamiento de un sensor LiDAR bidimensional, basado en la emisión de pulsos láser y la medición de distancias mediante un barrido plano de 360 grados alrededor del sensor. Fuente: Elaboración propia mediante generación asistida por inteligencia artificial	53

Figura 3-5 Sensor LiDAR bidimensional de la familia RPLIDAR empleado para la adquisición de información geométrica del entorno mediante barrido plano. Fuente: [6]	53
Figura 3-6 Esquema inicial de la ubicación del sensor LiDAR sobre un soporte cilíndrico elevado en la parte frontal del techo del <i>rover</i> . Fuente: Elaboración propia mediante generación asistida por inteligencia artificial	55
Figura 3-7 Cámara USB de 2.0 megapíxeles empleada como sistema de visión para la captura de imágenes en color del entorno	55
Figura 3-8 Imagen adquirida por la cámara RGB del sistema durante la operación del vehículo ...	57
Figura 3-9 Sensor inercial <i>PhidgetSpatial Precision 3/3/3</i> empleado para la medida de aceleraciones, velocidades angulares y orientación del vehículo. Fuente: [17]	58
Figura 3-10 Sistema de ejes de referencia de la unidad de medida inercial (IMU), mostrando las aceleraciones lineales y las rotaciones medidas en los tres ejes espaciales X, Y y Z. Fuente: Elaboración propia mediante generación asistida por inteligencia artificial.....	58
Figura 3-11 Diferencias entre los montajes confeccionados a lo largo del desarrollo del proyecto	60
Figura 3-12 Receptor GNSS con conexión USB empleado para la obtención de información de posicionamiento global del vehículo.....	61
Figura 3-13 Esquema conceptual del sistema de posicionamiento global (GNSS), en el que varios satélites GNSS transmiten señales al vehículo, que actúa como receptor para la determinación de su posición. Fuente: Elaboración propia mediante generación asistida por inteligencia artificial.....	62
Figura 3-14 Integración del módulo GNSS dentro de Foxglove para la obtención de datos de posicionamiento	63
Figura 3-15 Boceto inicial de la planta del sistema mostrando la plataforma móvil, la unidad de procesamiento central (Raspberry Pi 4) y la relación funcional con los distintos sensores integrados. Fuente: Elaboración propia mediante generación asistida por inteligencia artificial	65
Figura 3-16 Modelo CAD del soporte diseñado en Autodesk Inventor para la integración conjunta del sensor LiDAR y la cámara USB en el vehículo de pruebas.....	66
Figura 3-17 Soporte impreso en 3D diseñado para la integración conjunta del sensor LiDAR y la cámara USB. La pieza permite alojar ambos dispositivos en una única estructura, posicionando el LiDAR en la parte superior y la cámara en la zona frontal del soporte.....	67
Figura 3-18 Modelo CAD del soporte diseñado en Autodesk Inventor para el alojamiento de la Raspberry Pi, adaptado a las dimensiones del dispositivo y a su integración en el chasis del vehículo	68
Figura 3-19 Soporte impreso en 3D destinado al alojamiento de la Raspberry Pi, diseñado para integrarse dentro de la estructura superior del vehículo y permitir el acceso a los distintos puertos de conexión.....	68
Figura 3-20 Vista lateral y frontal del vehículo con el sistema de sensores instalado mediante el uso de soportes.....	69
Figura 3-21 Ventana de elección del dispositivo dentro del proceso de configuración de la Raspberry Pi dentro del <i>Imager</i>	70
Figura 3-22 Ventana de elección del sistema operativo dentro del proceso de configuración de la Raspberry Pi dentro del <i>Imager</i>	71
Figura 3-23 Configuración del Netplan empleada durante la configuración de la unidad de procesamiento	72

Figura 3-24 Esquema de conectividad y acceso remoto del sistema, donde la Raspberry Pi se conecta a la red WiFi mediante una dirección IP estática, permitiendo el acceso remoto vía SSH y la visualización de datos desde un equipo externo mediante Foxglove. Fuente: Elaboración asistida por herramientas de inteligencia artificial	74
Figura 3-25 Arquitectura del flujo de datos software del sistema multisensor, desde la adquisición sensorial y su procesamiento mediante librerías específicas, hasta su distribución en ROS 2 y su visualización remota mediante Foxglove Bridge. Fuente: Elaboración asistida por herramientas de inteligencia artificial	82
Figura 3-26 Esquema de la arquitectura hardware del sistema. Fuente: Elaboración asistida por herramientas de inteligencia artificial	83
Figura 3-27 Interfaz de Foxglove con flujo de datos de los distintos sensores empleados.....	85
Figura 3-28 Fragmento de la inicialización del nodo dentro del script <i>multi_sensor_sim</i>	87
Figura 3-29 Fragmento de inicialización de temporizadores dentro del script <i>multi_sensor_sim</i> ...	87
Figura 3-30 Fragmento de la IMU dentro del script <i>multi_sensor_sim</i>	88
Figura 3-31 Fragmento del GNSS dentro del script <i>multi_sensor_sim</i>	88
Figura 3-32 Fragmento del LiDAR dentro del script <i>multi_sensor_sim</i>	89
Figura 3-33 Fragmento de la Cámara USB dentro del script <i>multi_sensor_sim</i>	89
Figura 3-34 Fragmento de los distintos <i>publishers</i> de cada sensor dentro del script <i>multi_sensor_sim</i>	90
Figura 3-35 Fragmento de la publicación de transformaciones dentro del script <i>multi_sensor_sim</i>	91
Figura 3-36 Árbol de transformaciones del sistema basado en tf2, donde se definen los marcos de referencia del vehículo y de los sensores. Fuente: Elaboración asistida por herramientas de inteligencia artificial	91
Figura 3-37 Arquitectura modular del sistema multisensor, organizada en capas desde la adquisición de datos hasta su visualización. Fuente: Elaboración asistida por herramientas de inteligencia artificial	94
Figura 3-38 Configuración de la conexión WebSocket en Foxglove para el acceso remoto a los tópicos ROS 2 del sistema embarcado	95
Figura 3-39 Visualización en tiempo real de los datos multisensor en Foxglove, incluyendo información de cámara, LiDAR, GNSS e IMU	96
Figura 4-1 Plataforma experimental utilizada durante las pruebas, incluyendo la integración de los sensores y la unidad de procesamiento embarcada	100
Figura 4-2 Interfaz de Foxglove durante la fase de validación, mostrando datos multisensor en tiempo real.....	101
Figura 4-3 Representación temporal de los datos de la IMU en Foxglove, incluyendo roll, pitch y yaw, donde se observa un comportamiento estable en condiciones estáticas y una respuesta coherente ante cambios de orientación.....	102
Figura 4-4 Visualización de la posición GNSS en Foxglove, donde se observa la estimación de la ubicación del sistema y las pequeñas variaciones asociadas al error de posicionamiento.	103
Figura 4-5 Visualización del flujo de imágenes procedente de la cámara USB en Foxglove, donde se observa la captura del entorno en tiempo real	104

Figura 4-6 Visualización de los datos del sensor LiDAR en Foxglove, mostrando la representación del entorno mediante una nube de puntos en tiempo real 105

ÍNDICE DE TABLAS

Tabla 2-1 Comparativa de las principales características de las cámaras RGB y las cámaras infrarrojas/térmicas en el contexto de sistemas no tripulados.....	21
Tabla 2-2 Comparativa de sensores utilizados en sistemas de percepción para vehículos autónomos, incluyendo sus características técnicas y su contribución en estrategias de fusión sensorial.....	27
Tabla 2-3 Comparativa entre los distintos niveles de fusión.....	29
Tabla 2-4 Principales métodos de fusión sensorial	29
Tabla 2-5 Comparativa de los principales sistemas no tripulados (UAV, UGV, USV y AUV) y de los sensores habitualmente integrados en cada dominio operativo.....	38
Tabla 2-6 Comparativa de los principales enfoques de SLAM en sistemas no tripulados según el tipo de sensor empleado	42
Tabla 2-7 Normativas en seguridad y sensores	47
Tabla 3-1 Especificaciones técnicas del vehículo base Traxxas TRX-4 empleado como plataforma móvil del UGV. Fuente: [44]	49
Tabla 3-2 Resumen de las características del RPLIDAR C1. Fuente: [6].....	54
Tabla 3-3 Resumen de las características generales de la Cámara Sypro High Speed USB. Fuente: [46]	56
Tabla 3-4 Resumen de las características principales de la IMU empleada. Fuente: [17]	59
Tabla 3-5 Resumen de las características técnicas del GNSS empleado en el sistema. Fuente: [48]	63
Tabla 3-6 Resumen de las plataformas y sensores empleados en el sistema, indicando el tipo de cada elemento, su función principal y la interfaz de conexión con la unidad de procesamiento.....	64
Tabla 3-7 Comandos utilizados en la configuración de red y acceso remoto	74
Tabla 3-8 Resumen de los diferentes paquetes instalados mediante apt.	78
Tabla 3-9 Resumen de las diferentes librerías instaladas mediante pip.	79
Tabla 3-10 Resumen de los bloques funcionales del sistema.....	82
Tabla 3-11 Flujos de Información según sensor, tipo de dato, interfaz y estado.	84
Tabla 3-12 Resumen de los métodos de adquisición de datos multisensor, incluyendo las librerías empleadas y el tipo de información generada por cada sensor	89
Tabla 3-13 Relación de tópicos ROS 2 generados por el nodo multisensor, incluyendo tipo de mensaje, frecuencia de publicación y descripción funcional	90
Tabla 3-14 Frecuencias de adquisición de los sensores del sistema, mostrando la heterogeneidad temporal característica de arquitecturas multisensor	93
Tabla 3-15 Síntesis del proceso de desarrollo software del sistema multisensor, incluyendo fases, objetivos, herramientas empleadas y resultados obtenidos.....	97
Tabla 4-1 Resumen del proceso de validación de los sensores, incluyendo comportamiento observado, incidencias detectadas y conclusiones	106

Tabla 4-2 Resumen del rendimiento del sistema multisensor, incluyendo características operativas, impacto observado y posibles mejoras.....	107
Tabla 4-3 Problemas identificados durante el desarrollo del sistema y soluciones adoptadas.	109
Tabla 5-1 Evaluación del grado de cumplimiento de los objetivos	112

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Antecedentes

En las últimas décadas, la robótica móvil y los sistemas autónomos han experimentado un crecimiento notable gracias a los avances en electrónica, sensores y capacidad de procesamiento. En este escenario, los vehículos terrestres no tripulados (UGV, *Unmanned Ground Vehicles*) han adquirido una gran relevancia en ámbitos como la inspección de infraestructuras, la exploración de entornos complejos, la logística y las aplicaciones militares, al permitir reducir la exposición del personal humano a situaciones de riesgo [1], [2].

El funcionamiento de estos sistemas depende en gran medida de su capacidad para percibir el entorno, lo que se logra mediante la integración de múltiples sensores. Tecnologías como el LiDAR, las cámaras RGB, las unidades de medición inercial (IMU) y los sistemas de posicionamiento global (GNSS) permiten obtener información tanto del entorno como del propio estado del vehículo, siendo esenciales para tareas de navegación, localización y análisis del medio [3], [4].

En determinados sectores, como el de la defensa, estas tecnologías adquieren una importancia aún mayor, ya que los vehículos no tripulados permiten realizar tareas de reconocimiento, inspección o monitorización en entornos potencialmente peligrosos, reduciendo la exposición directa del personal. En este marco, el desarrollo de sistemas de percepción embarcados resulta clave para mejorar la autonomía y la capacidad operativa de estas plataformas.

Además, este trabajo se enmarca dentro de la evolución hacia sistemas cada vez más digitalizados, en los que herramientas como el diseño asistido por ordenador (*CAD*), la fabricación aditiva y los sistemas embebidos permiten desarrollar prototipos funcionales de forma rápida y eficiente. Gracias a ello, es posible validar soluciones en fases tempranas del desarrollo y trasladar conceptos propios de la industria a entornos experimentales y académicos.

A partir de este contexto, el presente Trabajo Fin de Grado aborda el diseño e implementación de un sistema multisensor embarcado en un vehículo terrestre no tripulado, integrando distintas tecnologías de percepción y procesamiento en una única plataforma.

1.2 Motivación

En la actualidad, el desarrollo de sistemas autónomos y semiautónomos constituye una de las principales líneas de innovación dentro del ámbito de la ingeniería, especialmente en aplicaciones vinculadas a la robótica móvil y a la percepción del entorno. La capacidad de un sistema para interpretar

su entorno mediante sensores es un factor decisivo para su funcionamiento, ya que condiciona tareas como la navegación, la monitorización y el análisis de datos.

Este proyecto surge con el propósito de profundizar, de forma práctica, en la integración de diferentes tecnologías de sensorización sobre una plataforma móvil real. La combinación de sensores como LiDAR, cámara, IMU y GNSS permite abordar el problema de la percepción desde un enfoque multisensorial, más robusto y representativo de aplicaciones reales.

Desde el punto de vista académico, este trabajo permite aplicar conocimientos adquiridos durante el grado en áreas como programación, diseño mecánico, electrónica y sistemas embebidos, integrándolos en un proyecto completo que abarca tanto el diseño hardware como la adquisición de datos. Asimismo, el uso de herramientas como Autodesk Inventor y la impresión 3D facilita el desarrollo de soluciones adaptadas a necesidades concretas, reforzando un enfoque práctico y orientado a la ingeniería aplicada.

Por último, el empleo de plataformas accesibles y componentes comerciales demuestra que es posible desarrollar sistemas experimentales de adquisición multisensor en entornos docentes y de investigación sin recurrir a soluciones industriales de alto coste.

1.3 Objetivos

El objetivo principal de este Trabajo Fin de Grado es el diseño e implementación de un sistema de adquisición de datos basado en múltiples sensores embarcados en una plataforma móvil terrestre.

Para alcanzar este objetivo general, se plantean los siguientes objetivos específicos:

- Seleccionar e integrar distintos sensores (LiDAR, cámara USB, IMU y GNSS) que permitan obtener información complementaria del entorno y del estado del vehículo.
- Diseñar, mediante software CAD (Autodesk Inventor), soportes específicos que permitan la correcta fijación de los sensores al chasis del vehículo.
- Fabricar los soportes diseñados mediante técnicas de impresión 3D, empleando materiales adecuados para prototipado, como PLA.
- Configurar la Raspberry Pi como unidad de procesamiento para la lectura, gestión y almacenamiento de los datos procedentes de los sensores.
- Integrar todos los componentes en la plataforma móvil, garantizando su funcionamiento conjunto y la estabilidad del sistema durante la operación.
- Evaluar el comportamiento del sistema integrado, identificando limitaciones y posibles mejoras tanto en la disposición de los sensores como en la calidad de los datos obtenidos.

El grado de cumplimiento de estos objetivos se evaluará en el capítulo 5 a partir de la integración funcional del sistema y de los resultados de validación obtenidos.

1.4 Estructura del trabajo

La memoria se organiza en los siguientes capítulos:

1. **Introducción y objetivos**, donde se presentan los antecedentes, la motivación y los objetivos del trabajo.
2. **Estado del arte**, en el que se revisan las principales tecnologías relacionadas con los sensores, robótica móvil y percepción empleadas en sistemas no tripulados.
3. **Desarrollo del TFG**, donde se describe el proceso de diseño, fabricación e integración del sistema multisensor, así como las herramientas y metodologías empleadas.
4. **Resultados y validación**, apartado en el que se analiza el funcionamiento del sistema a partir de las pruebas realizadas.

5. **Conclusiones y líneas futuras**, donde se sintetizan los principales resultados del trabajo y se plantean posibles mejoras.
6. **Bibliografía**, que recoge las referencias empleadas.
7. **Anexos**, donde se incorpora información complementaria relevante para la comprensión del proyecto.

2 ESTADO DEL ARTE

2.1 Introducción

El desarrollo de sistemas no tripulados ha experimentado un crecimiento notable en las últimas décadas debido a los avances en electrónica, sensores, capacidad de procesamiento y comunicaciones. En particular, los vehículos terrestres no tripulados (UGV, *Unmanned Ground Vehicles*) se han consolidado como plataformas de gran interés tanto en el ámbito civil como en el militar, ya que permiten operar en entornos potencialmente peligrosos o de difícil acceso, reduciendo la exposición directa del personal humano.

Uno de los elementos que más condiciona las capacidades de un UGV es su sistema de percepción, entendido como el conjunto de sensores y mecanismos de adquisición de información que permiten al vehículo conocer tanto su estado interno como las características del entorno que lo rodea. La calidad, fiabilidad e integración de estos sensores resultan determinantes en tareas como la navegación, la supervisión remota, la transmisión de datos en tiempo real y, en niveles más avanzados, la autonomía parcial o total del sistema.

En este contexto, el presente estado del arte tiene como finalidad revisar las principales tecnologías de percepción empleadas en sistemas no tripulados, con especial atención a aquellas más relevantes para un vehículo terrestre de bajo coste. Para ello, se analizan los sensores más utilizados, sus ventajas y limitaciones, así como distintos enfoques de integración, fusión de datos y percepción avanzada.

Asimismo, se incluyen apartados dedicados a tecnologías emergentes, al papel de la inteligencia artificial, a aplicaciones en el ámbito de la Armada y a las consideraciones normativas y de seguridad asociadas a este tipo de sistemas. Todo ello permite situar el trabajo desarrollado dentro del contexto actual de la robótica móvil y justificar las decisiones técnicas adoptadas en los capítulos posteriores.

En conjunto, este capítulo proporciona la base conceptual necesaria para comprender el diseño e implementación del sistema de sensores y transmisión de datos propuesto en este Trabajo Fin de Grado.

2.2 Vehículos terrestres no tripulados (UGV)

Los vehículos terrestres no tripulados (UGV, *Unmanned Ground Vehicles*) son plataformas móviles capaces de desplazarse sobre el terreno sin la presencia directa de un operador a bordo. Estos sistemas pueden operar de forma tele-controlada, semiautónoma o completamente autónoma, en función del nivel de desarrollo y de las tecnologías integradas [1], [2].

Los UGV han adquirido una gran relevancia en los últimos años debido a su capacidad para operar en entornos complejos o potencialmente peligrosos, donde el acceso humano resulta limitado o implica riesgos elevados. Entre sus principales aplicaciones se incluyen la inspección de infraestructuras, la exploración de entornos desconocidos, tareas logísticas y operaciones en el ámbito de la defensa.

Uno de los aspectos más críticos en el funcionamiento de estos sistemas es su capacidad de percepción del entorno. Para ello, los UGV incorporan distintos sensores que permiten obtener información sobre su posición, orientación y las características del entorno. La correcta integración de estos sensores resulta fundamental para garantizar un funcionamiento fiable, especialmente en tareas de navegación, supervisión y adquisición de datos.

En este contexto, los sistemas de sensores embarcados adquieren un papel clave, ya que determinan en gran medida las capacidades del vehículo. La combinación de sensores con características complementarias permite mejorar la robustez del sistema y compensar las limitaciones individuales de cada tecnología.

El sistema desarrollado en este Trabajo Fin de Grado se integra en una plataforma UGV, sobre la cual se ha implementado una arquitectura multisensor basada en la combinación de distintos dispositivos de adquisición de datos. Este enfoque permite analizar el comportamiento de los sensores en un entorno real y establecer una base para el desarrollo de aplicaciones más avanzadas en el ámbito de la robótica móvil.

2.3 Sensores empleados en sistemas no tripulados

Los sensores constituyen el elemento fundamental de los sistemas de percepción en vehículos no tripulados, ya que permiten adquirir información tanto del entorno como del estado del propio sistema. La elección y combinación de sensores determina en gran medida las capacidades del vehículo, así como sus limitaciones en términos de precisión, robustez y coste.

En robótica móvil terrestre es habitual emplear una arquitectura multisensor, en la que diferentes dispositivos proporcionan información complementaria. Esta estrategia permite mejorar la fiabilidad del sistema, compensando las limitaciones individuales de cada sensor mediante la redundancia y la diversidad de datos [5].

Se pondrá un especial énfasis en aquellos sensores utilizados en el sistema desarrollado en este Trabajo Fin de Grado, profundizando en su relevancia para vehículos terrestres no tripulados y justificando su selección frente a otras alternativas tecnológicas. Los sensores no empleados directamente se abordarán desde una perspectiva contextual, con el objetivo de ofrecer una visión global del estado actual de la tecnología sensorial en sistemas no tripulados y reforzar el carácter crítico y comparativo del análisis.

A continuación, se describen los sensores más empleados en sistemas no tripulados, analizando sus características principales, ventajas, limitaciones y aplicaciones en el contexto de los UGV.

2.3.1 Sensores LiDAR

Los sensores LiDAR (*Light Detection and Ranging*) constituyen una de las tecnologías más empleadas en los sistemas de percepción de vehículos terrestres no tripulados debido a su capacidad para medir distancias con alta precisión y generar representaciones geométricas del entorno en tiempo real. Su principio de funcionamiento se basa en la emisión de pulsos de radiación láser y la posterior medición del tiempo que tarda dicha señal en reflejarse en un objeto y regresar al sensor.

A partir del tiempo de vuelo de la señal emitida y conociendo la velocidad de propagación de la luz, el sistema LiDAR es capaz de calcular la distancia al objeto detectado. Mediante la repetición de este proceso en múltiples direcciones, el sensor genera un conjunto de medidas espaciales que pueden representarse como una nube de puntos, proporcionando información detallada sobre la geometría del entorno circundante [4], [6]. A continuación, en la Figura 2-1 se muestra un esquema básico del funcionamiento previamente explicado.

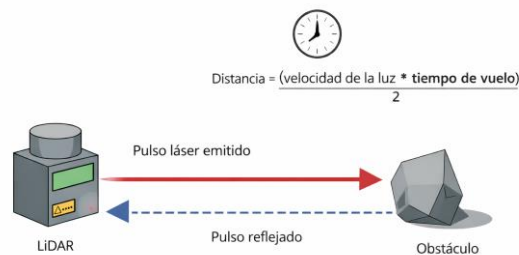


Figura 2-1 El funcionamiento básico de un sensor LiDAR. Fuente: Elaboración asistida por herramientas de inteligencia artificial

Principio de funcionamiento

En los sistemas LiDAR más habituales, especialmente en aplicaciones de robótica móvil, el sensor emite pulsos láser en el espectro infrarrojo cercano. Un sistema óptico y mecánico, generalmente basado en espejos giratorios o cabezales rotativos, permite barrer el entorno en uno o varios planos. Cada pulso reflejado es detectado por un fotodiodo, y el tiempo transcurrido entre la emisión y la recepción se utiliza para calcular la distancia correspondiente.

Dependiendo de su arquitectura, los sensores LiDAR pueden clasificarse principalmente en LiDAR bidimensionales (2D) y LiDAR tridimensionales (3D). Los LiDAR 2D realizan el barrido en un único plano, normalmente horizontal, generando una representación plana del entorno a la altura de montaje del sensor. Por su parte, los LiDAR 3D realizan múltiples barridos en distintos planos, permitiendo la reconstrucción volumétrica del entorno [7].

Aplicaciones del LiDAR en vehículos terrestres no tripulados

En el contexto de los UGV, los sensores LiDAR se emplean principalmente para tareas de detección de obstáculos, navegación, mapeado del entorno y localización. En aplicaciones de navegación, los datos LiDAR permiten identificar obstáculos cercanos y estimar distancias con gran precisión, lo que resulta fundamental para evitar colisiones y planificar trayectorias seguras.

Los sensores LiDAR 2D son ampliamente utilizados en plataformas terrestres de bajo y medio coste, debido a su simplicidad, fiabilidad y menor requerimiento computacional. Estos sensores proporcionan información suficiente para aplicaciones como el mapeado bidimensional de entornos interiores, la navegación en pasillos o la detección de obstáculos a nivel del suelo.

Por otro lado, los sensores LiDAR 3D ofrecen una representación más completa del entorno, permitiendo detectar objetos a distintas alturas y generar mapas tridimensionales detallados. Sin embargo, su mayor complejidad, coste económico y demanda de procesamiento hacen que su uso se reserve normalmente para aplicaciones más avanzadas o plataformas con mayores capacidades computacionales.

Ventajas y limitaciones

Entre las principales ventajas de los sensores LiDAR destaca su alta precisión en la medición de distancias, su independencia de las condiciones de iluminación y su capacidad para generar información geométrica estructurada del entorno. A diferencia de los sensores basados en visión, el rendimiento del LiDAR no se ve afectado de forma significativa por cambios bruscos de luz o ausencia de iluminación.

No obstante, los sensores LiDAR también presentan limitaciones relevantes. Su rendimiento puede verse degradado en condiciones ambientales adversas, como lluvia intensa, niebla o polvo en suspensión, que pueden provocar reflexiones no deseadas o pérdidas de señal. Además, los sensores LiDAR tienen dificultades para detectar superficies altamente reflectantes o transparentes, como cristales, lo que puede generar errores en la percepción del entorno.

Otra limitación importante es el campo de visión restringido en los sensores LiDAR 2D, que únicamente proporcionan información en el plano de escaneado. Esto implica que obstáculos situados fuera de dicho plano pueden no ser detectados, haciendo necesaria la integración con otros sensores o el uso de configuraciones de montaje específicas [3], [4].

Relevancia del LiDAR en el sistema propuesto

En el sistema desarrollado en este Trabajo Fin de Grado, el sensor LiDAR desempeña un papel fundamental dentro del sistema de percepción del vehículo terrestre no tripulado. Su capacidad para medir distancias con precisión y generar información estructurada del entorno lo convierte en un sensor idóneo para tareas de supervisión y análisis del entorno en tiempo real.

La elección de un sensor LiDAR frente a otras tecnologías se justifica por su equilibrio entre precisión, robustez y coste, así como por la disponibilidad de sensores comerciales compatibles con plataformas de procesamiento embarcado de bajo consumo, como la Raspberry Pi. En combinación con otros sensores, el LiDAR contribuye a mejorar la fiabilidad global del sistema de sensores y sienta las bases para posibles ampliaciones futuras orientadas a la navegación autónoma.

2.3.2 Sensores Radar

Los sensores radar (*Radio Detection and Ranging*) emplean ondas electromagnéticas para la detección y localización de objetos en el entorno. A diferencia de los sensores basados en radiación óptica, como el LiDAR o las cámaras, el radar opera en bandas de radiofrecuencia, lo que le confiere características particulares en términos de alcance, robustez y comportamiento frente a condiciones ambientales adversas [4], [8].

En vehículos terrestres no tripulados, el radar se emplea principalmente como sensor complementario para la detección de obstáculos y la estimación de distancias en entornos complejos.

Principio de funcionamiento

El funcionamiento de un sensor radar se basa en la emisión de ondas electromagnéticas que se propagan por el entorno hasta incidir sobre un objeto, reflejándose parcialmente hacia el sensor. A partir del análisis de la señal reflejada, el sistema es capaz de estimar la distancia al objeto detectado y, en algunos casos, su velocidad relativa.

La distancia se calcula generalmente mediante la medición del tiempo de propagación de la señal emitida, mientras que la velocidad puede estimarse a partir del efecto Doppler, que se manifiesta como una variación en la frecuencia de la señal reflejada cuando el objeto se encuentra en movimiento. Estos principios permiten al radar detectar objetos a distancias superiores a las alcanzables por sensores ópticos simples.

Aplicaciones en robótica móvil terrestre

En el ámbito de los UGV, los sensores radar se utilizan principalmente para la detección de obstáculos a medio y largo alcance, así como para la supervisión del entorno en condiciones en las que otros sensores presentan limitaciones significativas. Su capacidad para operar de manera fiable en presencia de lluvia, niebla, polvo o condiciones de baja visibilidad los convierte en una tecnología especialmente valiosa en entornos adversos.

El radar es habitual en aplicaciones de seguridad, defensa y automoción, donde se emplea como sistema redundante para mejorar la robustez del sistema de percepción. En robótica terrestre, su integración suele realizarse junto a sensores LiDAR y cámaras, aprovechando la complementariedad entre tecnologías.

Ventajas y limitaciones

Entre las principales ventajas de los sensores radar destaca su alta robustez frente a condiciones ambientales adversas, así como su capacidad para detectar objetos a distancias relativamente grandes. Además, el radar no depende de las condiciones de iluminación y puede proporcionar información de velocidad, lo que resulta útil en escenarios dinámicos.

No obstante, los sensores radar presentan limitaciones importantes cuando se emplean como único sistema de percepción. Su resolución espacial es inferior a la de sensores LiDAR o cámaras, lo que dificulta la identificación precisa de la forma y dimensiones de los objetos detectados. Asimismo, la interpretación de los datos radar suele requerir algoritmos de procesamiento más complejos para extraer información útil del entorno.

Relación con el sistema propuesto

En el sistema desarrollado en este Trabajo Fin de Grado, los sensores radar no se incluyen como parte del conjunto principal de sensores del UGV. Esta decisión se justifica por la necesidad de priorizar sensores capaces de proporcionar información geométrica detallada del entorno a distancias cortas y medias, como el LiDAR y las cámaras, así como por las limitaciones de coste y complejidad asociadas a los sistemas radar.

No obstante, el estudio de esta tecnología resulta relevante para comprender su papel como sensor complementario en sistemas de percepción avanzados y para contextualizar las decisiones de diseño adoptadas en el presente trabajo.

2.3.3 Cámaras RGB

Las cámaras RGB constituyen uno de los sensores más utilizados en sistemas no tripulados debido a su capacidad para proporcionar información visual rica del entorno. A diferencia de sensores como el LiDAR o el radar, que ofrecen principalmente información geométrica o de distancia, las cámaras permiten capturar detalles de color, textura y forma, facilitando la interpretación visual del entorno tanto por operadores humanos como por algoritmos de procesamiento.

En vehículos terrestres no tripulados, las cámaras RGB se emplean ampliamente en tareas de supervisión remota, detección de obstáculos, reconocimiento de objetos y asistencia a la navegación, especialmente en aplicaciones donde la intervención humana o la interpretación visual directa resulta necesaria [4].

Principio de funcionamiento

Una cámara RGB se basa en la captación de radiación electromagnética en el espectro visible, separando la información luminosa en tres componentes principales: rojo (R), verde (G) y azul (B). Mediante la combinación de estos tres canales, el sensor es capaz de generar imágenes en color que representan fielmente la escena capturada, tal y como se muestra en la Figura 2-2.

El sistema óptico de la cámara dirige la luz incidente hacia un sensor de imagen, normalmente de tipo CCD o CMOS, compuesto por una matriz de fotodiodos. Cada fotodiodo convierte la luz recibida en una señal eléctrica proporcional a su intensidad, que posteriormente es digitalizada y procesada para formar la imagen final [9].

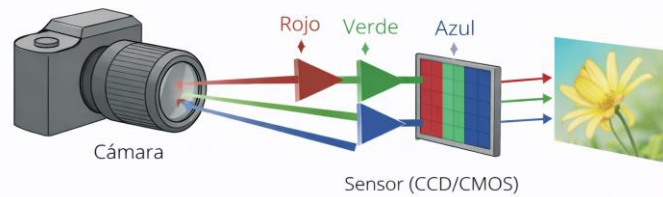


Figura 2-2 Esquema de funcionamiento de una cámara RGB digital. Fuente: Elaboración asistida por herramientas de inteligencia artificial

Aplicaciones en robótica móvil terrestre

En el contexto de los UGV, las cámaras RGB se utilizan tanto para la visualización en tiempo real del entorno como para el desarrollo de sistemas de visión artificial. En aplicaciones de tele-operación, la cámara actúa como el principal canal de percepción del operador, permitiendo supervisar el movimiento del vehículo y detectar posibles riesgos.

Asimismo, las cámaras RGB pueden emplearse en tareas de detección y seguimiento de objetos, identificación de marcas visuales o reconocimiento del entorno, especialmente cuando se combinan con técnicas de procesamiento de imagen o aprendizaje automático. No obstante, el uso exclusivo de cámaras para navegación presenta limitaciones importantes, lo que ha impulsado su integración con otros sensores complementarios.

Ventajas y limitaciones

Entre las principales ventajas de las cámaras RGB se encuentran su bajo coste, su alta resolución y la gran cantidad de información visual que proporcionan. Además, se trata de sensores ligeros y de fácil integración, compatibles con plataformas de procesamiento embarcado de bajo consumo.

Sin embargo, las cámaras RGB presentan limitaciones significativas. Su rendimiento depende en gran medida de las condiciones de iluminación, pudiendo verse afectado por sombras, deslumbramientos o cambios bruscos de luz. Asimismo, carecen de información directa de profundidad, lo que dificulta la estimación precisa de distancias sin el apoyo de algoritmos adicionales o sensores complementarios.

Estas limitaciones hacen que, en aplicaciones críticas, las cámaras se utilicen generalmente en combinación con sensores como el LiDAR o la IMU, con el fin de mejorar la robustez y fiabilidad del sistema de percepción.

Relación con el sistema propuesto

En el sistema desarrollado en este Trabajo Fin de Grado, la cámara RGB desempeña un papel relevante como sensor de supervisión visual y adquisición de información del entorno. Su integración permite transmitir imágenes en tiempo real a una estación base, facilitando el control remoto del vehículo y la evaluación de su entorno operativo.

La elección de una cámara RGB se justifica por su versatilidad, su compatibilidad con plataformas como la Raspberry Pi, y su utilidad como sensor complementario dentro de un sistema de sensores multimodal.

2.3.4 Cámaras Infrarrojas/Térmicas

Las cámaras infrarrojas o térmicas son sensores capaces de detectar la radiación electromagnética emitida por los objetos en el espectro infrarrojo, permitiendo obtener información relacionada con su temperatura superficial. A diferencia de las cámaras RGB, que dependen de la iluminación visible

reflejada, las cámaras térmicas generan imágenes a partir de la emisión térmica propia de los cuerpos, lo que les confiere características diferenciadas en determinadas aplicaciones.

En sistemas no tripulados, este tipo de cámaras resulta especialmente útil en entornos donde la visibilidad es limitada o nula, como condiciones de baja iluminación, presencia de humo, niebla o durante operaciones nocturnas. Por este motivo, las cámaras térmicas son ampliamente empleadas en aplicaciones de seguridad, vigilancia y defensa [4].

Principio de funcionamiento

El funcionamiento de una cámara térmica se basa en la detección de radiación infrarroja emitida por los objetos en función de su temperatura. Esta radiación es captada por un sensor especializado, generalmente de tipo microbolómetro, que convierte las variaciones de energía térmica en señales eléctricas.

Cada elemento sensible del sensor responde a la radiación incidente, permitiendo construir una imagen térmica en la que los distintos niveles de intensidad representan variaciones de temperatura. A diferencia de las cámaras RGB, las cámaras infrarrojas no requieren una fuente de iluminación externa y pueden operar de forma eficaz en completa oscuridad.

Aplicaciones en robótica móvil terrestre

En vehículos terrestres no tripulados, las cámaras térmicas se emplean principalmente para detección de personas o animales, vigilancia, búsqueda y rescate y supervisión en entornos degradados visualmente. Su capacidad para distinguir fuentes de calor las hace especialmente valiosas en escenarios donde las cámaras RGB presentan limitaciones significativas [5].

En el ámbito militar y de seguridad, estas cámaras permiten identificar objetivos ocultos o camuflados, así como realizar operaciones nocturnas con mayor fiabilidad. No obstante, su uso en robótica civil suele verse condicionado por su mayor coste económico y menor resolución espacial en comparación con las cámaras RGB.

Ventajas y limitaciones

Entre las principales ventajas de las cámaras infrarrojas destaca su independencia de las condiciones de iluminación, así como su capacidad para detectar diferencias térmicas que no son visibles en el espectro óptico. Esto las convierte en sensores especialmente robustos para aplicaciones críticas.

Por el contrario, presentan limitaciones relevantes, como su menor resolución, la dificultad para identificar detalles geométricos precisos y su elevado coste en comparación con cámaras convencionales. Además, las imágenes térmicas pueden resultar menos intuitivas para operadores no entrenados, lo que limita su uso en sistemas de teleoperación básica.

Las principales diferencias entre las cámaras RGB y térmicas se resumen en la Tabla 2-1 a continuación.

Característica	Cámara RGB	Cámara infrarroja / térmica
Espectro de operación	Visible ($\approx 400\text{--}700\text{ nm}$)	Infrarrojo ($\approx 8\text{--}14\text{ }\mu\text{m}$)
Dependencia de la iluminación	Alta	Nula
Información proporcionada	Color, textura y forma	Distribución térmica
Resolución espacial	Alta	Media o baja
Información de temperatura	No	Sí
Capacidad de detección nocturna	Muy limitada	Muy alta

Característica	Cámara RGB	Cámara infrarroja / térmica
Sensibilidad a condiciones ambientales	Alta (sombras, deslumbramientos)	Baja (excepto lluvia intensa)
Interpretación por operador humano	Intuitiva	Menos intuitiva
Coste económico	Bajo	Alto
Uso típico en UGV	Supervisión visual, tele-operación	Vigilancia, detección térmica
Papel en el sistema	Sensor principal de visión	Sensor complementario

Tabla 2-1 Comparativa de las principales características de las cámaras RGB y las cámaras infrarrojas/térmicas en el contexto de sistemas no tripulados

Relación con el sistema propuesto

En el sistema desarrollado en este Trabajo Fin de Grado, las cámaras infrarrojas no se incluyen como parte del conjunto principal de sensores del UGV. Esta decisión se debe a la orientación del sistema hacia tareas de adquisición y transmisión de información visual general, así como a las restricciones de coste y complejidad del sistema.

No obstante, su análisis resulta relevante desde un punto de vista comparativo y contextual, especialmente en relación con aplicaciones militares y de vigilancia, donde este tipo de sensores desempeña un papel fundamental.

2.3.5 Sensores Sonar

Los sensores sonar (*Sound Navigation and Ranging*) emplean ondas acústicas para la detección de objetos y la medición de distancias en un entorno determinado. Su funcionamiento se basa en la emisión de pulsos sonoros y en el análisis del eco reflejado por los objetos presentes en el medio, permitiendo estimar su distancia en función del tiempo de propagación de la señal acústica.

Este tipo de sensores ha sido ampliamente utilizado en entornos acuáticos, donde la propagación del sonido resulta más eficiente que la de las ondas electromagnéticas. Por este motivo, los sistemas sonar constituyen una tecnología fundamental en aplicaciones marítimas y submarinas, especialmente en vehículos no tripulados de superficie y submarinos.

Principio de funcionamiento

El principio de funcionamiento de un sensor sonar se basa en la emisión de una onda acústica a través de un transductor, que se propaga por el medio hasta incidir sobre un objeto. Parte de esta onda se refleja y regresa al sensor, donde es detectada y analizada. Conociendo la velocidad de propagación del sonido en el medio, el sistema puede calcular la distancia al objeto detectado a partir del tiempo transcurrido entre la emisión y la recepción del eco.

La precisión del sonar depende de factores como la frecuencia de la señal acústica, las características del medio de propagación y la geometría del entorno. En entornos terrestres, la propagación del sonido se ve afectada por la atenuación en el aire y la presencia de ruido ambiental, lo que limita el alcance y la fiabilidad de este tipo de sensores [10], [11].

Aplicaciones en robótica móvil

En robótica móvil terrestre, los sensores sonar se han utilizado tradicionalmente como sensores de proximidad, especialmente en sistemas de bajo coste y plataformas educativas. Su simplicidad de

implementación y su capacidad para detectar obstáculos cercanos los convierten en una solución básica para evitar colisiones a corta distancia.

No obstante, el uso del sonar en UGV presenta limitaciones significativas en comparación con otros sensores más avanzados. La baja resolución espacial, la sensibilidad al ruido ambiental y la dificultad para discriminar la forma y orientación de los objetos reducen su utilidad en aplicaciones de navegación avanzada o percepción detallada del entorno.

Ventajas y limitaciones

Entre las ventajas de los sensores sonar destacan su bajo coste, su simplicidad de integración y su capacidad para detectar objetos a corta distancia independientemente de las condiciones de iluminación. Estas características han favorecido su uso en sistemas sencillos y plataformas experimentales.

Sin embargo, presentan limitaciones importantes, como su alcance reducido, su baja precisión en la estimación de distancias y su elevada sensibilidad a interferencias acústicas. Además, en entornos abiertos o ruidosos, el rendimiento del sonar puede verse significativamente degradado, lo que limita su fiabilidad en aplicaciones reales de robótica terrestre.

Relación con el sistema propuesto

En el sistema desarrollado en este Trabajo Fin de Grado, los sensores sonar no se emplean como parte del conjunto principal de sensores del UGV. Esta decisión se justifica por las limitaciones inherentes de esta tecnología en entornos terrestres.

No obstante, su inclusión en el estado del arte resulta relevante para contextualizar las distintas tecnologías de detección disponibles y justificar la selección de sensores más adecuados para los objetivos del sistema propuesto.

2.3.6 Sistemas de posicionamiento global (GPS/GNSS)

Los sistemas de posicionamiento global, comúnmente conocidos como GPS (*Global Positioning System*), forman parte de un conjunto más amplio de sistemas de navegación por satélite denominados GNSS (*Global Navigation Satellite Systems*). Estos sistemas permiten determinar la posición de un receptor en la superficie terrestre mediante la recepción de señales emitidas por una constelación de satélites.

En vehículos terrestres no tripulados, los sistemas GPS constituyen una fuente fundamental de información para la estimación de la posición global, especialmente en entornos exteriores. Su integración resulta esencial en aplicaciones de navegación, seguimiento de trayectorias y transmisión de datos de localización a una estación base.

Principio de funcionamiento

El funcionamiento de un sistema GNSS se basa en la recepción simultánea de señales procedentes de varios satélites cuya posición es conocida con gran precisión. Cada satélite transmite información temporal y orbital, permitiendo al receptor calcular la distancia a cada uno de ellos mediante la medición del tiempo que tarda la señal en propagarse desde el satélite hasta el receptor [12].

A partir de estas distancias, el receptor GNSS emplea técnicas de trilateración para determinar su posición en coordenadas geográficas tal y como se muestra en la Figura 2-3. Para obtener una solución completa en tres dimensiones (latitud, longitud y altitud), así como la corrección del error de reloj interno del receptor, es necesario recibir señales de al menos cuatro satélites [13], [14].

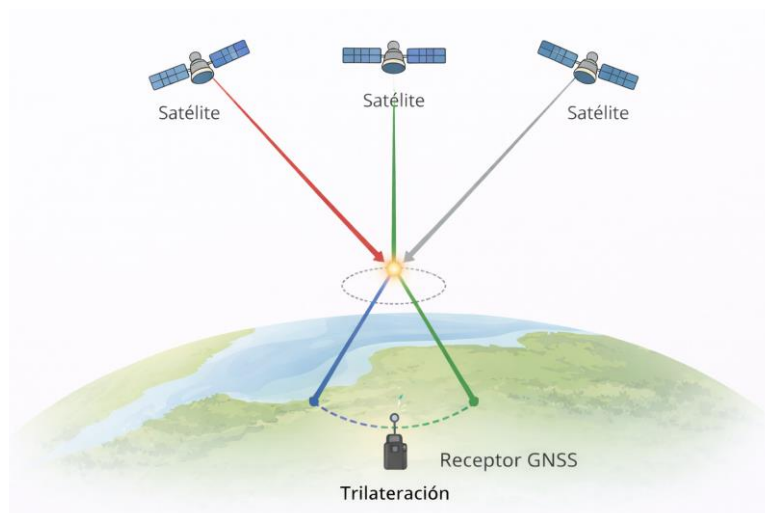


Figura 2-3 Esquema conceptual del funcionamiento de un sistema GNSS basado en la recepción de señales de varios satélites y el cálculo de la posición del receptor mediante el principio de trilateración. Fuente: Elaboración asistida por herramientas de inteligencia artificial

Aplicaciones en robótica móvil terrestre

En el ámbito de los UGV, los sistemas GNSS se utilizan principalmente para la localización global del vehículo, el registro de trayectorias y la supervisión remota de su desplazamiento. Esta información resulta especialmente útil en entornos abiertos, donde el acceso a señales satelitales no se ve obstruido por edificaciones u obstáculos significativos.

El GNSS permite además la integración del vehículo en sistemas de información geográfica y facilita la transmisión de datos de posición en tiempo real, lo que resulta de gran utilidad en aplicaciones de telemetría, monitorización y análisis posterior del recorrido realizado por el UGV.

Ventajas y limitaciones

Entre las principales ventajas de los sistemas GNSS destaca su capacidad para proporcionar una referencia de posición absoluta, independiente de la trayectoria previa del vehículo. Asimismo, se trata de una tecnología ampliamente extendida, con receptores comerciales de bajo coste y fácil integración en sistemas embarcados.

No obstante, los sistemas GNSS presentan limitaciones importantes. Su precisión puede verse degradada por errores atmosféricos, multitrayectoria de la señal o pérdidas de cobertura en entornos urbanos, boscosos o interiores. Además, la frecuencia de actualización y la latencia de las mediciones pueden resultar insuficientes para aplicaciones que requieran alta precisión dinámica [13].

Estas limitaciones hacen que, en sistemas de robótica móvil, el GNSS se utilice habitualmente en combinación con otros sensores, como las unidades de medida inercial, con el objetivo de mejorar la robustez y continuidad de la estimación de la posición.

Relación con el sistema propuesto

En el sistema desarrollado en este Trabajo Fin de Grado, el sistema GNSS constituye uno de los sensores principales para la obtención de información de posicionamiento global del UGV. Su integración permite conocer la ubicación del vehículo en tiempo real y transmitir dicha información a una estación base, facilitando tareas de supervisión y análisis del movimiento.

La elección de un receptor GNSS se justifica por su simplicidad de integración, su compatibilidad con plataformas como la Raspberry Pi, y su papel complementario dentro de un sistema de sensores multimodal, especialmente cuando se combina con sensores inerciales.

2.3.7 Unidades de medida inercial (IMU)

Las Unidades de Medida Inercial (IMU, *Inertial Measurement Units*) son sensores encargados de medir el movimiento y la orientación de un sistema mediante la adquisición de aceleraciones lineales y velocidades angulares. En vehículos terrestres no tripulados, las IMU constituyen un componente fundamental para la estimación del estado dinámico del vehículo, especialmente en situaciones en las que otras fuentes de información, como el GNSS, presentan limitaciones.

Una IMU típica integra acelerómetros y giroscopios en los tres ejes del espacio, y en algunos casos incluye también un magnetómetro, lo que permite obtener información adicional sobre la orientación absoluta respecto al campo magnético terrestre. Esta combinación de sensores proporciona una medida continua del movimiento del vehículo, independiente de referencias externas.

Principio de funcionamiento

Los acelerómetros miden la aceleración lineal específica experimentada por el sensor en cada eje, mientras que los giroscopios miden la velocidad angular de rotación alrededor de cada eje. A partir de estas mediciones, es posible estimar la orientación y los cambios de velocidad del sistema mediante procesos de integración temporal.

Sin embargo, debido a la presencia de ruido y errores sistemáticos, las mediciones inerciales presentan una deriva acumulativa con el tiempo. Este fenómeno hace que la estimación de posición y orientación basada únicamente en una IMU se degrade progresivamente, lo que limita su uso como fuente única de navegación a largo plazo.

El magnetómetro, cuando está presente, proporciona una referencia adicional para la orientación, aunque su fiabilidad puede verse afectada por perturbaciones magnéticas del entorno, especialmente en plataformas móviles con componentes metálicos o sistemas eléctricos cercanos [15], [16], [17].

Aplicaciones en robótica móvil terrestre

En UGV, las IMU se emplean principalmente para la estimación de la orientación, la detección de movimientos y la mejora de la estabilidad del sistema. Asimismo, permiten conocer la actitud del vehículo frente a irregularidades del terreno, lo que resulta relevante para el control del movimiento y la interpretación de otros sensores.

Una de las aplicaciones más habituales de las IMU en robótica móvil es su integración con sistemas GNSS mediante técnicas de fusión sensorial, con el objetivo de compensar las limitaciones de cada sensor. Mientras el GNSS proporciona una referencia absoluta de posición, la IMU ofrece información continua y de alta frecuencia sobre el movimiento del vehículo, permitiendo mejorar la robustez y continuidad de la estimación del estado [15], [16].

Ventajas y limitaciones

Entre las principales ventajas de las IMU destaca su independencia de referencias externas, su alta frecuencia de actualización y su bajo consumo energético. Estas características las convierten en sensores especialmente adecuados para sistemas embarcados y plataformas móviles de bajo coste.

No obstante, presentan limitaciones significativas, siendo la más relevante la deriva acumulativa de las estimaciones debido al ruido y a los errores de calibración. Además, la precisión de una IMU depende en gran medida de la calidad del sensor, existiendo diferencias notables entre dispositivos de bajo coste y soluciones de grado industrial o militar [15], [16], [17].

Relación con el sistema propuesto

En el sistema desarrollado en este Trabajo Fin de Grado, la IMU constituye uno de los sensores principales para la obtención de información sobre la orientación y el movimiento del UGV. Su integración resulta esencial para complementar la información proporcionada por el sistema GNSS y mejorar la continuidad y fiabilidad de la estimación del estado del vehículo.

La elección de una IMU se justifica por su facilidad de integración, su compatibilidad con plataformas de procesamiento embarcado como la Raspberry Pi, y su papel clave dentro de un sistema de sensores multimodal orientado a la adquisición y transmisión de datos en tiempo real.

2.3.8 Sistema AIS

El Sistema de Identificación Automática (AIS, *Automatic Identification System*) es un sistema de comunicación y posicionamiento diseñado originalmente para el seguimiento e identificación de embarcaciones. Su uso está ampliamente extendido en el ámbito marítimo como herramienta para mejorar la seguridad de la navegación y la gestión del tráfico naval [18].

El AIS permite a las plataformas equipadas con este sistema transmitir y recibir información relativa a su identidad, posición, rumbo y velocidad, facilitando el conocimiento situacional entre buques y estaciones costeras [18]. Aunque su aplicación principal se encuentra en el entorno marítimo, su estudio resulta relevante dentro del estado del arte por su uso en sistemas no tripulados navales y su importancia en el ámbito de la Armada.

Principio de funcionamiento

El sistema AIS se basa en la transmisión periódica de mensajes digitales a través de canales de radio VHF (*Very High Frequency*), empleando un esquema de acceso múltiple por división en el tiempo. Cada estación AIS emite información que es recibida por otros buques cercanos y por estaciones terrestres, permitiendo construir una imagen compartida del tráfico marítimo en una determinada zona.

La información transmitida incluye datos estáticos (identificador del buque), dinámicos (posición GNSS, velocidad, rumbo) y datos relacionados con el viaje. El sistema está diseñado para operar de forma automática y continua, sin intervención directa del operador [18].

Aplicaciones en sistemas no tripulados

En el ámbito de los sistemas no tripulados, el AIS se emplea principalmente en USV y en plataformas navales autónomas o tele-operadas. Su integración permite mejorar la conciencia situacional y garantizar la interoperabilidad con otros buques tripulados que operan en la misma zona.

En aplicaciones militares y de seguridad marítima, el AIS constituye una herramienta fundamental para la monitorización del entorno naval, aunque su uso puede verse restringido o modificado en función de los requisitos operativos y de confidencialidad.

Ventajas y limitaciones

Entre las ventajas del sistema AIS destaca su amplia implantación, su capacidad para proporcionar información de identificación y posicionamiento de forma estandarizada y su interoperabilidad con infraestructuras existentes.

Sin embargo, presenta limitaciones significativas, como su dependencia de la transmisión voluntaria de información, su vulnerabilidad a interferencias o su escasa aplicabilidad fuera del entorno marítimo. Además, el AIS no está diseñado para proporcionar información detallada del entorno físico, sino únicamente datos relativos a otras plataformas equipadas con el sistema.

Relación con el sistema propuesto

En el sistema desarrollado en este Trabajo Fin de Grado, el sistema AIS no se emplea, dado que el vehículo considerado es un UGV destinado a operar en entornos terrestres. No obstante, aunque no forman parte del sistema implementado, se incluyen por su relevancia comparativa en el ámbito naval y multisensor.

2.3.9 Sistema DVL

El *Doppler Velocity Log* (DVL) es un sensor utilizado principalmente en entornos submarinos para medir la velocidad de un vehículo con respecto al fondo marino o a una masa de agua. Su funcionamiento se basa en el efecto Doppler, permitiendo estimar la velocidad relativa a partir del desplazamiento en frecuencia de una señal acústica reflejada.

El DVL constituye un sensor fundamental en vehículos submarinos no tripulados, donde la ausencia de señales GNSS hace necesaria la utilización de sensores alternativos para la navegación y la estimación del movimiento.

Principio de funcionamiento

El funcionamiento de un sistema DVL se basa en la emisión de ondas acústicas hacia el fondo marino mediante varios transductores dispuestos con un determinado ángulo respecto al eje del vehículo. Al reflejarse dichas ondas en el fondo, el sensor analiza el desplazamiento en frecuencia de la señal recibida, lo que permite calcular la componente de velocidad relativa del vehículo [10], [19], [20].

A partir de la información proporcionada por varios haces acústicos, el sistema DVL puede estimar la velocidad tridimensional del vehículo respecto al fondo.

Aplicaciones en sistemas no tripulados

El DVL se emplea de forma predominante en vehículos submarinos no tripulados (AUV) y en vehículos operados remotamente (ROV), donde permite obtener información precisa de velocidad y desplazamiento en ausencia de referencias externas. Su uso resulta especialmente relevante en operaciones de exploración submarina, inspección de infraestructuras y aplicaciones militares.

En estos sistemas, el DVL constituye uno de los pilares de la navegación submarina, complementando a otros sensores como la IMU y el sonar para mejorar la precisión y estabilidad de la estimación del estado del vehículo.

Ventajas y limitaciones

Entre las principales ventajas del DVL destaca su capacidad para proporcionar medidas de velocidad precisas respecto al fondo, incluso en ausencia de referencias externas como señales satelitales. Asimismo, su integración con sistemas inerciales permite reducir significativamente la deriva en la estimación de posición en vehículos submarinos.

No obstante, el DVL presenta limitaciones importantes, como su dependencia de la presencia de un fondo adecuado para la reflexión de la señal acústica y su elevado coste económico. Además, su aplicación se restringe prácticamente en exclusiva al entorno submarino, resultando inapropiado para sistemas terrestres o aéreos.

Relación con el sistema propuesto

En el sistema desarrollado en este Trabajo Fin de Grado, el sensor DVL no se emplea, dado que se trata de una tecnología específica para entornos submarinos y no resulta aplicable a vehículos terrestres no tripulados. Su inclusión en el estado del arte responde a la necesidad de ofrecer una visión completa de los sensores empleados en sistemas no tripulados dentro del ámbito naval y de la Armada.

2.3.10 Síntesis de los diferentes sensores empleados en sistemas de percepción

Con el objetivo de comparar de forma estructurada los sensores empleados en sistemas de percepción, se presenta la Tabla 2-2 donde se recogen sus principales características, así como su papel dentro de arquitecturas de fusión sensorial.

Sensor	Tipo de Información	Alcance Típico	Precisión/ Resolución	Dependencia del entorno	Coste Relativo	Rol en fusión sensorial
LiDAR	Distancia y geometría 3D	Medio (50–200 m)	Alta	Media (niebla, lluvia, polvo)	Medio–alto	Mapeado 3D y percepción precisa del entorno
Sensor láser	Distancia puntual	Corto	Media	Media	Bajo	Medición simple de proximidad
Radar	Distancia y velocidad	Medio–largo	Baja	Baja (muy robusto)	Alto	Detección robusta en condiciones adversas
Cámara RGB	Imagen (color, forma)	Variable	Alta	Alta (iluminación, clima)	Bajo	Clasificación y reconocimiento de objetos
Cámara térmica	Radiación infrarroja	Medio	Media–baja	Baja	Alto	Detección en baja visibilidad/nocturna
Sonar	Distancia acústica	Corto (2–10 m)	Baja	Alta (ruido, superficie)	Bajo	Detección de proximidad
GNSS	Posición global	Global	Media	Alta (obstáculos, interferencias)	Bajo	Posicionamiento absoluto
IMU	Aceleración y orientación	—	Media	Muy baja (independiente)	Bajo	Estimación de movimiento
AIS	Identificación y posición	Largo	Media	Baja	Medio	Uso marítimo (no aplicable en UGV)
DVL	Velocidad relativa	Corto–medio	Alta	Media	Alto	Uso submarino (no aplicable en UGV)

Tabla 2-2 Comparativa de sensores utilizados en sistemas de percepción para vehículos autónomos, incluyendo sus características técnicas y su contribución en estrategias de fusión sensorial

Como se observa, cada sensor presenta ventajas y limitaciones específicas, lo que refuerza la necesidad de emplear técnicas de fusión sensorial para obtener una percepción más robusta y fiable del entorno.

2.4 Fusión Sensorial

La integración y fusión de sensores constituye un elemento fundamental en el diseño de sistemas no tripulados modernos, ya que permite mejorar la percepción del entorno y del estado del sistema mediante la combinación de múltiples fuentes de información.

En vehículos terrestres no tripulados (UGV), los sensores presentan limitaciones inherentes derivadas de su principio de funcionamiento, resolución, alcance y dependencia de las condiciones del entorno. Por ello, el uso aislado de un único sensor resulta insuficiente para garantizar una percepción

robusta en escenarios reales. En este contexto, la integración y fusión sensorial permiten aumentar la precisión, la fiabilidad y la continuidad de la información disponible [4], [21], [22].

Integración de sensores

La integración de sensores hace referencia al proceso de incorporación de distintos dispositivos de adquisición de datos dentro de una arquitectura común. Este proceso abarca tanto aspectos hardware como software.

Desde el punto de vista hardware, incluye la disposición física de los sensores, su alimentación y su conexión a la unidad de procesamiento. Desde el punto de vista software, implica la gestión de los datos, su sincronización temporal y su transmisión dentro del sistema, por ejemplo, mediante el uso de middleware como ROS 2.

En plataformas móviles terrestres, la integración sensorial debe considerar restricciones de espacio, peso y consumo energético, así como la compatibilidad entre dispositivos y la capacidad de procesamiento del sistema embarcado. Una correcta integración es imprescindible para garantizar la coherencia y disponibilidad de los datos que serán posteriormente utilizados en procesos de fusión.

Fusión de sensores

La fusión de sensores consiste en la combinación de la información procedente de múltiples sensores con el objetivo de obtener una estimación más precisa, robusta y fiable del estado del sistema o del entorno.

Este proceso permite compensar las limitaciones individuales de cada sensor, reduciendo la incertidumbre asociada a las mediciones. En robótica móvil, la fusión sensorial se emplea habitualmente en tareas como la estimación de la posición y orientación, la detección de obstáculos o la interpretación del entorno [23].

Un ejemplo representativo es la combinación de datos de GNSS e IMU: mientras que el GNSS proporciona una referencia absoluta de posición con baja frecuencia, la IMU ofrece información de alta frecuencia sobre el movimiento, aunque con deriva acumulativa. La fusión de ambos permite obtener una estimación continua y precisa del estado del vehículo [5].

Niveles de fusión sensorial

La fusión sensorial puede clasificarse en función del nivel de procesamiento de la información [22]:

- Fusión a bajo nivel (data-level fusion): combina directamente datos en crudo provenientes de los sensores. Ofrece alta precisión, pero requiere una elevada capacidad computacional y una sincronización precisa.
- Fusión a nivel de características (feature-level fusion): utiliza información previamente procesada, como características extraídas (bordes, puntos clave, etc.), logrando un equilibrio entre complejidad y rendimiento.
- Fusión a alto nivel (decision-level fusion): combina decisiones o interpretaciones individuales de cada sensor, siendo común en sistemas basados en inteligencia artificial.

A modo de síntesis la Tabla 2-3 muestra los niveles de fusión junto a sus ventajas e inconvenientes.

Nivel de fusión	Tipo de información fusionada	Ventajas	Inconvenientes
Bajo nivel	Datos en crudo	Alta precisión	Alta complejidad computacional
Medio nivel	Características (features)	Equilibrio precisión-coste	Dependencia del procesado

Nivel de fusión	Tipo de información fusionada	Ventajas	Inconvenientes
Alto nivel	Decisiones	Baja complejidad	Menor precisión

Tabla 2-3 Comparativa entre los distintos niveles de fusión

Los distintos niveles de fusión no están asociados a un único algoritmo específico, sino que pueden implementarse mediante diferentes técnicas. Entre los métodos más utilizados destacan los enfoques probabilísticos, como el filtro de Kalman, ampliamente empleado en la estimación del estado en sistemas multisensor.

Métodos de fusión sensorial

La fusión de sensores se implementa mediante distintos algoritmos matemáticos y probabilísticos. Entre los más utilizados destacan:

- Filtro de Kalman: ampliamente empleado en la estimación del estado (posición, velocidad, orientación) mediante la combinación de sensores como GNSS e IMU [24].
- Filtro de partículas: adecuado para sistemas no lineales o con incertidumbre elevada.
- Filtros complementarios: utilizados en sistemas de bajo coste para combinar sensores con diferentes características dinámicas.

Estos métodos constituyen las principales herramientas utilizadas para implementar la fusión sensorial en sistemas reales, permitiendo materializar los distintos niveles de fusión descritos anteriormente.

Además de estos enfoques, existen otros métodos basados en diferentes paradigmas matemáticos y computacionales, como los enfoques probabilísticos, algebraicos, lógica difusa o aprendizaje automático [23].

En la Tabla 2-4 se presenta una clasificación de los principales métodos de fusión sensorial en función del enfoque matemático o computacional empleado, así como sus aplicaciones más habituales.

Tipo de método	Descripción	Ejemplos	Aplicación típica
Probabilísticos	Basados en modelos estadísticos para estimar el estado del sistema en presencia de ruido	Filtro de <i>Kalman</i> , Filtro de partículas	Estimación de posición (GNSS + IMU)
Algebraicos	Métodos deterministas basados en relaciones matemáticas	Mínimos cuadrados, interpolación	Ajuste y calibración de datos
Lógica difusa	Uso de reglas difusas para gestionar incertidumbre	Sistemas <i>fuzzy</i>	Control y toma de decisiones
Aprendizaje automático	Modelos que aprenden a partir de datos	Redes neuronales, <i>reinforcement learning</i>	Percepción avanzada
Métodos híbridos	Combinación de varios enfoques	<i>Kalman</i> + redes neuronales	Sistemas complejos de percepción

Tabla 2-4 Principales métodos de fusión sensorial

Ventajas de la integración y fusión sensorial en UGV

El uso de técnicas de fusión sensorial aporta múltiples ventajas en vehículos terrestres no tripulados:

- Mejora de la precisión en la estimación del estado
- Mayor robustez frente a fallos o pérdidas temporales de sensores
- Continuidad de la información
- Adaptabilidad a diferentes entornos y condiciones operativas

Estas características resultan especialmente relevantes en aplicaciones de supervisión remota y transmisión de datos en tiempo real.

Relación con el sistema propuesto

En el sistema desarrollado en este Trabajo Fin de Grado, la integración y fusión de sensores constituye un elemento clave para la adquisición y transmisión de información del UGV.

La combinación de sensores como GNSS, IMU, LiDAR y cámara RGB permite obtener una representación más completa y fiable del estado del vehículo y de su entorno. En particular, la fusión de GNSS e IMU resulta esencial para la estimación de la posición y orientación, mientras que el LiDAR y la cámara contribuyen a la percepción del entorno.

Este enfoque mejora la calidad y coherencia de los datos enviados a la estación base, permitiendo una monitorización más precisa del sistema.

2.5 Arquitecturas Sensoriales Avanzadas

Las arquitecturas sensoriales avanzadas describen la forma en que la información procedente de sensores heterogéneos es organizada, combinada y procesada dentro de un sistema no tripulado. A diferencia de la integración sensorial básica, estas arquitecturas abordan la fusión de datos en distintos niveles de abstracción, permitiendo adaptar el tratamiento de la información a los objetivos del sistema.

En vehículos terrestres no tripulados, la elección de una arquitectura sensorial adecuada resulta determinante para la eficiencia, robustez y escalabilidad del sistema de percepción. En función del nivel en el que se realice la combinación de la información, es posible distinguir tres enfoques principales: nivel físico, nivel de características y nivel semántico [21].

Arquitecturas de fusión a nivel físico

La fusión a nivel físico se realiza directamente sobre las señales sin procesar proporcionadas por los sensores, antes de que estas sean interpretadas. En este enfoque, la información se combina en su forma más básica, permitiendo aprovechar la redundancia entre sensores para mejorar la precisión y reducir el ruido de las mediciones.

Sin embargo, este tipo de arquitectura requiere una elevada sincronización temporal y una alta compatibilidad entre sensores, lo que limita su aplicación práctica. En el contexto de los UGV, su uso se restringe generalmente a sistemas específicos y controlados, debido a su complejidad y a los requisitos de procesamiento asociados.

Arquitecturas de fusión a nivel de características

La fusión a nivel de características combina información que ya ha sido parcialmente procesada, como posiciones, velocidades, distancias u otros parámetros extraídos de los datos brutos. Este enfoque permite integrar información procedente de sensores heterogéneos, manteniendo un equilibrio entre complejidad, flexibilidad y coste computacional.

En robótica móvil terrestre, este tipo de arquitectura es ampliamente utilizada para la estimación del estado del vehículo, por ejemplo, mediante la combinación de datos GNSS e IMU. Su principal ventaja

radica en la mejora de la robustez frente a fallos puntuales, lo que la convierte en uno de los enfoques más empleados en plataformas no tripuladas.

Arquitecturas de fusión a nivel semántico

La fusión a nivel semántico se realiza sobre información de alto nivel, previamente interpretada, como la identificación de objetos, eventos o estados del entorno. En este enfoque, los sensores no se combinan directamente a nivel de señal o parámetros, sino a nivel de significado, integrando resultados de procesos de percepción más avanzados.

Este tipo de arquitectura resulta especialmente útil en aplicaciones donde la toma de decisiones se basa en la interpretación del entorno, como la detección de obstáculos complejos o el reconocimiento de situaciones. No obstante, su implementación suele requerir una mayor capacidad de procesamiento y algoritmos avanzados, por lo que se emplea principalmente en sistemas con mayores recursos computacionales.

Relación con el sistema propuesto

El sistema desarrollado en este Trabajo Fin de Grado se basa principalmente en una arquitectura de fusión a nivel de características, combinando información procedente de distintos sensores para mejorar la fiabilidad y continuidad de los datos adquiridos. Este enfoque permite integrar sensores heterogéneos manteniendo una complejidad moderada, lo que resulta adecuado para una plataforma terrestre no tripulada de bajo coste.

El análisis de las arquitecturas sensoriales avanzadas proporciona el marco conceptual necesario para justificar las decisiones de diseño adoptadas y sitúa el sistema propuesto dentro del contexto actual de los sistemas de percepción en vehículos no tripulados.

Ejemplo conceptual de arquitectura sensorial avanzada

A modo de ejemplo conceptual, puede considerarse una arquitectura sensorial multimodal en la que un sistema no tripulado integra sensores heterogéneos, como un sensor LiDAR, un radar y cámaras, con el objetivo de obtener una percepción más completa y robusta del entorno. Cada uno de estos sensores proporciona información distinta y complementaria, basada en principios físicos diferentes, que es adquirida y procesada inicialmente de forma individual [21].

La información extraída de los distintos sensores se combina posteriormente mediante un proceso de fusión multimodal, permitiendo generar una representación coherente del entorno a partir de datos parciales y heterogéneos. Esta representación constituye la base para tareas de percepción avanzada, como la localización y el mapeado del entorno, así como para procesos de planificación y toma de decisiones del sistema.

Este enfoque se inspira en modelos de percepción empleados en sistemas inteligentes, donde cada sensor puede considerarse análogo a un “sentido” especializado, contribuyendo de forma conjunta a la interpretación global del entorno. De este modo, la arquitectura sensorial avanzada permite aprovechar las ventajas de cada tecnología, compensando sus limitaciones individuales y mejorando la robustez global del sistema. A modo de ejemplo, la Figura 2-4 muestra la arquitectura sensorial de nuestro vehículo UGV.

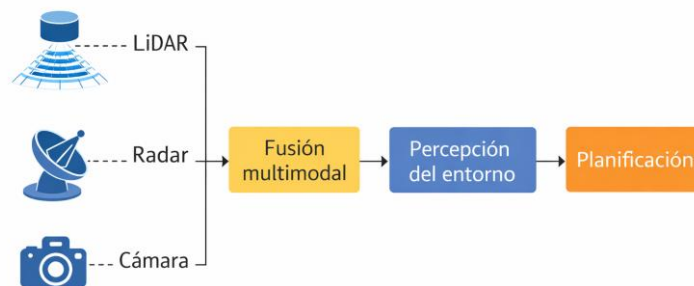


Figura 2-4 Ejemplo conceptual de una arquitectura sensorial multimodal basada en la fusión de información procedente de sensores heterogéneos para tareas de percepción y planificación en sistemas no tripulados. Fuente: Elaboración asistida por herramientas de inteligencia artificial

2.6 Aplicaciones reales de arquitecturas sensoriales en sistemas no tripulados

Las arquitecturas sensoriales avanzadas encuentran aplicación en una amplia variedad de sistemas no tripulados que operan en entornos muy diversos y presentan requisitos específicos en términos de percepción, navegación y toma de decisiones. En función del medio en el que operan, estos sistemas se clasifican habitualmente en vehículos aéreos, terrestres, de superficie y submarinos.

A pesar de las diferencias entre estos dominios, todos ellos comparten la necesidad de integrar sensores heterogéneos y fusionar la información adquirida para obtener una representación fiable del entorno y del estado del vehículo. A continuación, se describen aplicaciones reales representativas de estas arquitecturas en cada tipo de sistema no tripulado.

2.6.1 Vehículos Aéreos No Tripulados (UAV)

En los vehículos aéreos no tripulados, las arquitecturas sensoriales avanzadas desempeñan un papel fundamental debido a la dinámica del vuelo y a la necesidad de operar con precisión en entornos complejos. Estos sistemas integran habitualmente sensores como IMU, GNSS, cámaras y, en aplicaciones específicas, sensores LiDAR, cámaras infrarrojas o sensores hiperspectrales [25].

En aplicaciones de inspección de infraestructuras, la combinación de sensores LiDAR y cámaras infrarrojas permite detectar deformaciones estructurales y microfisuras en puentes, edificios o líneas eléctricas. En escenarios de búsqueda y rescate, las cámaras térmicas se emplean para la identificación de personas en entornos con visibilidad reducida o de difícil acceso. Asimismo, en el ámbito de la agricultura de precisión, el uso de cámaras hiperspectrales permite detectar el estrés de los cultivos y monitorizar su estado de salud, optimizando la gestión de recursos. A modo ilustrativo, la Figura 2-5 muestra la integración de distintos sensores en un vehículo aéreo no tripulado, así como sus principales aplicaciones en inspección de infraestructuras, búsqueda y rescate y agricultura de precisión.

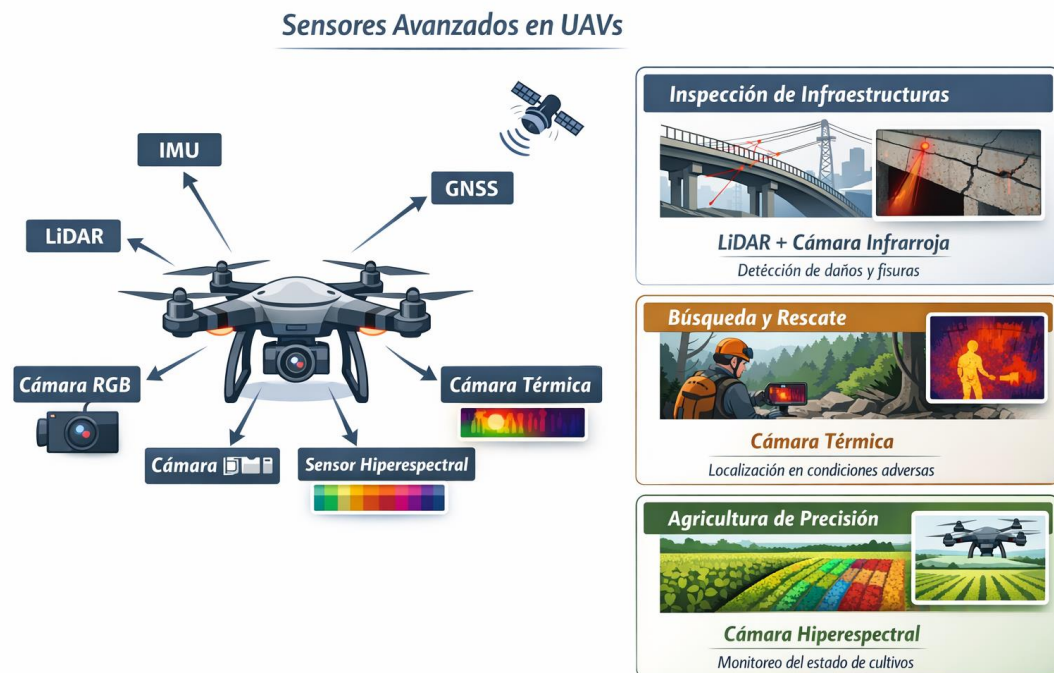


Figura 2-5 Arquitectura sensorial en vehículos aéreos no tripulados (UAV) y principales aplicaciones asociadas.
Fuente: Elaboración asistida por herramientas de inteligencia artificial

2.6.2 Vehículos Terrestres No Tripulados (UGV)

En los vehículos terrestres no tripulados, las arquitecturas sensoriales avanzadas se orientan principalmente a la percepción del entorno cercano y a la navegación segura en escenarios estructurados y no estructurados. Estos sistemas integran sensores como LiDAR, radar, cámaras y unidades de medida inercial, aprovechando la complementariedad entre información geométrica, visual y dinámica.

Un ejemplo representativo es la conducción autónoma en entornos urbanos, donde la combinación de sensores LiDAR de 360°, radar de alta resolución y sistemas de visión semántica permite detectar obstáculos, interpretar la escena y tomar decisiones de navegación en tiempo real. En aplicaciones industriales, como la minería subterránea, los UGV se emplean para la exploración y operación autónoma en entornos hostiles, donde la ausencia de señal GNSS y las condiciones ambientales adversas hacen imprescindible el uso de arquitecturas sensoriales robustas. A modo ilustrativo, la Figura 2-6 muestra la integración de sensores en un vehículo terrestre no tripulado, así como su aplicación en escenarios como la conducción autónoma en entornos urbanos y la operación en entornos hostiles como la minería subterránea [26], [27].



Figura 2-6 Arquitectura sensorial en vehículos terrestres no tripulados (UGV) y principales aplicaciones asociadas. Fuente: Elaboración asistida por herramientas de inteligencia artificial

2.6.3 Vehículos de Superficie No Tripulados (USV)

Los vehículos de superficie no tripulados operan en entornos marítimos caracterizados por condiciones dinámicas y por la presencia de otras embarcaciones. En estos sistemas, las arquitecturas sensoriales avanzadas integran sensores como radar, GNSS, cámaras electroópticas e infrarrojas y sistemas AIS, con el objetivo de garantizar una navegación segura y conforme a la normativa vigente.

En aplicaciones de vigilancia marítima, la fusión de información procedente de radar, AIS y sensores EO/IR permite detectar y seguir otras embarcaciones, facilitando el cumplimiento del reglamento internacional para prevenir abordajes en la mar (COLREG). Por otro lado, en tareas de monitorización oceanográfica, los USV integran sensores meteo-marinos que permiten adquirir datos ambientales de forma continua y autónoma. A modo ilustrativo, la Figura 2-7 muestra la arquitectura sensorial típica en vehículos de superficie no tripulados, así como sus principales aplicaciones en vigilancia marítima y monitorización oceanográfica [28], [29], [30].



Figura 2-7 Arquitectura sensorial en vehículos de superficie no tripulados (USV), incluyendo sensores principales y aplicaciones en vigilancia marítima y monitorización oceanográfica. Fuente: Elaboración asistida por herramientas de inteligencia artificial

2.6.4 Vehículos Submarinos No Tripulados (AUV)

En los vehículos submarinos no tripulados, la percepción del entorno presenta desafíos específicos debido a la imposibilidad de utilizar sistemas GNSS y a la limitada propagación de señales electromagnéticas bajo el agua. Por este motivo, las arquitecturas sensoriales avanzadas en AUV se basan en la integración de sensores acústicos y sistemas inerciales [28], [31].

En aplicaciones de inspección de infraestructuras submarinas, como oleoductos o cables, los AUV emplean sensores sonar multihaz para generar representaciones detalladas del entorno. Asimismo, en el ámbito de la defensa naval, estos vehículos se utilizan en tareas de reconocimiento y detección de minas, donde la fusión de información procedente de sonar e IMU resulta esencial para garantizar una navegación precisa. A modo ilustrativo, la Figura 2-8 muestra la arquitectura sensorial en vehículos submarinos no tripulados, así como sus principales aplicaciones en inspección de infraestructuras submarinas y detección de objetos en el ámbito de la defensa naval.

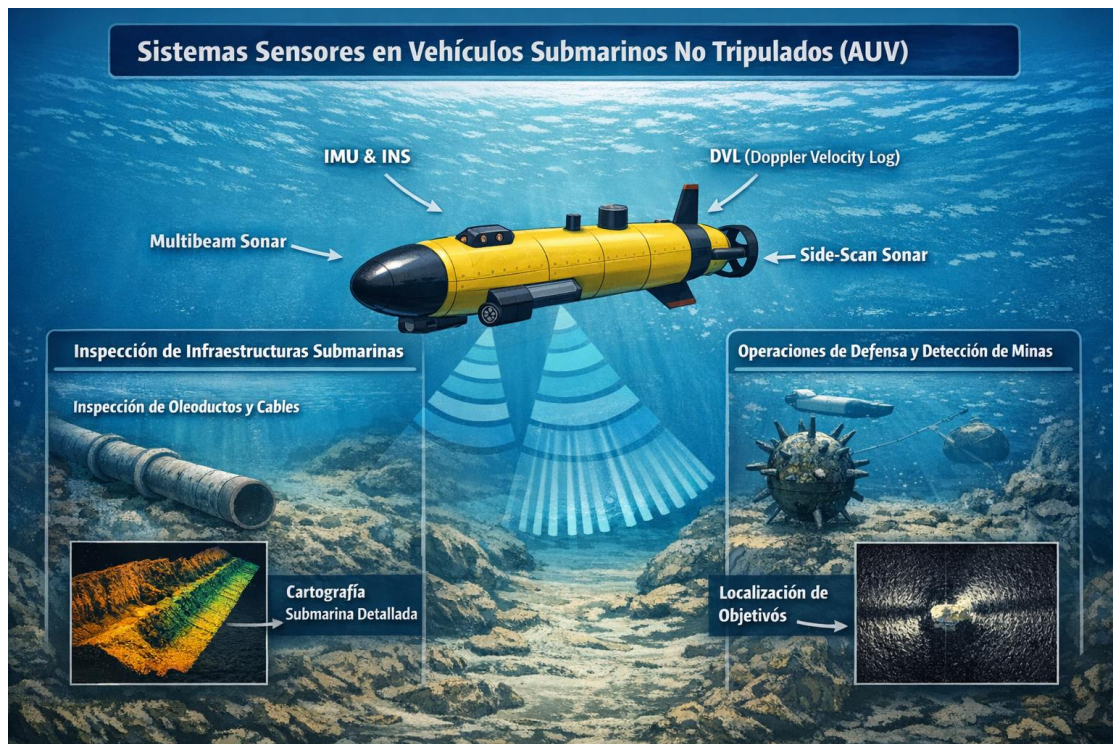


Figura 2-8 Arquitectura sensorial en vehículos submarinos no tripulados (AUV) y principales aplicaciones asociadas. Fuente: Elaboración asistida por herramientas de inteligencia artificial

La Figura 2-9 resume de forma comparativa los principales tipos de sistemas no tripulados y los sensores habitualmente empleados en cada uno de ellos, en función del entorno operativo.



Figura 2-9 Comparativa de los principales sistemas no tripulados (UAV, UGV, USV y AUV) y de los sensores habitualmente integrados en cada dominio operativo. Fuente: Elaboración asistida por herramientas de inteligencia artificial

2.7 Tecnologías emergentes en sensorización y percepción avanzada

El desarrollo reciente de sistemas no tripulados ha impulsado la aparición de tecnologías emergentes orientadas a mejorar la percepción del entorno, aumentar la autonomía y superar las limitaciones de los sensores tradicionales. Estas tendencias abarcan tanto la evolución del hardware en los sensores como el procesamiento distribuido en el borde (*edge computing*), con el objetivo de reducir latencias y optimizar el uso de recursos.

2.7.1 Tecnologías emergentes en sensores

Una línea destacada es el desarrollo de sensores bioinspirados, como las cámaras basadas en eventos (visión neuromórfica), que capturan cambios de intensidad en lugar de imágenes completas. Este enfoque permite una alta resolución temporal y baja latencia, siendo especialmente útil en entornos dinámicos, aunque requiere algoritmos específicos para su procesamiento [32].

En el ámbito de sensores de rango, el LiDAR FMCW (*Frequency-Modulated Continuous Wave*) representa una evolución respecto a los sistemas tradicionales, al permitir medir simultáneamente distancia y velocidad relativa, mejorando la robustez frente a interferencias [33].

Por su parte, los radares de imagen 4D amplían las capacidades del radar convencional al proporcionar información espacial más detallada, manteniendo su alta fiabilidad en condiciones ambientales adversas [34].

En entornos marítimos y submarinos (USV/AUV), el desarrollo de técnicas acústicas avanzadas, como el sonar holográfico, resulta clave para aumentar la calidad de la percepción y la seguridad operativa.

2.7.2 Tecnologías emergentes en procesamiento

Otra tendencia relevante es la integración de capacidades de procesamiento directamente en el sensor o en sistemas cercanos, dando lugar a sensores inteligentes con IA integrada. Este enfoque permite transformar datos brutos en información útil, reduciendo el volumen de datos transmitidos y mejorando la respuesta en tiempo real.

El procesamiento en *el borde* resulta especialmente útil en sistemas con restricciones de ancho de banda o que requieren operación autónoma, aunque introduce desafíos relacionados con la validación, la complejidad del sistema y la dependencia de los modelos utilizados [35].

2.7.3 Impacto potencial en sistemas no tripulados

Estas tecnologías permiten avanzar hacia sistemas de percepción más eficientes y robustos. En plataformas terrestres, mejoran la percepción en entornos complejos; en sistemas aéreos, contribuyen a reducir peso y consumo; y en entornos marítimos y submarinos, permiten mejorar la percepción en condiciones adversas.

En conjunto, estas tendencias refuerzan el desarrollo de arquitecturas multisensoriales avanzadas, en las que la combinación de sensores heterogéneos y el procesamiento inteligente de la información resultan clave para mejorar la autonomía y fiabilidad de los sistemas no tripulados [33], [34], [35].

A modo de síntesis, la Tabla 2-5 recoge las principales tecnologías emergentes descritas, destacando su aportación, retos asociados y su impacto potencial en los distintos dominios de operación.

Tecnología emergente	Qué aporta	Principales retos	Dominio de mayor impacto	Relevancia para el TFG
Sensores bioinspirados (visión neuromórfica)	Alta resolución temporal y baja latencia	Algoritmos específicos, integración compleja	UAV / UGV	Media
LiDAR FMCW	Medición simultánea de distancia y velocidad, mayor robustez	Complejidad técnica, coste	UGV / UAV	Alta

Tecnología emergente	Qué aporta	Principales retos	Dominio de mayor impacto	Relevancia para el TFG
Radar 4D imaging	Percepción espacial robusta en condiciones adversas	Resolución limitada, procesado complejo	UGV / USV	Media–Alta
Sonar holográfico	Imagen acústica de alta resolución	Coste y complejidad de procesamiento	AUV / USV	Media
Sensores con IA integrada (edge AI)	Reducción de latencia y volumen de datos	Validación y dependencia del modelo	Todos	Alta

Tabla 2-5 Comparativa de los principales sistemas no tripulados (UAV, UGV, USV y AUV) y de los sensores habitualmente integrados en cada dominio operativo

2.8 Modelos de Error, Ruido y Calibración

En los sistemas no tripulados, la información proporcionada por los sensores constituye la base para la percepción del entorno, la estimación del estado del vehículo y la toma de decisiones. Sin embargo, los sensores reales no ofrecen mediciones ideales, sino que están sujetos a errores, ruido e incertidumbre, derivados tanto de sus limitaciones físicas como de las condiciones del entorno operativo. Por este motivo, el análisis de los modelos de error y de los procesos de calibración resulta fundamental para comprender las limitaciones de los sistemas de percepción y justificar el uso de técnicas de fusión sensorial.

2.8.1 Errores y ruido en sensores

Los errores en sensores pueden clasificarse de forma general en errores sistemáticos y errores aleatorios. Los errores sistemáticos presentan un comportamiento repetible y predecible, y suelen estar asociados a imperfecciones de fabricación, desajustes o configuraciones incorrectas del sensor. Por el contrario, los errores aleatorios se manifiestan como fluctuaciones impredecibles en las mediciones, generalmente modeladas como ruido, y dependen tanto del sensor como del entorno [9], [15], [16].

En el caso de las unidades de medida inercial (IMU), uno de los fenómenos más relevantes es la deriva, que provoca que pequeños errores en la medición de aceleraciones o velocidades angulares se acumulen con el tiempo. Este efecto puede dar lugar a errores crecientes en la estimación de la posición y la orientación, llegando a comportamientos de deriva de carácter cuadrático si no se aplican mecanismos de corrección.

En los sensores radar, los errores están relacionados principalmente con el ruido Doppler y con las interferencias, especialmente en entornos donde operan múltiples vehículos o fuentes emisoras. Estas interferencias pueden afectar a la fiabilidad de la estimación de velocidad y a la correcta detección de objetivos, introduciendo incertidumbre en la percepción del entorno [20].

Las cámaras presentan errores asociados a la distorsión óptica de las lentes, que provoca desviaciones geométricas entre la escena real y su representación en la imagen capturada. Estas distorsiones, si no se corrigen adecuadamente, pueden afectar a la precisión de la detección de características y a la estimación de la geometría del entorno.

En sensores acústicos como el sonar, los principales factores de error incluyen la reverberación y el ruido ambiental, que pueden generar ecos no deseados y dificultar la correcta interpretación de las señales reflejadas. Estos efectos resultan especialmente relevantes en entornos complejos o con múltiples superficies reflectantes.

2.8.2 Modelo conceptual de error en las mediciones

De forma general, el comportamiento de un sensor puede representarse mediante un modelo conceptual de error, en el que la medición obtenida se expresa como la suma del valor real y distintos términos de error. De manera simplificada, este modelo puede expresarse como Ecuación 2-1:

$$x_{medido} = x_{real} + e_{sistemático} + e_{aleatorio} \quad \text{Ecuación 2-1}$$

Donde x_{real} representa el valor físico real de la magnitud medida, $e_{sistemático}$ engloba los errores deterministas o reproducibles asociados al sensor, y $e_{aleatorio}$ recoge las componentes de ruido impredecibles [9], [36].

Este modelo permite ilustrar cómo las mediciones se ven afectadas por distintos tipos de error y pone de manifiesto que, incluso en sensores correctamente configurados, siempre existe un grado de incertidumbre inherente a la medición. Además, en sistemas dinámicos como los vehículos no tripulados, estos errores pueden acumularse a lo largo del tiempo, degradando progresivamente la calidad de la información disponible.

2.8.3 Calibración de sensores

La calibración constituye el proceso mediante el cual se identifican y corrigen los errores sistemáticos de los sensores, con el objetivo de mejorar la precisión y coherencia de las mediciones. En sistemas no tripulados, la calibración resulta especialmente relevante cuando se integran sensores heterogéneos, ya que pequeñas discrepancias pueden dar lugar a inconsistencias significativas en la información fusionada.

De forma general, pueden distinguirse distintos tipos de calibración. La calibración intrínseca se refiere a los parámetros internos de cada sensor, como las características ópticas de una cámara o los sesgos de una IMU. La calibración extrínseca se centra en la relación espacial entre distintos sensores, siendo esencial para garantizar que la información procedente de múltiples dispositivos se combine de forma coherente. Asimismo, la calibración temporal resulta crítica en sistemas donde los sensores operan a distintas frecuencias o presentan desfases temporales.

Una calibración adecuada no elimina por completo el ruido aleatorio, pero reduce de forma significativa los errores sistemáticos, facilitando el uso de técnicas de fusión sensorial y mejorando la robustez del sistema frente a condiciones adversas.

2.8.4 Impacto del error y relación con la fusión sensorial

La presencia de errores y ruido en los sensores tiene un impacto directo en la percepción del entorno, la localización del vehículo y la toma de decisiones. En sistemas no tripulados, la acumulación de errores puede conducir a estimaciones incorrectas del estado del sistema, comprometiendo la seguridad y el rendimiento de la plataforma.

Por este motivo, la fusión sensorial se presenta como una estrategia fundamental para mitigar los efectos del error, combinando información procedente de sensores con características complementarias. La correcta comprensión de los modelos de error y de los procesos de calibración constituye, por tanto, un requisito previo para el diseño de arquitecturas de fusión avanzadas, que se abordarán en los apartados siguientes.

2.9 SLAM específico por sensor

La localización y mapeado simultáneos (SLAM, *Simultaneous Localization and Mapping*) constituyen un componente clave en los sistemas no tripulados que operan en entornos desconocidos o parcialmente estructurados. Su objetivo es estimar de forma conjunta la trayectoria del vehículo y la

representación del entorno a partir de información sensorial, enfrentándose a problemas de incertidumbre, ruido y acumulación de errores [37].

El enfoque adoptado para resolver el problema de SLAM depende en gran medida del tipo de sensor empleado, ya que cada modalidad proporciona información distinta y presenta ventajas y limitaciones específicas. A modo ilustrativo, la Figura 2-10 representa el proceso general del algoritmo SLAM, desde la adquisición de datos sensoriales hasta la estimación conjunta de la trayectoria del vehículo y la generación del mapa del entorno.

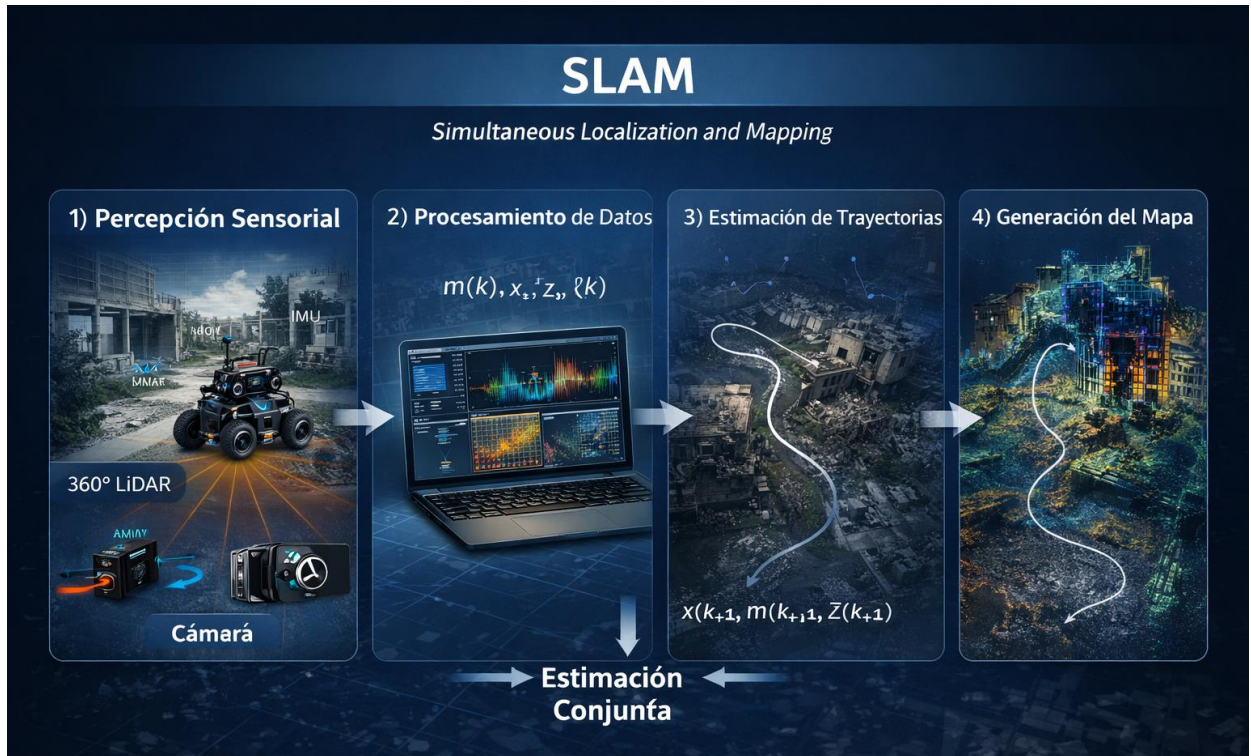


Figura 2-10 Proceso de localización y mapeado simultáneos (SLAM), incluyendo las etapas de percepción sensorial, procesamiento de datos, estimación de trayectoria y generación del mapa del entorno. Fuente: Elaboración asistida por herramientas de inteligencia artificial

2.9.1 SLAM basado en visión (Visual SLAM)

El SLAM visual utiliza cámaras para extraer información del entorno a partir de imágenes. Su principal ventaja es el bajo coste y la elevada resolución espacial, lo que permite detectar características del entorno y estimar el movimiento del vehículo.

Sin embargo, su rendimiento depende en gran medida de las condiciones de iluminación y de la presencia de textura en la escena, pudiendo degradarse en entornos poco estructurados o con baja visibilidad [37].

2.9.2 SLAM basado en LiDAR (LiDAR SLAM)

El SLAM basado en LiDAR emplea sensores de rango para obtener información geométrica precisa del entorno, permitiendo construir mapas tridimensionales y estimar la trayectoria del vehículo con alta precisión.

Este enfoque destaca por su robustez frente a condiciones de iluminación adversas, aunque presenta limitaciones en términos de coste, consumo energético y procesamiento de grandes volúmenes de datos [7], [38].

2.9.3 SLAM basado en radar (Radar SLAM)

El SLAM basado en radar ofrece una alta robustez frente a condiciones ambientales adversas como niebla, lluvia o polvo, lo que lo hace especialmente adecuado en escenarios donde otros sensores pueden fallar.

No obstante, su menor resolución espacial y la ambigüedad en las mediciones dificultan la construcción de mapas detallados, por lo que suele emplearse como complemento a otros sensores [34].

2.9.4 SLAM basado en sonar (Sonar SLAM)

El SLAM basado en sonar se utiliza principalmente en entornos submarinos, donde los sensores ópticos y electromagnéticos presentan limitaciones. Este enfoque permite realizar navegación y mapeado mediante señales acústicas.

Sin embargo, factores como la reverberación y el ruido acústico dificultan la interpretación de los datos, limitando la precisión de los mapas generados [10], [19].

2.9.5 Comparación y complementariedad entre enfoques

El análisis de los distintos enfoques de SLAM pone de manifiesto que no existe una solución universal, sino que la elección depende del entorno operativo y de los requisitos de la aplicación. Mientras que los enfoques visuales y LiDAR destacan por su capacidad de reconstrucción geométrica, los basados en radar y sonar ofrecen una mayor robustez en condiciones adversas.

Por este motivo, en sistemas no tripulados avanzados resulta habitual combinar distintos sensores mediante técnicas de fusión sensorial, con el objetivo de aprovechar las ventajas de cada modalidad y compensar sus limitaciones [7], [34], [37].

En este contexto, el uso combinado de sensores como LiDAR, IMU y GNSS se alinea con enfoques multisensor orientados a mejorar la robustez de la estimación de estado y la percepción del entorno.

A modo de síntesis, la Tabla 2-6 compara los principales enfoques de SLAM en función del tipo de sensor empleado, destacando la información aportada, sus ventajas y limitaciones, así como los dominios de aplicación más habituales.

Tipos de SLAM	Sensor principal	Información aportada	Ventajas principales	Limitaciones	Dominio de aplicación típico
Visual SLAM	Cámaras RGB / monoculares o estéreo	Información visual y características del entorno	Bajo coste, alta resolución espacial	Dependencia de textura e iluminación, ambigüedad de escala	UAV / UGV
LiDAR SLAM	LiDAR 2D / 3D	Geometría precisa del entorno	Alta precisión y robustez geométrica	Coste, consumo y volumen de datos	UGV / UAV
Radar SLAM	Radar	Distancia y velocidad, alta robustez ambiental	Operación en niebla, lluvia o polvo	Baja resolución espacial, ambigüedad	UGV / USV
Sonar SLAM	Sonar	Información acústica del entorno	Funcionamiento en entornos submarinos	Ruido acústico, reverberación	AUV

Tabla 2-6 Comparativa de los principales enfoques de SLAM en sistemas no tripulados según el tipo de sensor empleado

2.10 Inteligencia artificial en percepción cognitiva

El avance de la inteligencia artificial ha transformado de forma significativa los sistemas de percepción en vehículos no tripulados, no mediante la introducción de nuevos sensores, sino a través de la mejora en la capacidad de procesamiento e interpretación de la información sensorial. En este contexto, la IA actúa como una capa superior del sistema de percepción, permitiendo extraer conocimiento a partir de datos procedentes de múltiples fuentes.

La incorporación de técnicas de inteligencia artificial en sistemas de percepción abarca diferentes niveles, desde el procesamiento directo de los datos sensoriales hasta la fusión de información multimodal y la interpretación del entorno. Este enfoque permite avanzar hacia sistemas más autónomos, robustos y adaptativos, capaces de operar en entornos complejos y no estructurados.

2.10.1 IA en sensores y procesamiento de datos

Una de las principales aplicaciones de la inteligencia artificial en sistemas de percepción es su integración en etapas tempranas del procesamiento de datos. En este contexto, surgen los denominados sensores inteligentes, capaces de realizar tareas de preprocesado directamente en el propio dispositivo o en sistemas cercanos (*edge computing*).

Este enfoque permite transformar datos brutos en información más útil, como detecciones, estimaciones o características relevantes, reduciendo el volumen de datos a transmitir y mejorando la respuesta en tiempo real. La IA aplicada en esta etapa contribuye a optimizar el uso de recursos computacionales y energéticos, especialmente en plataformas con restricciones de tamaño, peso y consumo [35].

2.10.2 IA en fusión sensorial

La inteligencia artificial ha impulsado el desarrollo de técnicas avanzadas de fusión sensorial, capaces de integrar información procedente de sensores heterogéneos de forma más eficiente que los

enfoques clásicos. A través de modelos de aprendizaje automático, es posible aprender representaciones conjuntas que capturan relaciones complejas y no lineales entre distintas modalidades sensoriales.

Entre los enfoques más relevantes destacan las arquitecturas neuronales multimodales, diseñadas para procesar simultáneamente datos visuales, geométricos, inerciales o acústicos. Asimismo, la combinación de redes convolucionales (CNN) con modelos basados en transformadores permite integrar información espacial y contextual de forma eficaz. Por otro lado, las *Graph Neural Networks* (GNN) ofrecen un marco flexible para modelar relaciones estructuradas entre sensores, características u objetos [4], [35].

Estos enfoques permiten mejorar la robustez del sistema frente a ruido e incertidumbre, así como aumentar la capacidad de generalización en entornos complejos.

2.10.3 IA en percepción cognitiva

Más allá del procesamiento y la fusión de datos, la inteligencia artificial permite avanzar hacia el concepto de percepción cognitiva, entendido como la capacidad del sistema para interpretar el entorno y generar conocimiento de alto nivel a partir de la información sensorial.

En este contexto, la IA permite identificar objetos, clasificar escenas, detectar eventos relevantes y establecer relaciones entre distintos elementos del entorno. Este nivel de interpretación facilita la toma de decisiones y mejora la autonomía del sistema, permitiendo operar con menor supervisión humana.

La percepción cognitiva constituye un paso clave en la evolución de los sistemas no tripulados, al permitir pasar de una descripción puramente geométrica del entorno a una comprensión más completa y contextualizada [1], [4], [21].

2.10.4 Beneficios y limitaciones

La integración de inteligencia artificial en sistemas de percepción aporta beneficios significativos, como la mejora de la robustez frente a condiciones adversas, la capacidad de operar en entornos no estructurados y el aumento del nivel de autonomía del sistema.

No obstante, estos enfoques también presentan desafíos relevantes. Entre ellos destacan la dependencia de datos de entrenamiento representativos, la dificultad de garantizar un comportamiento fiable en todos los escenarios y la limitada interpretabilidad de algunos modelos. Además, el uso de técnicas avanzadas de IA implica un mayor coste computacional, lo que puede limitar su aplicación en sistemas embarcados con recursos restringidos.

2.10.5 Perspectiva futura

Desde una perspectiva futura, la integración progresiva de inteligencia artificial en los sistemas de percepción apunta hacia plataformas más autónomas, capaces de adaptarse a entornos cambiantes y de tomar decisiones basadas en una comprensión más profunda del contexto operativo.

Esta evolución se orienta hacia sistemas que no solo reaccionen a estímulos sensoriales, sino que sean capaces de anticipar situaciones, priorizar información relevante y colaborar de forma más eficiente con otros sistemas y operadores humanos. En este sentido, la inteligencia artificial se perfila como un elemento clave en el desarrollo de sistemas no tripulados de nueva generación.

2.11 Aplicaciones en la Armada y ejemplos reales

El empleo de sensores avanzados constituye una capacidad transversal en la Armada, presente tanto en plataformas tripuladas como no tripuladas, y en dominios marítimos, aéreos y terrestres. Los sistemas de percepción no se limitan a vehículos no tripulados, sino que forman parte integral de buques, aeronaves, vehículos terrestres, sistemas de armas y plataformas de apoyo, proporcionando información crítica para la vigilancia, la navegación, la identificación de amenazas y la toma de decisiones [28], [39].

En este contexto, los sistemas no tripulados deben entenderse como elementos complementarios que amplían las capacidades de percepción existentes, permitiendo operar en entornos de mayor riesgo, aumentar la cobertura sensorial y mejorar la persistencia de la vigilancia, sin sustituir a los sistemas tradicionales.

A modo ilustrativo, la Figura 2-11 representa el uso operativo de sistemas no tripulados como elemento complementario a las capacidades de percepción en entornos complejos.



Figura 2-11 Empleo de un vehículo aéreo no tripulado (UAV) en un entorno operativo para tareas de reconocimiento, vigilancia y adquisición de información sensorial. Fuente: [28]

2.11.1 Aplicaciones en el dominio marítimo

En el dominio marítimo, la Armada emplea una amplia variedad de sensores embarcados en buques de superficie, como radares, sistemas electroópticos e infrarrojos (EO/IR), sonares y sistemas de identificación automática (AIS), orientados a tareas de vigilancia, control del tráfico marítimo y protección de fuerzas. Estos sensores permiten generar una conciencia situacional completa del entorno marítimo, incluso en condiciones adversas.

Los vehículos de superficie no tripulados (USV) y los vehículos submarinos no tripulados (AUV) complementan estas capacidades al permitir operaciones de vigilancia persistente, inspección de infraestructuras portuarias y detección de amenazas submarinas, como minas. En particular, los AUV equipados con sensores acústicos avanzados desempeñan un papel relevante en misiones de guerra de minas y exploración submarina, donde la exposición directa de plataformas tripuladas puede resultar arriesgada [30], [31], [39], [40].

2.11.2 Aplicaciones en el dominio aéreo

En el ámbito aéreo, la aviación naval emplea sensores avanzados integrados en aeronaves tripuladas para misiones de reconocimiento, vigilancia marítima y apoyo a operaciones navales. Estos sistemas incluyen radares, sensores EO/IR y otros dispositivos de percepción que permiten la detección y seguimiento de objetivos a gran distancia.

Los vehículos aéreos no tripulados (UAV) amplían estas capacidades al proporcionar una mayor persistencia en la vigilancia y la posibilidad de operar en escenarios de riesgo elevado. Equipados con sensores ópticos, infrarrojos o de otro tipo, los UAV permiten realizar misiones de reconocimiento, evaluación de daños y apoyo a la toma de decisiones, integrándose con los sistemas tripulados en arquitecturas de percepción conjuntas [31].

2.11.3 Aplicaciones en el dominio terrestre

En el dominio terrestre, la Infantería de Marina emplea sistemas de percepción basados en sensores ópticos, infrarrojos y otros dispositivos tácticos para el reconocimiento, la vigilancia y la protección de fuerzas en entornos complejos. Estos sensores pueden integrarse tanto en plataformas tripuladas como en sistemas fijos o portátiles.

Los vehículos terrestres no tripulados (UGV) complementan estas capacidades al permitir la exploración de zonas peligrosas, la detección de amenazas y el apoyo a operaciones terrestres sin exponer directamente al personal. La integración de sensores en UGV facilita la obtención de información del entorno cercano y refuerza la seguridad en escenarios de alta incertidumbre [26], [27].

2.11.4 Integración de sistemas tripulados y no tripulados

Una tendencia clave en el estado del arte es la integración de sistemas tripulados y no tripulados dentro de arquitecturas de percepción y mando más amplias. En estas arquitecturas, la información sensorial procedente de distintas plataformas se comparte y fusiona para construir una representación coherente del entorno operativo, mejorando la conciencia situacional y la capacidad de respuesta.

Esta cooperación hombre-máquina permite aprovechar las fortalezas de cada tipo de plataforma: la capacidad de decisión y supervisión humana, junto con la flexibilidad y persistencia de los sistemas no tripulados. Desde una perspectiva futura, esta integración refuerza el papel de la percepción avanzada como elemento central en las operaciones navales modernas [28], [31], [39].

La Figura 2-12 muestra de forma conceptual la integración de sensores avanzados en plataformas tripuladas y no tripuladas de la Armada, distribuidas en los dominios marítimo, aéreo y terrestre, y su contribución conjunta a la percepción del entorno operativo.



Figura 2-12 Integración conceptual de plataformas tripuladas y no tripuladas de la Armada en los dominios marítimo, aéreo y terrestre mediante el uso transversal de sensores avanzados y la compartición de información sensorial. Fuente: Elaboración asistida por herramientas de inteligencia artificial

2.12 Normativa y Seguridad en Sensores

El diseño y despliegue de sistemas de percepción en plataformas tripuladas y no tripuladas debe realizarse dentro de un marco normativo sólido, especialmente cuando estos sistemas intervienen en funciones críticas como la navegación, la vigilancia o la toma de decisiones. En este contexto, la seguridad no se limita al funcionamiento individual de los sensores, sino que abarca la arquitectura

completa del sistema, incluyendo la fusión de datos, la gestión de fallos y la protección frente a amenazas externas [18].

Existen diversos estándares que establecen principios de seguridad funcional, fiabilidad y validación en sistemas complejos. Entre los más relevantes destacan:

- IEC 61508, como referencia general en seguridad funcional para sistemas eléctricos, electrónicos y programables.
- ISO 26262, orientada a sistemas de movilidad, que introduce metodologías estructuradas de análisis de riesgos y gestión de fallos.
- EN 13849, centrada en la fiabilidad y validación de funciones de seguridad.
- DO-178C, aplicada al desarrollo de software crítico en el ámbito aeronáutico, con énfasis en la verificación y trazabilidad.

Estas normas no deben entenderse de forma aislada, sino como un conjunto coherente de principios orientados a garantizar la seguridad y fiabilidad de sistemas en los que la percepción desempeña un papel crítico [41], [42], [43], [44], [45], [46].

La aplicación de estos estándares pone de manifiesto la necesidad de diseñar arquitecturas sensoriales capaces de detectar, aislar y gestionar fallos. En este contexto, la redundancia sensorial y la validación cruzada de la información constituyen estrategias fundamentales para aumentar la fiabilidad del sistema.

La capacidad de operar en modos degradados y la supervisión continua del estado de los sensores permiten reducir el impacto de fallos parciales, mejorando la robustez global del sistema.

Además de los fallos físicos, los sistemas de percepción deben afrontar riesgos asociados a la ciberseguridad. La manipulación o alteración de los datos sensoriales puede comprometer la percepción del entorno y generar decisiones incorrectas.

Por ello, resulta esencial garantizar la integridad, autenticidad y disponibilidad de la información, integrando mecanismos de protección frente a ataques o interferencias externas [44].

En sistemas de percepción avanzados basados en inteligencia artificial, normas como ISO 21448 (*SOTIF*) abordan escenarios en los que el sistema puede comportarse de forma insegura sin que exista un fallo técnico, debido a limitaciones en la interpretación del entorno.

Este enfoque pone de relieve la necesidad de evaluar no solo los fallos del sistema, sino también sus limitaciones inherentes, especialmente en entornos complejos o no previstos [42], [44].

La normativa y los principios de seguridad deben considerarse desde las primeras etapas del diseño, y no como elementos añadidos posteriormente. Este enfoque permite desarrollar sistemas de percepción coherentes, verificables y alineados con los requisitos operativos.

En sistemas autónomos y semiautónomos, donde la percepción influye directamente en la interacción con el entorno, la integración de estos principios resulta esencial para garantizar un funcionamiento seguro y fiable [42], [43], [45].

La Tabla 2-7 recoge las normativas tratadas con su respectivo, ámbito y aplicación a modo resumen.

Norma	Ámbito	Aplicación
IEC 61508	General	Seguridad funcional
ISO 26262	Automoción	Vehículos

Norma	Ámbito	Aplicación
DO-178C	Aeronáutico	Software crítico
ISO 21448	IA	SOTIF

Tabla 2-7 Normativas en seguridad y sensores

2.13 Conclusiones Generales

El análisis realizado en este capítulo pone de manifiesto la importancia de los sistemas de percepción en el desarrollo de plataformas no tripuladas. Los sensores constituyen la base para la adquisición de información del entorno y del estado del vehículo, siendo necesarios para tareas como la localización, el mapeado y la toma de decisiones.

La revisión de las principales tecnologías sensoriales ha mostrado que no existe una solución única válida para todos los entornos y aplicaciones, sino que cada sensor presenta ventajas y limitaciones específicas. Esta circunstancia justifica el uso de arquitecturas multisensoriales y de técnicas de integración y fusión, orientadas a mejorar la robustez y fiabilidad del sistema.

Asimismo, el estudio de los modelos de error, ruido y calibración ha evidenciado que la incertidumbre es inherente a cualquier sistema sensorial real, por lo que su correcta gestión resulta esencial en el diseño de sistemas de percepción fiables. Del mismo modo, la incorporación de inteligencia artificial y el desarrollo de nuevas arquitecturas de percepción permiten avanzar hacia sistemas más autónomos, adaptativos y capaces de interpretar el entorno con mayor nivel de complejidad.

La revisión de aplicaciones en el ámbito de la Armada y del marco normativo asociado ha puesto de relieve que estas tecnologías no solo deben ser eficaces desde el punto de vista técnico, sino también seguras, fiables y coherentes con los requisitos operativos de sistemas críticos [28], [44], [45].

En conjunto, este estado del arte proporciona la base conceptual necesaria para justificar las decisiones de diseño adoptadas en el desarrollo del sistema multisensor propuesto en este Trabajo Fin de Grado.

3 DESARROLLO DEL TFG

3.1 Plataformas y Sensores del Sistema

En este apartado se describen los principales elementos físicos que conforman el sistema desarrollado, incluyendo la plataforma móvil, la unidad de procesamiento y los sensores embarcados. Además de presentar sus características técnicas más relevantes, se justifica su selección en función de criterios de integración, coste, compatibilidad y adecuación a los objetivos del proyecto.

3.1.1 Plataforma Móvil

Descripción General del Vehículo

La plataforma móvil constituye el elemento base sobre el que se integran los distintos sensores y sistemas de procesamiento desarrollados en este trabajo. Para ello, se ha seleccionado un vehículo radiocontrol tipo *crawler*, concretamente el *Land Rover Defender TRX-4* (Model 822256-4). En la Figura 3-1 se muestra el vehículo base empleado como punto de partida para la integración del sistema sensorial [47].



Figura 3-1 Plataforma móvil empleada en el proyecto, basada en un vehículo teledirigido Land Rover Defender TRX4, que sirve como base para la integración de los distintos sensores y sistemas de procesamiento. Fuente: [47]

La elección de esta plataforma se ha realizado atendiendo a varios criterios fundamentales. En primer lugar, su elevada capacidad de tracción y suspensión permite operar en entornos no estructurados, lo que resulta adecuado para escenarios reales donde el terreno puede presentar irregularidades. En segundo lugar, su robustez mecánica facilita la integración de sensores y componentes electrónicos sin comprometer la estabilidad del sistema. Por último, su tamaño y capacidad de carga permiten alojar la unidad de procesamiento, los sensores y el sistema de alimentación de forma adecuada.

Desde el punto de vista estructural, el TRX-4 dispone de un chasis que permite la fijación de soportes adicionales, lo que facilita la instalación de sensores en posiciones estratégicas. Esto resulta

especialmente relevante en sistemas multisensor, donde la orientación y ubicación de cada dispositivo influye directamente en la calidad de los datos adquiridos.

Además, la configuración de este tipo de vehículo permite mantener una buena estabilidad durante el movimiento, lo que reduce el impacto de vibraciones en sensores sensibles como las unidades de medida inercial (IMU). Este aspecto resulta clave en el diseño del sistema, ya que la calidad de las mediciones depende en gran medida de las condiciones de montaje.

En conjunto, la plataforma seleccionada proporciona una base adecuada para el desarrollo de un sistema de percepción embarcado, permitiendo integrar de forma eficiente los distintos sensores y garantizando un funcionamiento estable en entornos reales.

Como se aprecia en la Figura 3-1, el vehículo presenta una configuración 4WD y una suspensión de largo recorrido característica de los modelos tipo *crawler*. Estas propiedades proporcionan una elevada capacidad todoterreno y estabilidad en terrenos irregulares, aspectos especialmente relevantes para la evaluación del sistema sensorial en entornos no completamente controlados.

Características Técnicas

Con el fin de definir con precisión las características físicas y dinámicas de la plataforma base empleada, en la Tabla 3-1 se resumen sus principales especificaciones técnicas.

Parámetro	Especificación aproximada
Modelo	<i>Traxxas TRX-4 Defender (822256-4)</i>
Tipo de vehículo	<i>Crawler</i> radiocontrol 4x4
Escala	1:10
Longitud total	~586 mm
Anchura	~249 mm
Altura	~291 mm
Distancia entre ejes	~324 mm
Configuración de tracción	4WD permanente
Tipo de motor	Motor eléctrico <i>brushed</i>
Sistema de transmisión	2 velocidades + diferenciales bloqueables
Peso aproximado (sin sensores)	~3,2 kg
Tipo de suspensión	Suspensión independiente con gran recorrido
Material principal del chasis	Estructura plástica reforzada

Tabla 3-1 Especificaciones técnicas del vehículo base *Traxxas TRX-4* empleado como plataforma móvil del UGV. Fuente: [47]

Las características propias de un vehículo tipo *crawler*, orientado a par motor y capacidad todoterreno, lo convierten en una plataforma adecuada para el desarrollo de un UGV experimental, ya que permite operar en entornos irregulares y superar obstáculos sin comprometer la estabilidad del conjunto sensorial. No obstante, su masa y sistema de suspensión obligan a considerar cuidadosamente la distribución de pesos y la mitigación de vibraciones en el diseño de la plataforma de acople.

Justificación de la Elección de la Plataforma

La elección del Land Rover Defender TRX-4 como plataforma móvil se basa principalmente en criterios de estabilidad, modularidad y versatilidad. A diferencia de otras plataformas más ligeras o de tipo diferencial, el TRX-4 ofrece una mayor capacidad de carga y una estructura mecánica preparada para soportar la integración simultánea de múltiples sensores.

Asimismo, su diseño permite realizar modificaciones reversibles, lo que resulta especialmente relevante en el contexto de un Trabajo Fin de Grado, donde se prioriza la reutilización del material y la posibilidad de ajustar el sistema a diferentes configuraciones experimentales. Por último, el uso de una plataforma ampliamente conocida y documentada facilita la reproducibilidad del proyecto y la comparación con otros trabajos similares.

3.1.2 Unidad de Procesamiento

Raspberry Pi 4

La unidad de procesamiento constituye el núcleo del sistema embarcado, siendo responsable de la adquisición, procesamiento y gestión de los datos procedentes de los distintos sensores. En este trabajo se ha seleccionado la Raspberry Pi 4 Model B como plataforma principal de procesamiento. Una plataforma de computación de placa reducida basada en arquitectura ARM, ampliamente utilizada en aplicaciones de robótica y sistemas embebidos. En la Figura 3-2 se muestra la placa utilizada en el desarrollo del proyecto. Este dispositivo actúa como el núcleo del sistema, centralizando la adquisición y el procesamiento de la información procedente de los distintos sensores integrados en la plataforma móvil.



Figura 3-2 Raspberry Pi 4 Model B empleada como unidad central de procesamiento para la adquisición, sincronización y gestión de los datos procedentes de los distintos sensores del sistema

La Raspberry Pi 4 incorpora un procesador *ARM Cortex-A72* de cuatro núcleos, con una frecuencia de funcionamiento suficiente para gestionar tareas de adquisición de datos sensoriales, procesamiento básico y comunicaciones en tiempo real. Su compatibilidad con sistemas operativos basados en Linux permite el uso de entornos de desarrollo estándar y facilita la integración de librerías y herramientas ampliamente utilizadas en robótica [48].

Como se observa en la Figura 3-2, la Raspberry Pi 4 dispone de múltiples interfaces de comunicación, tales como puertos USB, GPIO, I2C y UART, lo que permite la conexión simultánea de sensores heterogéneos como cámara, LiDAR, IMU y GNSS. Asimismo, su compatibilidad con sistemas operativos basados en Linux facilita la instalación de librerías y herramientas necesarias para el desarrollo del sistema sensorial.

Capacidades de Procesamiento y Conectividad

La Raspberry Pi 4 Model B incorpora un procesador *Broadcom BCM2711*, basado en una arquitectura ARM Cortex-A72 de cuatro núcleos a 64 bits, con una frecuencia de reloj de hasta 1.5 GHz, lo que proporciona capacidad suficiente para la ejecución simultánea de tareas de adquisición de sensores, procesamiento de datos y gestión del sistema operativo. Esta capacidad resulta especialmente relevante para la integración de sensores como el LiDAR, la cámara y las unidades inerciales, cuyos datos deben ser tratados de forma concurrente [48].

En cuanto a memoria, la plataforma dispone de memoria RAM LPDDR4 (en la configuración empleada en el proyecto), lo que permite el manejo eficiente de flujos de datos continuos y el funcionamiento estable de aplicaciones de robótica y visión artificial sin comprometer el rendimiento del sistema.

Desde el punto de vista de la conectividad, la Raspberry Pi 4 Model B ofrece un conjunto de interfaces que facilitan la integración de los distintos dispositivos del sistema. Dispone de dos puertos USB 3.0 y dos puertos USB 2.0, utilizados para la conexión directa del sensor LiDAR, la cámara USB, las unidades IMU y el receptor GNSS. Asimismo, incorpora un puerto Ethernet Gigabit, que permite la comunicación con un ordenador externo para tareas de monitorización, depuración y transferencia de datos.

Adicionalmente, la plataforma cuenta con interfaces de comunicación inalámbrica WiFi de doble banda y Bluetooth 5.0, lo que posibilita la conexión remota al sistema sin necesidad de enlaces físicos, facilitando el control y la supervisión del vehículo durante las pruebas experimentales [48].

En conjunto, estas capacidades de procesamiento y conectividad hacen de la Raspberry Pi 4 Model B una plataforma adecuada para actuar como núcleo del sistema, permitiendo la integración de múltiples sensores y la ejecución de software de control y adquisición de datos de forma fiable.

Aplicación de la Raspberry Pi 4 en el proyecto y justificación de la elección

Dentro del sistema global, la Raspberry Pi 4 desempeña el papel de unidad central de control y procesamiento, encargándose de coordinar la adquisición de datos provenientes de los diferentes sensores y de su publicación dentro de la arquitectura ROS 2.

La elección de esta plataforma se realizó inicialmente en base a criterios como su bajo consumo energético, tamaño reducido, coste contenido y compatibilidad con sistemas operativos basados en Linux, lo que permite el uso de herramientas ampliamente utilizadas en robótica, como ROS 2. Asimismo, su disponibilidad de múltiples interfaces de comunicación facilita la integración de sistemas multisensor de forma sencilla.

No obstante, durante la fase de validación experimental se ha comprobado que la Raspberry Pi 4 presenta limitaciones significativas en términos de capacidad de procesamiento, especialmente en tareas que implican un alto volumen de datos, como la transmisión de imágenes en tiempo real. En particular, se han observado retardos elevados en el flujo de vídeo, lo que pone de manifiesto que esta plataforma

resulta insuficiente para aplicaciones que requieren un procesamiento intensivo o una respuesta estrictamente en tiempo real.

En este sentido, aunque la Raspberry Pi constituye una solución adecuada para el desarrollo de prototipos y sistemas de bajo coste, los resultados obtenidos indican la necesidad de emplear hardware de mayor capacidad en aplicaciones más exigentes, especialmente aquellas relacionadas con percepción avanzada o procesamiento de imagen.

En la Figura 3-3 se muestra la ubicación de la Raspberry Pi dentro del montaje del UGV.

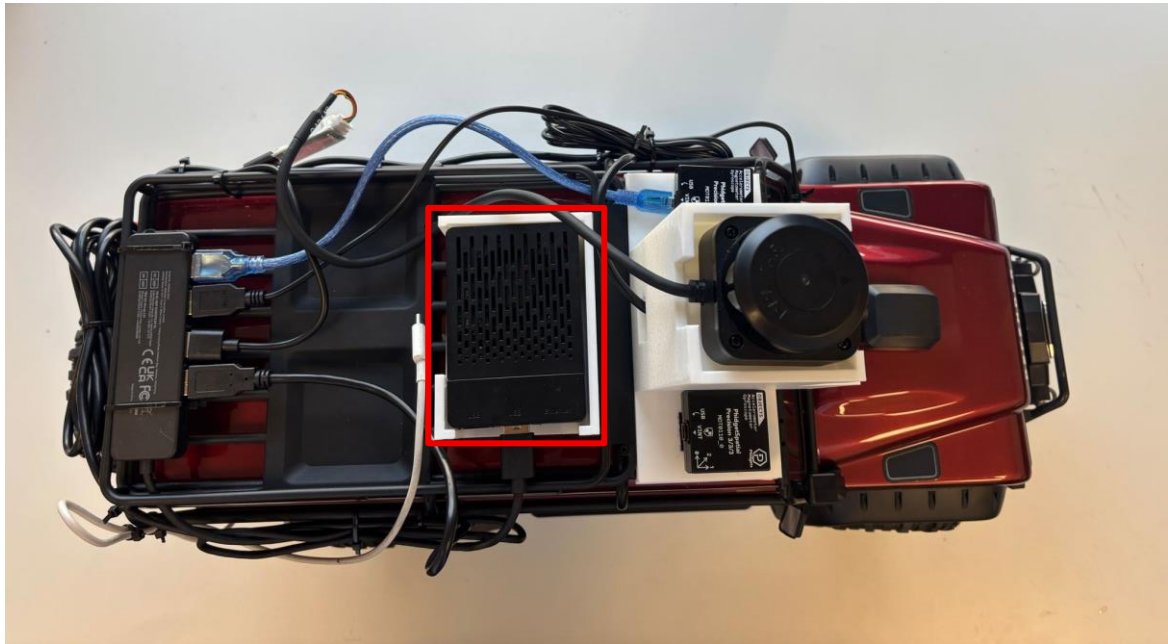


Figura 3-3 Ubicación de la Raspberry Pi dentro del montaje en el UGV

3.1.3 Sensor LiDAR

Principio de Funcionamiento de un LiDAR 2D

Un sensor LiDAR (*Light Detection and Ranging*) es un dispositivo de percepción que permite medir distancias mediante la emisión de pulsos de luz láser como se explica en el apartado Sensores LiDAR.

El sensor LiDAR constituye uno de los elementos principales del sistema de percepción, ya que permite obtener información geométrica del entorno mediante la medición de distancias en un plano. En este trabajo se ha empleado el modelo RPLIDAR C1, un LiDAR 2D de bajo coste orientado a aplicaciones de robótica móvil [21].

Este sensor proporciona medidas de distancia distribuidas en un campo de visión de 360° como se puede apreciar en la Figura 3-4, generando una representación del entorno en forma de nube de puntos bidimensional. Esta información resulta especialmente útil para tareas de detección de obstáculos y análisis del entorno cercano del vehículo.

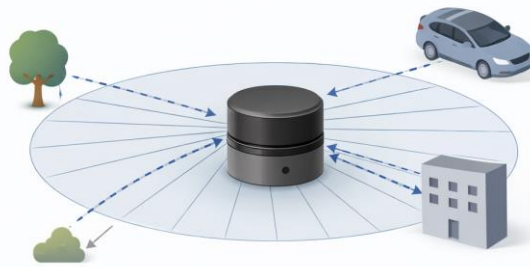


Figura 3-4 Principio de funcionamiento de un sensor LiDAR bidimensional, basado en la emisión de pulsos láser y la medición de distancias mediante un barrido plano de 360 grados alrededor del sensor. Fuente: Elaboración propia mediante generación asistida por inteligencia artificial

Como se observa en la Figura 3-4, el sensor realiza un barrido angular completo en un plano horizontal, generando una representación bidimensional del entorno en forma de nube de puntos.

Características del Sensor RPLiDAR

El RPLIDAR C1, Figura 3-5, es un sensor LiDAR bidimensional de escaneo rotativo diseñado para aplicaciones de robótica móvil y percepción del entorno. Este dispositivo es capaz de realizar un barrido completo de 360°, proporcionando medidas de distancia distribuidas angularmente que permiten generar representaciones del entorno en tiempo real.



Figura 3-5 Sensor LiDAR bidimensional de la familia RPLIDAR empleado para la adquisición de información geométrica del entorno mediante barrido plano. Fuente: [6]

El sensor presenta un alcance de hasta 12 metros y una frecuencia de escaneo configurable entre 5 y 15 Hz, lo que lo hace adecuado para aplicaciones donde se requiere una actualización continua de la información del entorno. Además, su elevada tasa de muestreo, que puede alcanzar hasta 8.000 muestras por segundo, permite obtener una resolución angular suficientemente detallada para tareas de detección de obstáculos y navegación [6].

Desde el punto de vista de integración, el RPLIDAR C1 destaca por su interfaz USB y su bajo consumo energético, lo que facilita su uso en plataformas embarcadas con recursos limitados. Estas características, junto con su tamaño compacto y su coste reducido, lo convierten en una solución adecuada para sistemas multisensor orientados a entornos de investigación, desarrollo y validación.

A modo resumen la Tabla 3-2 muestra las características del LiDAR empleado

Parámetro	Valor
Tipo de sensor	LiDAR 2D de escaneo rotativo
Campo de visión	360°
Alcance máximo	Hasta 12 m
Frecuencia de escaneo	5 – 15 Hz
Frecuencia de muestreo	Hasta 8.000 muestras/s
Precisión	±3 cm (aprox.)
Resolución angular	≤ 0,9°
Interfaz de comunicación	USB
Alimentación	5 V
Consumo	Bajo (apto para sistemas embebidos)
Dimensiones	Compactas
Aplicaciones típicas	Robótica móvil, SLAM, detección de obstáculos

Tabla 3-2 Resumen de las características del RPLIDAR C1. Fuente: [6]

Aplicación del LiDAR en el proyecto y justificación de su elección

En el contexto de este proyecto, el sensor LiDAR desempeña un papel fundamental en la percepción del entorno. La información proporcionada por el sensor permite obtener una representación bidimensional del espacio circundante al vehículo, que puede ser utilizada para la detección de obstáculos y el análisis de la geometría del entorno.

La colocación del LiDAR en una posición elevada sobre la plataforma móvil contribuye a maximizar su campo de visión y a reducir posibles interferencias producidas por elementos estructurales del vehículo. Tal como se muestra en la Figura 3-6. De este modo, el sensor proporciona datos consistentes y repetibles, adecuados para su posterior procesamiento y utilización en tareas de navegación y experimentación.

La elección de este modelo se ha realizado atendiendo a varios factores. En primer lugar, su compatibilidad con sistemas basados en ROS 2 facilita su integración dentro de la arquitectura del sistema. En segundo lugar, su bajo consumo energético y su reducido tamaño lo hacen adecuado para plataformas embarcadas con limitaciones de espacio y energía. Asimismo, su coste accesible permite su utilización en entornos de desarrollo y validación sin requerir equipos de alta gama.

Desde el punto de vista de la integración, el sensor se conecta a la unidad de procesamiento mediante interfaz USB, lo que simplifica su configuración y uso. Además, su diseño rotativo permite obtener información continua del entorno sin necesidad de mecanismos adicionales.

No obstante, el uso de un LiDAR 2D presenta ciertas limitaciones, como la ausencia de información en el eje vertical, lo que puede dificultar la detección de obstáculos elevados o estructuras complejas. Asimismo, su alcance y precisión son inferiores a los de sensores LiDAR de mayor gama, lo que condiciona su uso en escenarios de mayor escala.

A pesar de estas limitaciones, el RPLIDAR C1 proporciona información suficiente para la caracterización del entorno cercano y resulta adecuado para los objetivos de este trabajo, especialmente en el contexto de un sistema multisensor de bajo coste.



Figura 3-6 Esquema inicial de la ubicación del sensor LiDAR sobre un soporte cilíndrico elevado en la parte frontal del techo del rover. Fuente: Elaboración propia mediante generación asistida por inteligencia artificial

3.1.4 Cámara USB

Descripción de la Cámara Sipro High Speed USB

El sistema de visión del proyecto se basa en una cámara digital USB de 2.0 megapíxeles, diseñada para su uso en sistemas embebidos y aplicaciones de adquisición de imagen en tiempo real. La cámara se conecta directamente a la unidad de procesamiento mediante interfaz USB y se alimenta a 5 V, lo que permite su integración sin necesidad de circuitería adicional [49].

Parámetros Técnicos

El sistema de visión del vehículo se basa en una cámara USB de 2.0 megapíxeles, conectada directamente a la Raspberry Pi 4 Model B mediante interfaz USB. Se trata de una cámara compatible con el estándar UVC (USB Video Class), lo que permite su funcionamiento nativo en sistemas Linux sin necesidad de controladores específicos. La Figura 3-7 muestra la cámara utilizada para la captura de imágenes del entorno durante el desplazamiento del UGV.



Figura 3-7 Cámara USB de 2.0 megapíxeles empleada como sistema de visión para la captura de imágenes en color del entorno

Como se aprecia en la Figura 3-7, la cámara presenta un diseño compacto y conexión directa mediante interfaz USB, lo que facilita su integración con la Raspberry Pi 4 sin necesidad de hardware

adicional. Su resolución resulta suficiente para tareas de supervisión visual y monitorización remota del entorno, manteniendo un equilibrio adecuado entre calidad de imagen y consumo de recursos computacionales

La cámara proporciona imágenes en color, empleando el modelo de color RGB, lo que permite la captura simultánea de la información correspondiente a los canales rojo (R), verde (G) y azul (B). Esta característica resulta fundamental para aplicaciones de visión artificial en las que se requiere información cromática del entorno, como la identificación visual de escenas o elementos del entorno.

En cuanto a la resolución, el sensor dispone de una resolución máxima de 2.0 megapíxeles, permitiendo la adquisición de imágenes con resoluciones habituales de hasta 1920×1080 píxeles (Full HD), dependiendo de la configuración seleccionada. Esta resolución proporciona un nivel de detalle suficiente para tareas de percepción visual manteniendo un compromiso adecuado entre calidad de imagen y carga computacional [49].

La cámara funciona con una tensión de alimentación de 5 V, suministrada directamente a través del puerto USB, simplificando su integración eléctrica en el sistema. El uso de una única interfaz para alimentación y transmisión de datos reduce el cableado necesario y facilita el montaje sobre la plataforma móvil.

A modo de síntesis la Tabla 3-3 muestra un resumen de las características de la cámara usada dentro del proyecto.

Parámetro	Valor
Tipo de sensor	Cámara RGB
Interfaz	USB
Resolución	Hasta 1080p
Frecuencia de captura	Hasta 30 fps
Campo de visión	Variable (según lente)
Formato de imagen	RGB
Alimentación	A través de USB
Compatibilidad	Linux / ROS 2
Aplicaciones	Visión artificial, monitorización, percepción

Tabla 3-3 Resumen de las características generales de la Cámara Sypro High Speed USB. Fuente: [49]

Gracias a su conexión USB y a su compatibilidad con herramientas estándar de captura de vídeo en Linux, la cámara puede integrarse fácilmente en el sistema software del vehículo, permitiendo la adquisición continua de imágenes y su procesamiento en tiempo real o diferido según las necesidades del proyecto.

Aplicación de la Cámara en el proyecto y justificación en el proyecto

La elección de una cámara USB se ha realizado principalmente por su facilidad de integración con la Raspberry Pi, su compatibilidad con sistemas basados en Linux y su bajo coste. Además, este tipo de cámaras permite una rápida configuración y puesta en marcha, lo que resulta adecuado en entornos de desarrollo y validación.

Desde el punto de vista de la integración, la cámara se conecta directamente a la unidad de procesamiento mediante interfaz USB, permitiendo la captura y transmisión de imágenes en tiempo real.

Estas imágenes son posteriormente procesadas y enviadas a la estación base, donde pueden ser visualizadas mediante herramientas como Foxglove [50].

No obstante, este tipo de sensor presenta ciertas limitaciones, como su dependencia de las condiciones de iluminación y su menor capacidad de percepción en entornos con baja visibilidad. Asimismo, el procesamiento de imagen puede suponer una carga computacional significativa para la unidad de procesamiento, lo que condiciona la resolución y frecuencia de captura empleadas.

A pesar de estas limitaciones, la cámara RGB proporciona información visual de gran valor para el sistema, permitiendo enriquecer la percepción del entorno y complementando el resto de los sensores integrados.

A modo ilustrativo, la Figura 3-8 muestra la imagen capturada por la cámara del sistema durante su operación.

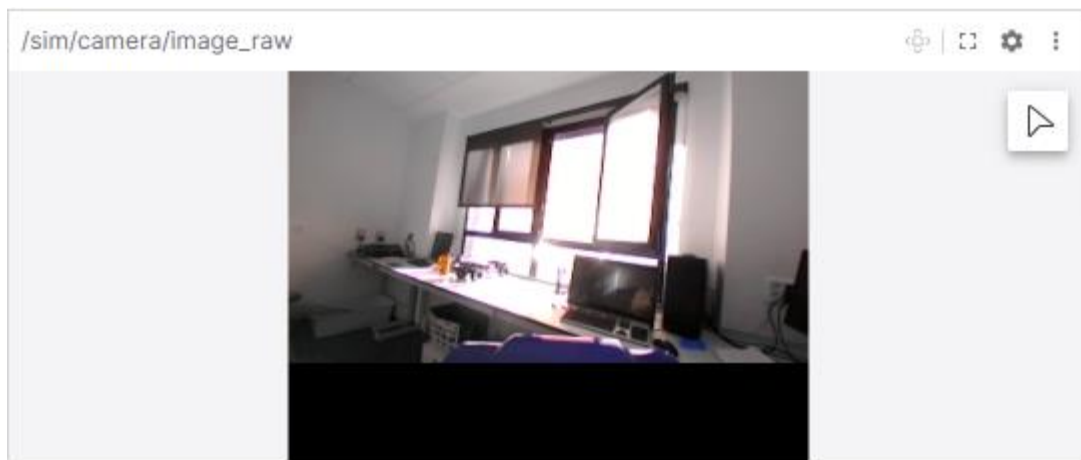


Figura 3-8 Imagen adquirida por la cámara RGB del sistema durante la operación del vehículo

3.1.5 Sensores Inerciales IMU

Phidget Spatial Precision 3/3/3

El sistema incorpora dos sensores inerciales *Phidget Spatial Precision 3/3/3*. Estos dispositivos pertenecen a la categoría de unidades de medida inercial (IMU) y están diseñados para proporcionar información precisa sobre el movimiento y la orientación de un sistema en el espacio tridimensional.

Cada unidad integra tres tipos de sensores: un acelerómetro, un giróscopo y un magnetómetro, todos ellos de tres ejes, lo que da lugar a un total de nueve grados de libertad (9-DOF) por dispositivo. La conexión con la unidad de procesamiento se realiza mediante interfaz USB, lo que facilita su integración directa con la Raspberry Pi [17].



Figura 3-9 Sensor inercial *PhidgetSpatial Precision 3/3/3* empleado para la medida de aceleraciones, velocidades angulares y orientación del vehículo. Fuente: [17]

Como se aprecia en la Figura 3-9, el sensor integra acelerómetro, giróscopo y magnetómetro en un único módulo, permitiendo estimar la orientación espacial del vehículo mediante la combinación de estas mediciones. La incorporación de una IMU resulta fundamental para analizar el comportamiento dinámico del UGV y complementar la información posicional proporcionada por el GNSS, especialmente en situaciones donde la señal pueda verse degradada.

Magnitudes Medidas y ejes de referencia

Las IMU proporcionan información sobre las principales magnitudes inerciales necesarias para caracterizar el movimiento del vehículo. El acelerómetro triaxial mide las aceleraciones lineales a lo largo de los tres ejes espaciales, mientras que el giróscopo triaxial permite registrar las velocidades angulares asociadas a las rotaciones del sistema. Por su parte, el magnetómetro triaxial proporciona información relativa al campo magnético terrestre, lo que resulta útil para la estimación de la orientación.

La combinación de estas magnitudes permite obtener una descripción completa del estado dinámico del vehículo, incluyendo su orientación, inclinación y movimientos relativos. El uso de sensores de tres ejes facilita la definición de un sistema de referencia cartesiano, fundamental para el análisis del comportamiento del vehículo durante las pruebas experimentales [9], [16].

Para interpretar correctamente las magnitudes medidas por la unidad inercial, es necesario definir el sistema de referencia asociado al sensor. La Figura 3-10 muestra la representación de los ejes cartesianos X, Y y Z utilizados por la IMU para registrar aceleraciones lineales y velocidades angulares.



Figura 3-10 Sistema de ejes de referencia de la unidad de medida inercial (IMU), mostrando las aceleraciones lineales y las rotaciones medidas en los tres ejes espaciales X, Y y Z. Fuente: Elaboración propia mediante generación asistida por inteligencia artificial

Parámetros técnicos

Las principales características técnicas de la IMU empleada se resumen en la Tabla 3-4.

Parámetro	Valor
Tipo de sensor	IMU (acelerómetro + giroscopio + magnetómetro)
Ejes de medida	3 ejes (X, Y, Z)
Rango acelerómetro	± 16 g
Resolución acelerómetro	30 μ g
Rango giroscopio	± 2000 °/s
Resolución giroscopio	0.004 °/s
Rango magnetómetro	± 8 G
Resolución magnetómetro	1.5 mG
Frecuencia de muestreo	Hasta 1 ms por muestra (≈ 1000 Hz)
Interfaz de comunicación	USB / VINT
Compatibilidad software	Phidget22
Alimentación	4.1 – 5.3 V (USB)
Consumo máximo	60 mA
Temperatura de operación	-40 °C a 85 °C
Limitaciones principales	Deriva, ruido, sensibilidad a vibraciones

Tabla 3-4 Resumen de las características principales de la IMU empleada. Fuente: [17]

Estas características permiten la adquisición de información dinámica del vehículo con alta frecuencia, lo que resulta adecuado para aplicaciones de estimación del movimiento, aunque con limitaciones asociadas al ruido y la deriva inherentes a este tipo de sensores.

Aplicación de los sensores iniciales en el proyecto y justificación en el proyecto

En este trabajo se han empleado dos unidades de medida inercial (IMU), integradas en el sistema con el objetivo de mejorar la fiabilidad de las mediciones.

El uso de sensores IMU permite obtener información continua sobre el estado dinámico del vehículo, lo que resulta especialmente útil en situaciones donde otros sensores, como el GNSS, pueden presentar limitaciones o pérdidas de señal. Esta información es clave para tareas de estimación del movimiento y análisis del comportamiento del sistema.

La elección de emplear dos IMU responde a la necesidad de aumentar la robustez del sistema frente a errores de medición y perturbaciones externas. La disponibilidad de múltiples sensores permite comparar las mediciones obtenidas y detectar posibles inconsistencias, lo que resulta especialmente relevante en entornos reales donde las condiciones de operación pueden afectar a la calidad de los datos.

Desde el punto de vista de la integración, las IMU se conectan a la unidad de procesamiento mediante interfaz USB, permitiendo la adquisición de datos en tiempo real. No obstante, durante la fase

de desarrollo se observó que la ubicación inicial de estos sensores en soportes rígidos impresos en 3D introducía vibraciones que afectaban negativamente a la calidad de las mediciones.

Como resultado, se optó por una solución de montaje basada en velcro adhesivo, que permite amortiguar parcialmente las vibraciones transmitidas desde la estructura del vehículo. Esta decisión permitió mejorar la estabilidad de las mediciones sin comprometer la facilidad de integración y ajuste de los sensores.

A modo ilustrativo, la Figura 3-11 muestra la ubicación de los sensores IMU en la plataforma, así como la solución de montaje adoptada.

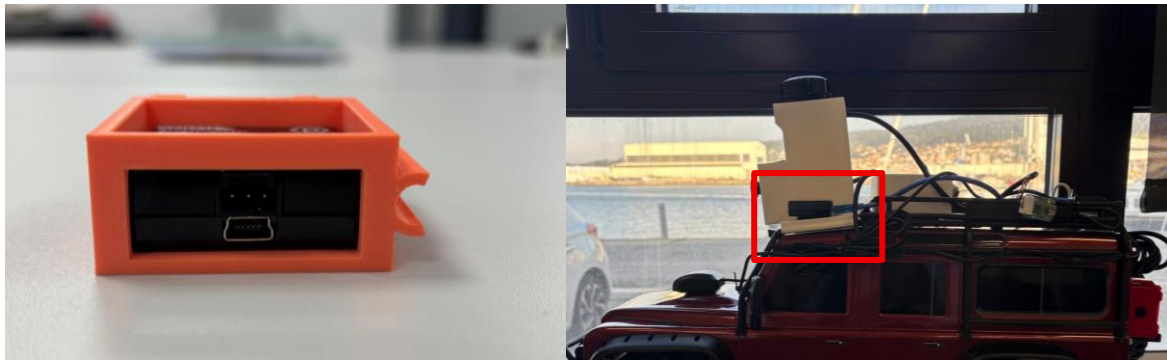


Figura 3-11 Diferencias entre los montajes confeccionados a lo largo del desarrollo del proyecto

Esta experiencia pone de manifiesto la importancia del diseño mecánico en la calidad de las mediciones sensoriales.

A pesar de estas mejoras, los sensores inerciales presentan limitaciones inherentes, como la acumulación de errores a lo largo del tiempo (deriva), lo que hace necesario su uso combinado con otros sensores dentro de una arquitectura multisensor.

En conjunto, el uso de dos IMU en el sistema permite mejorar la robustez y fiabilidad de las mediciones, constituyendo un elemento clave en la percepción del estado dinámico del vehículo.

3.1.6 Sistema de Posicionamiento Global (GNSS)

Receptor GNSS USB

El sistema de posicionamiento global empleado en el proyecto se basa en un receptor GNSS con conexión USB, diseñado para proporcionar información de localización a plataformas móviles y sistemas embebidos. La Figura 3-12 muestra el dispositivo empleado en el desarrollo del sistema. Este tipo de receptor permite la obtención de datos de posicionamiento mediante una única interfaz, que proporciona tanto la alimentación eléctrica como la comunicación con la unidad de procesamiento.

El receptor presenta un formato compacto e integra la antena necesaria para la recepción de señales procedentes de sistemas de navegación por satélite. La conexión directa mediante USB facilita su integración con la Raspberry Pi 4 Model B, sin necesidad de hardware adicional, reduciendo la complejidad del sistema electrónico y simplificando el montaje sobre la plataforma móvil.



Figura 3-12 Receptor GNSS con conexión USB empleado para la obtención de información de posicionamiento global del vehículo

Información proporcionada por el Sensor

El receptor GNSS proporciona información de posicionamiento absoluto del vehículo, incluyendo las coordenadas geográficas de latitud y longitud, así como otros parámetros asociados al movimiento, como la altitud.

Con el fin de comprender el funcionamiento del sistema de posicionamiento empleado, resulta necesario describir el principio general del sistema GNSS. La Figura 3-13 representa el esquema conceptual mediante el cual varios satélites transmiten señales al receptor instalado en el vehículo, permitiendo la determinación de su posición.

Los datos de posicionamiento se actualizan de forma periódica y son adecuados para aplicaciones de robótica móvil a baja velocidad y para la realización de pruebas experimentales en entornos exteriores. La precisión proporcionada por el sensor es suficiente para tareas de localización general y seguimiento de trayectorias, aunque no está orientada a aplicaciones que requieran una precisión al centímetro [12].

La información obtenida puede ser utilizada tanto en tiempo real como para el análisis posterior de los recorridos realizados por el vehículo durante las distintas fases de experimentación.



Figura 3-13 Esquema conceptual del sistema de posicionamiento global (GNSS), en el que varios satélites GNSS transmiten señales al vehículo, que actúa como receptor para la determinación de su posición. Fuente: Elaboración propia mediante generación asistida por inteligencia artificial

Características técnicas

Las principales características técnicas del módulo GNSS empleado se resumen en la Tabla 3-5.

Parámetro	Valor
Modelo	VK-162 G-Mouse USB GNSS
Tipo de sensor	GNSS (GPS)
Chipset	u-blox (serie 7)
Función principal	Posicionamiento global
Sistema soportado	GPS
Salida de datos	Coordenadas (latitud, longitud, altitud)
Formato de datos	NMEA 0183
Interfaz de comunicación	USB (emulación puerto serie)
Frecuencia de actualización	1 Hz (configurable hasta 10 Hz)
Precisión típica	~2.5 m
Tiempo de arranque (TTFF)	~1 s (hot start), ~30 s (cold start)
Antena	Integrada de alta sensibilidad
Alimentación	5 V (USB)
Compatibilidad	Windows / Linux
Integración	Directa con Raspberry Pi

Parámetro	Valor
Aplicaciones	Navegación, tracking, localización
Limitaciones	Sensible a obstrucciones, pérdida de señal en interiores

Tabla 3-5 Resumen de las características técnicas del GNSS empleado en el sistema. Fuente: [51]

Este tipo de módulos GNSS de bajo coste proporciona una solución adecuada para aplicaciones de navegación básica, aunque con limitaciones en precisión frente a sistemas de mayor gama.

Aplicación del GNSS en el proyecto y justificación en el proyecto

En este trabajo se ha empleado un módulo GNSS conectado a la unidad de procesamiento, encargado de proporcionar datos de posición geográfica en tiempo real.

La información proporcionada por este sensor resulta fundamental para la localización global del vehículo, permitiendo conocer su posición en coordenadas geográficas. Esta capacidad complementa la información relativa proporcionada por otros sensores, como las IMU, facilitando una estimación más completa del estado del sistema.

Desde el punto de vista de la integración, el módulo GNSS se conecta a la Raspberry Pi mediante interfaz USB o serie, permitiendo la recepción de datos en formato estándar (por ejemplo, tramas NMEA). Estos datos pueden ser procesados y utilizados dentro del sistema para tareas de monitorización y análisis de la trayectoria del vehículo.

No obstante, los sistemas GNSS presentan limitaciones importantes, especialmente en entornos donde la señal puede degradarse o perderse, como zonas urbanas, interiores o entornos con obstáculos. Además, la precisión de las mediciones puede verse afectada por factores como la geometría de los satélites o las condiciones ambientales [13], [14].

Por este motivo, el GNSS se emplea en combinación con otros sensores dentro de una arquitectura multisensor, permitiendo mejorar la robustez del sistema y compensar las limitaciones individuales de cada tecnología.

En conjunto, el sistema GNSS proporciona una referencia de posicionamiento absoluto que resulta esencial para la caracterización global del movimiento del vehículo.

A modo ilustrativo, la Figura 3-14 muestra la integración del sistema en el programa Foxglove.

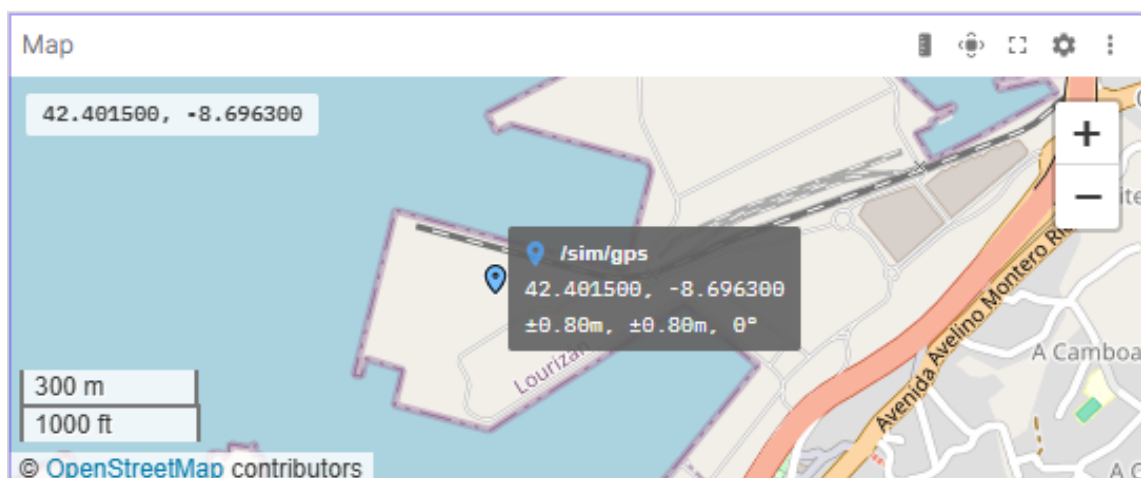


Figura 3-14 Integración del módulo GNSS dentro de Foxglove para la obtención de datos de posicionamiento

3.1.7 Resumen del Conjunto de Plataformas y Sensores

Tabla Resumen de plataformas y Sensores

Una vez descritos individualmente los distintos componentes que conforman el sistema, resulta conveniente sintetizar la información en un formato estructurado. La Tabla 3-6 resume las plataformas y sensores empleados, indicando su tipo, función principal y la interfaz de conexión utilizada para su integración con la unidad de procesamiento.

Elemento	Modelo	Tipo	Características principales	Interfaz
Plataforma móvil	Land Rover Defender TRX-4 (822256-4)	Vehículo terrestre RC	Tracción 4×4, alta estabilidad, capacidad de carga para sensores, operación a baja velocidad	—
Unidad de procesamiento	Raspberry Pi 4 Model B	SBC / Computador embebido	CPU quad-core ARM Cortex-A72 1.5 GHz, RAM LPDDR4, soporte Linux	USB, Ethernet, WiFi, Bluetooth
LiDAR	SLAMTEC RPLIDAR C1	Sensor de distancia 2D	Barrido 360°, tecnología ToF, hasta ~12 m de alcance, ~5 k muestras/s	USB
Cámara	SV-USBFHD08S-BH120	Cámara USB RGB	Resolución 2.0 MP, captura en color (RGB), alimentación 5 V	USB
IMU (×2)	PhidgetSpatial Precision 3/3/3 (MOT0110_0)	Sensor inercial	Acelerómetro 3 ejes, giróscopo 3 ejes, magnetómetro 3 ejes	USB
GNSS	Receptor GNSS USB (<i>G-Mouse / DIYmall</i>)	Sensor de posicionamiento	Latitud, longitud, altitud	USB

Tabla 3-6 Resumen de las plataformas y sensores empleados en el sistema, indicando el tipo de cada elemento, su función principal y la interfaz de conexión con la unidad de procesamiento

Relación entre sensores y funciones del sistema

Cada uno de los sensores integrados cumple una función específica dentro del sistema global. El sensor LiDAR proporciona información geométrica del entorno, permitiendo la detección de obstáculos y la representación espacial bidimensional del espacio circundante. La cámara de visión aporta información visual en color, complementando los datos geométricos con características visuales del entorno.

Por su parte, las unidades inerciales permiten caracterizar el movimiento y la orientación del vehículo, proporcionando información relevante sobre su estado dinámico durante la operación. Finalmente, el receptor GNSS aporta información de posicionamiento absoluto, situando al vehículo dentro de un marco de referencia global durante las pruebas en entornos exteriores.

La integración conjunta de estos elementos da lugar a un sistema versátil y modular, adecuado para la realización de experimentos relacionados con la percepción, la navegación y el análisis del comportamiento de plataformas móviles terrestres en un contexto académico.

Tras describir los distintos componentes que conforman el sistema, resulta necesario presentar una visión global de su integración. La Figura 3-15 muestra el esquema general del sistema, incluyendo la plataforma móvil, la unidad de procesamiento central y la interconexión funcional con los sensores instalados.

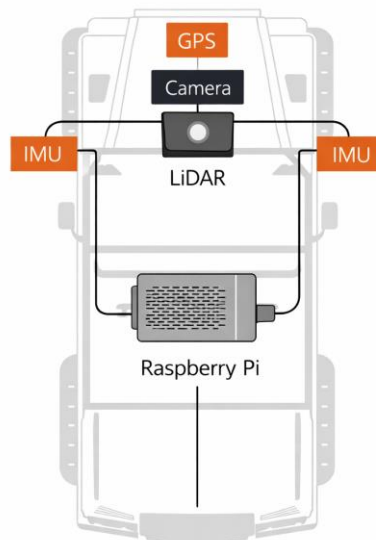


Figura 3-15 Boceto inicial de la planta del sistema mostrando la plataforma móvil, la unidad de procesamiento central (Raspberry Pi 4) y la relación funcional con los distintos sensores integrados. Fuente: Elaboración propia mediante generación asistida por inteligencia artificial

3.2 Diseño e integración de soportes para sensores

Para la integración de los distintos sensores y dispositivos electrónicos en el vehículo fue necesario diseñar una serie de soportes específicos que permitieran su fijación al chasis de forma segura y funcional. Estos soportes fueron desarrollados mediante modelado CAD utilizando el software Autodesk Inventor, lo que permitió adaptar cada pieza a las dimensiones y requisitos de los componentes del sistema.

El diseño de los soportes se llevó a cabo atendiendo a varios criterios fundamentales: estabilidad mecánica, facilidad de montaje, accesibilidad a los componentes y compatibilidad con la estructura existente del vehículo. En este sentido, se buscó evitar modificaciones permanentes en el chasis, priorizando soluciones reversibles y de bajo impacto.

El vehículo dispone de una estructura superior en forma de jaula tubular, la cual fue aprovechada como base para la integración de los soportes. Para ello, se diseñó un sistema de fijación basado en geometrías semicilíndricas huecas, cuyo diámetro coincide con el de los tubos de la estructura. Este diseño permite un acoplamiento por presión, proporcionando una solución sencilla, robusta y sin necesidad de elementos adicionales de fijación.

Inicialmente se planteó el diseño de tres tipos de soportes destinados a alojar el conjunto LiDAR–cámara, la unidad de procesamiento (Raspberry Pi) y las unidades de medición inercial (IMU). No obstante, tras las primeras pruebas de integración, se optó por simplificar el sistema eliminando los soportes específicos para las IMU, adoptando una solución alternativa basada en fijación mediante velcro adhesivo.

Los planos técnicos de las piezas diseñadas se recogen en el Anexo IV, donde se presenta el conjunto completo de documentación generada.

Soporte para LIDAR y CÁMARA

El primer soporte diseñado corresponde al conjunto principal, encargado de alojar el sensor LiDAR y la cámara USB. Este soporte constituye uno de los elementos más importantes del sistema, ya que ambos sensores proporcionan información fundamental sobre el entorno del vehículo.

El diseño de esta pieza se realizó de forma que el LiDAR se situara en la parte superior del soporte, mientras que la cámara se integrase en la parte frontal. Esta configuración permite que ambos sensores dispongan de un campo de visión adecuado sin interferir entre sí. Además, el soporte presenta una cierta elevación respecto al resto de componentes del vehículo, lo que contribuye a mejorar la visibilidad del LiDAR y a reducir posibles oclusiones provocadas por la propia estructura del vehículo.

El alojamiento de la cámara se diseñó específicamente para adaptarse a las dimensiones del dispositivo, garantizando que la lente quede correctamente orientada hacia el frente del vehículo. Además, se ha tenido en cuenta el paso del cable USB necesario para su conexión, incorporando el espacio suficiente para evitar tensiones o interferencias en el cableado.

En el caso del LiDAR, el soporte proporciona una base estable que permite fijar el sensor en una posición elevada y centrada, favoreciendo una correcta adquisición de datos del entorno. La integración de ambos sensores en una única pieza permite simplificar el montaje del sistema, reducir el número de elementos estructurales y mejorar la robustez del conjunto.

El modelo CAD de este soporte puede observarse en la Figura 3-16, mientras que la Figura 3-17 muestra la pieza una vez fabricada mediante impresión 3D. En la Figura 3-20 se puede apreciar además su integración final en el vehículo. Posteriormente el plano técnico detallado se incluye en el Anexo IV: Planimetría.

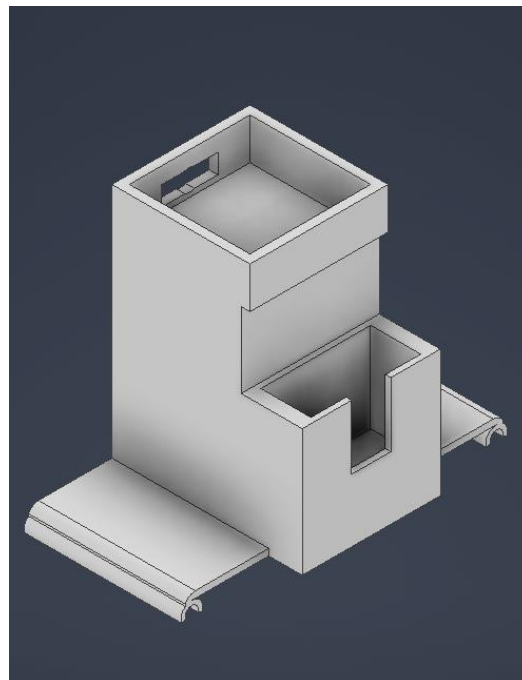


Figura 3-16 Modelo CAD del soporte diseñado en Autodesk Inventor para la integración conjunta del sensor LiDAR y la cámara USB en el vehículo de pruebas



Figura 3-17 Soporte impreso en 3D diseñado para la integración conjunta del sensor LiDAR y la cámara USB. La pieza permite alojar ambos dispositivos en una única estructura, posicionando el LiDAR en la parte superior y la cámara en la zona frontal del soporte

Soporte para la Raspberry Pi 4

El segundo soporte diseñado corresponde al alojamiento de la Raspberry Pi, que actúa como unidad de procesamiento principal del sistema. Este dispositivo se encarga de gestionar la adquisición de datos procedentes de los sensores y ejecutar los algoritmos necesarios para el funcionamiento del sistema.

El diseño del soporte se realizó con el objetivo de fijar la placa de forma segura dentro de la estructura superior del chasis, aprovechando el espacio disponible en el interior de la jaula del vehículo. Para ello, se tuvieron en cuenta tanto las dimensiones del dispositivo como la disposición de sus conectores.

Uno de los aspectos más relevantes del diseño fue garantizar que los puertos USB, Ethernet y de alimentación permanecieran accesibles, permitiendo la conexión de los distintos sensores y dispositivos externos sin dificultad. Para ello, el soporte presenta una abertura lateral que facilita el acceso a estos conectores.

Además, se dejó espacio suficiente alrededor de la placa para permitir una correcta ventilación del dispositivo, evitando posibles problemas de temperatura durante su funcionamiento. Este aspecto resulta especialmente importante en sistemas embebidos como la Raspberry Pi, que pueden generar calor durante la ejecución de tareas de procesamiento.

El modelo CAD de este soporte se muestra en la Figura 3-18, mientras que la Figura 3-19 presenta la pieza tras su fabricación mediante impresión 3D. En la Figura 3-20 se observa su instalación dentro de la estructura del vehículo. Posteriormente el plano técnico detallado se incluye en el Anexo IV: Planimetría.

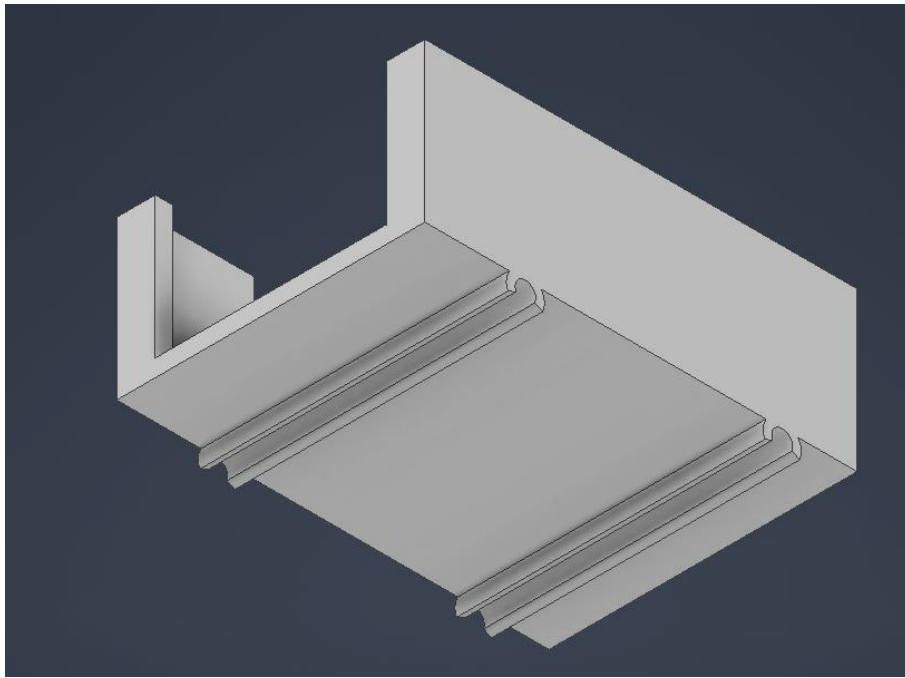


Figura 3-18 Modelo CAD del soporte diseñado en Autodesk Inventor para el alojamiento de la Raspberry Pi, adaptado a las dimensiones del dispositivo y a su integración en el chasis del vehículo



Figura 3-19 Soporte impreso en 3D destinado al alojamiento de la Raspberry Pi, diseñado para integrarse dentro de la estructura superior del vehículo y permitir el acceso a los distintos puertos de conexión

Integración de las unidades de medición inercial (IMU)

El sistema incorpora dos unidades de medición inercial (IMU) encargadas de proporcionar información sobre la aceleración y la orientación del vehículo. A diferencia del planteamiento inicial, en el que se diseñaron soportes específicos para estos sensores, finalmente se optó por una solución de fijación directa mediante velcro adhesivo.

Esta decisión se tomó tras evaluar las necesidades reales del sistema, considerando el reducido tamaño y peso de los sensores, así como la necesidad de flexibilidad en su posicionamiento. La utilización de velcro permite una fijación suficientemente estable para el tipo de ensayos realizados, al mismo tiempo que facilita la recolocación de los sensores en caso necesario.

Las IMU se fijaron sobre la base del soporte principal del conjunto LiDAR–cámara, lo que permite concentrar los sensores en una zona común del vehículo, simplificando el cableado y reduciendo la dispersión de componentes sobre el chasis. Esta configuración resulta adecuada para la adquisición de datos en el contexto del presente proyecto, donde no se requieren configuraciones distribuidas de sensores inerciales.

Fabricación de las piezas mediante impresión 3D

Una vez finalizado el diseño de los distintos soportes, se procedió a su fabricación mediante impresión 3D. Para la fabricación se utilizó filamento PLA, un material ampliamente empleado en prototipado debido a su facilidad de impresión, buena rigidez estructural y estabilidad dimensional.

Las piezas fueron impresas utilizando una impresora Bambu Lab, empleando los parámetros de impresión estándar proporcionados por el fabricante. Estos parámetros resultaron adecuados para la fabricación de los soportes, obteniendo piezas con suficiente precisión y resistencia para su uso en el sistema.

El uso de impresión 3D permitió fabricar rápidamente las piezas diseñadas, facilitando además la posibilidad de realizar modificaciones en el diseño tal y como se dio en alguna de las piezas diseñadas. Esta técnica resulta especialmente adecuada en fases de prototipado, donde la flexibilidad y la rapidez de fabricación son factores que ayudaron a la rápida corrección y modificación de errores encontrados en los soportes una vez finalizada su impresión, como fue el caso de las IMU.

Finalmente, una vez fabricadas todas las piezas, se procedió al montaje completo del sistema sobre el vehículo, integrando los sensores LiDAR, cámara, IMU, la Raspberry Pi y el receptor GNSS. El GNSS se fijó directamente al chasis mediante velcro adhesivo, debido a su reducido tamaño y peso, lo que permite una instalación sencilla sin necesidad de diseñar un soporte específico.

En la Figura 3-20 se muestra el montaje final del sistema de sensores sobre el vehículo.

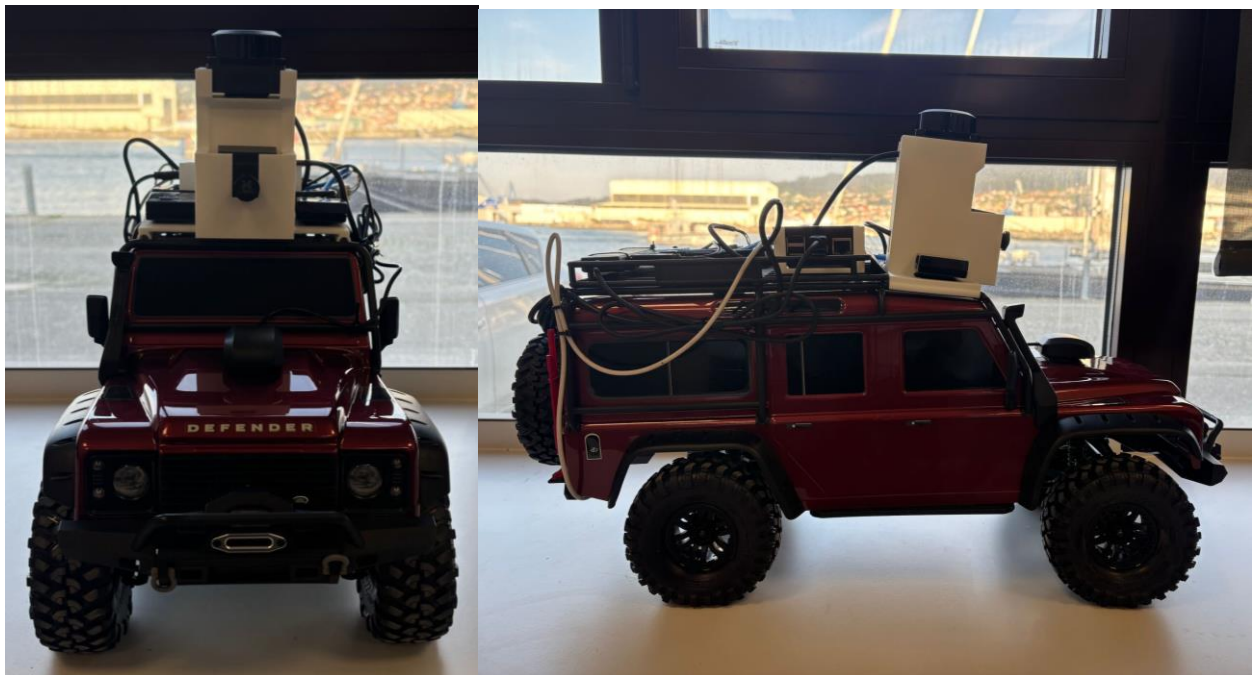


Figura 3-20 Vista lateral y frontal del vehículo con el sistema de sensores instalado mediante el uso de soportes

3.3 Desarrollo del sistema embarcado

El desarrollo del sistema embarcado se llevó a cabo sobre una plataforma Raspberry Pi, seleccionada por su bajo consumo, capacidad de procesamiento y compatibilidad con herramientas de robótica y adquisición de datos. A continuación, se describe detalladamente el procedimiento seguido, incluyendo la configuración del sistema, instalación de dependencias y validación de sensores.

3.3.1 Preparación del sistema operativo

En primer lugar, se procedió a la instalación del sistema operativo mediante la herramienta Raspberry Pi Imager, seleccionando como dispositivo la Raspberry Pi 4 y como sistema operativo la distribución Ubuntu 24.04 Server, elegida por su ligereza, su menor consumo de recursos y su compatibilidad con ROS 2 Jazzy. Asimismo, se configuró un usuario específico para el sistema y una contraseña.

A modo ilustrativo en la Figura 3-21 y la Figura 3-22 se aprecian las ventanas mostradas durante el proceso de instalación del sistema operativo, señalando con un recuadro rojo la selección hecha durante el proceso.

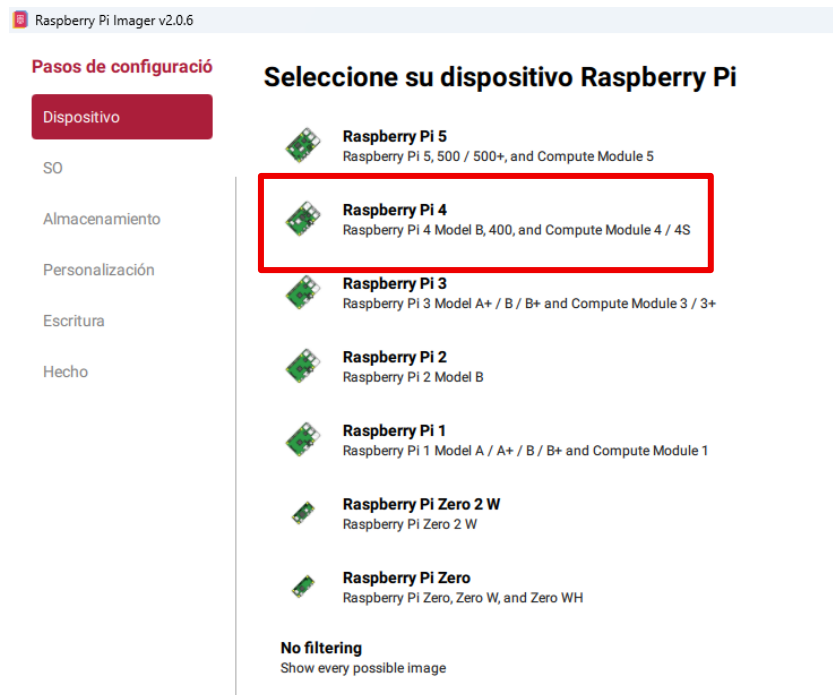


Figura 3-21 Ventana de elección del dispositivo dentro del proceso de configuración de la Raspberry Pi dentro del *Imager*

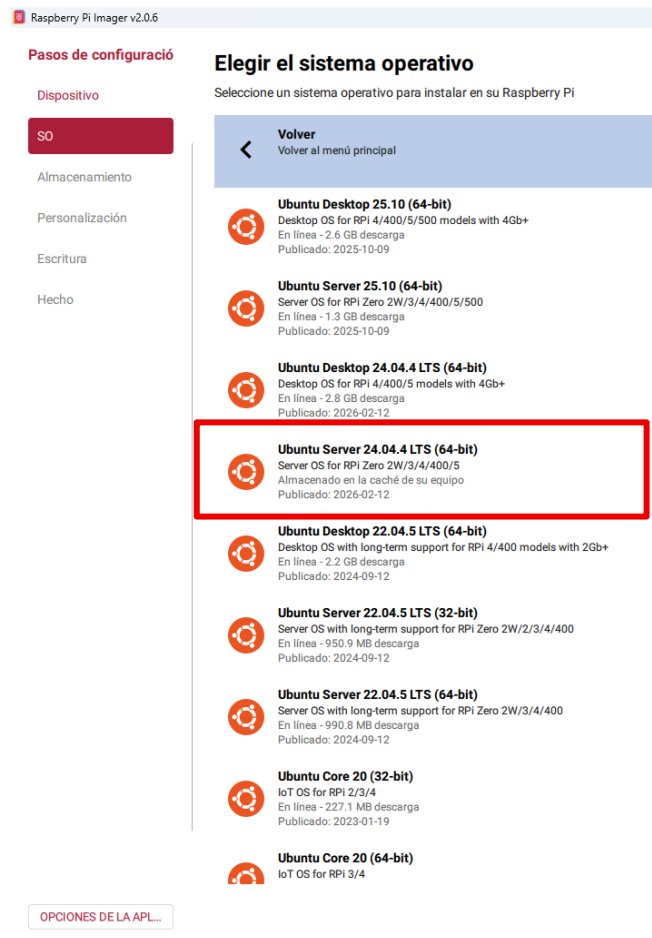


Figura 3-22 Ventana de elección del sistema operativo dentro del proceso de configuración de la Raspberry Pi dentro del *Imager*

Una vez grabada la imagen en la tarjeta microSD e insertada en la Raspberry Pi, se realizó el arranque inicial del sistema. Posteriormente, se actualizó el sistema mediante dos comandos:

- `sudo apt update`
- `sudo apt upgrade -y`

Esto permitió disponer de un entorno actualizado previo a la instalación de nuevas dependencias y librerías.

3.3.2 Configuración de red y acceso remoto

Dado que *Ubuntu Server* carece de una interfaz gráfica y con el objeto de garantizar la conectividad del sistema y permitir su acceso remoto, se llevó a cabo la configuración de red y la habilitación del servicio SSH.

En primer lugar, se instalaron herramientas de red adicionales mediante el siguiente comando:

- `sudo apt install net-tools wireless-tools wpasupplicant -y`

Estas herramientas permiten gestionar interfaces de red, conexiones inalámbricas y autenticación WiFi.

Configuración de Red WiFi

La configuración de la conexión inalámbrica se realizó mediante *Netplan*, que es el sistema de gestión de red empleado por *Ubuntu*. La configuración se aplica sobre la interfaz inalámbrica del sistema (*wlan0*), que representa el dispositivo WiFi de la Raspberry Pi. *Netplan* actúa como una capa de configuración que define los parámetros de red en archivos YAML y delega su aplicación en servicios del sistema, como *systemd-networkd*, encargado de gestionar las interfaces de red en segundo plano [52].

Antes de proceder a la configuración de red mediante *Netplan*, fue necesario verificar el correcto funcionamiento del servicio *systemd-networkd*. Para comprobar su estado, se utilizó el siguiente comando:

- `systemctl status systemd-networkd`

En el caso de que el servicio no se encontrase activo, se habilita mediante:

- `sudo systemctl enable systemd-networkd`
- `sudo systemctl start systemd-networkd`

La verificación de este servicio resulta fundamental para garantizar que la configuración de red definida después se aplique correctamente.

Tras verificar el *systemd-networkd* se configuró una red WiFi específica con el objetivo de que la Raspberry Pi se conectase automáticamente en cada arranque. Esta configuración resulta fundamental para garantizar que la Raspberry Pi y el equipo de desarrollo (donde se ejecuta *Foxglove*) se encuentren en la misma red, permitiendo la comunicación entre ambos dispositivos. La configuración se realizó editando el archivo correspondiente:

- `sudo nano /etc/netplan/01-netcfg.yaml`

Incluyendo la configuración mostrada en la Figura 3-23

```
network:
  version: 2
  wifis:
    wlan0:
      optional: true
      dhcp4: no
      addresses:
        - 10.42.0.126/24
      routes:
        - to: default
          via: 10.42.0.1
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
      access-points:
        "tfg":
          password: "tfgtfg2026"
```

Figura 3-23 Configuración del Netplan empleada durante la configuración de la unidad de procesamiento

Posteriormente se aplicaron los cambios mediante `sudo netplan apply`. En cuanto a la resolución de nombres de red, el sistema emplea servidores DNS que permiten traducir direcciones de dominio en direcciones IP. En este caso, al utilizar configuración automática mediante DHCP dentro del *netplan*, los servidores DNS se asignaron automáticamente por el *router*, sin necesidad de configuración manual adicional. Esta función resulta esencial para permitir el acceso a recursos externos, como repositorios de software.

Adicionalmente, con el objetivo de garantizar una conexión estable con el sistema embarcado, se configuró una dirección IP estática para la Raspberry Pi dentro de la red local. Esta decisión resulta especialmente relevante en el contexto del presente proyecto, ya que la visualización de datos mediante Foxglove se realiza a través de una conexión *WebSocket* que requiere especificar la dirección IP del dispositivo.

En configuraciones basadas en DHCP, la dirección IP puede variar entre reinicios, lo que obligaría a actualizar manualmente la dirección de conexión en cada sesión. Para evitar esta problemática, se estableció una IP fija asociada a la Raspberry Pi, asegurando que la dirección del servidor *WebSocket* permanezca constante.

Esta configuración permitió simplificar el acceso remoto al sistema, mejorar la estabilidad de la comunicación y facilitar las tareas de validación y monitorización en tiempo real.

Habilitar SSH

Para facilitar el control remoto del sistema, se habilitó el servicio SSH:

- `sudo apt install openssh-server -y`
- `sudo systemctl enable ssh`
- `sudo systemctl start ssh`

Esto permitió gestionar la Raspberry Pi desde el ordenador principal sin necesidad de periféricos adicionales.

A modo de síntesis, la Tabla 3-7 recoge los principales comandos empleados durante la configuración de red y habilitación del acceso remoto en el sistema embarcado.

Comando	Función	Descripción
<code>sudo apt install net-tools wireless-tools wpasupplicant -y</code>	Instalación de herramientas de red	Instala utilidades necesarias para la gestión de interfaces de red y conexiones WiFi
<code>systemctl status systemd-networkd</code>	Verificación del servicio de red	Comprueba si el servicio encargado de gestionar la red está activo
<code>sudo systemctl enable systemd-networkd</code>	Habilitar servicio	Configura el servicio para que se inicie automáticamente al arrancar el sistema
<code>sudo systemctl start systemd-networkd</code>	Iniciar servicio	Inicia manualmente el servicio de gestión de red
<code>sudo nano /etc/netplan/01-netcfg.yaml</code>	Configuración de red WiFi	Permite editar el archivo de configuración de red para definir la conexión inalámbrica
<code>sudo netplan apply</code>	Aplicar configuración	Aplica los cambios definidos en los archivos de Netplan
<code>sudo apt install openssh-server -y</code>	Instalación de SSH	Instala el servicio SSH para acceso remoto
<code>sudo systemctl enable ssh</code>	Habilitar SSH	Configura el servicio SSH para iniciarse automáticamente
<code>sudo systemctl start ssh</code>	Iniciar SSH	Activa el servicio SSH en el sistema

Tabla 3-7 Comandos utilizados en la configuración de red y acceso remoto

Por último, para facilitar la comprensión de la configuración de red y del acceso remoto al sistema embarcado, la Figura 3-24 muestra de forma esquemática la arquitectura de conectividad entre la Raspberry Pi y el equipo de desarrollo.

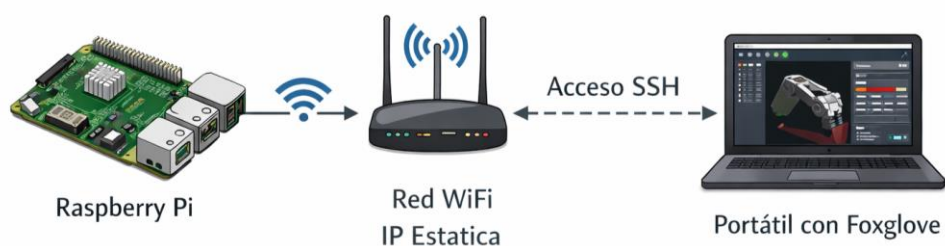


Figura 3-24 Esquema de conectividad y acceso remoto del sistema, donde la Raspberry Pi se conecta a la red WiFi mediante una dirección IP estática, permitiendo el acceso remoto vía SSH y la visualización de datos desde un equipo externo mediante Foxglove. Fuente: Elaboración asistida por herramientas de inteligencia artificial

3.3.3 Instalación de Paquetes Básicos y Herramientas del Sistema

Una vez completada la configuración inicial de red y acceso remoto, se procedió a la instalación de los paquetes básicos necesarios para el desarrollo y depuración del sistema. En esta fase se instalaron tanto utilidades de sistema como herramientas específicas para la detección de dispositivos y la gestión de dependencias.

En primer lugar, se ejecutó la actualización del índice de paquetes y la instalación del software base mediante los siguientes comandos:

- `sudo apt update`
- `sudo apt install -y python3-pip unzip curl wget git usbutils net-tools iw wireless-tools wpasupplicant`

Los paquetes instalados cumplen funciones diferenciadas dentro del sistema:

- *python3-pip*: permite instalar librerías de Python.
- *unzip*: se emplea para descomprimir archivos descargados.
- *curl* y *wget*: facilitan la descarga de ficheros y scripts desde terminal.
- *git*: permite clonar repositorios y gestionar versiones de código.
- *usbutils*: incorpora herramientas como *lsusb*, utilizadas para comprobar la detección de dispositivos USB.

La instalación de estos paquetes constituyó la base del entorno de trabajo, ya que permitió verificar la conectividad, gestionar la red y preparar el sistema para la integración de sensores.

Cabe destacar que algunas librerías utilizadas en el proyecto, como *pyserial*, *NumPy* o *SciPy*, no requirieron una instalación manual explícita, ya que fueron incorporadas automáticamente como dependencias de otros paquetes instalados mediante el gestor de paquetes del sistema, quedando instaladas en el directorio *dist-packages*.

Este comportamiento es habitual en sistemas basados en Linux, donde determinadas librerías se incluyen como dependencias para garantizar el correcto funcionamiento de herramientas relacionadas con la comunicación serie, el procesamiento de datos o la visión artificial.

En este caso, su disponibilidad en el sistema permitió su uso directo en los scripts desarrollados, sin necesidad de configuración adicional.

3.3.4 Configuración de permisos y gestión de paquetes Python

Durante la integración de dispositivos conectados mediante puertos serie, fue necesario ajustar los permisos del usuario para permitir el acceso sin necesidad de ejecutar los programas con privilegios de *superusuario*.

Con este objetivo, se añadió el usuario al grupo *dialout*:

- `sudo usermod -aG dialout $USER`

Este grupo permite el acceso a dispositivos serie como */dev/ttyACM0* o */dev/ttyUSB0*, empleados habitualmente por receptores GNSS y otros sensores conectados por USB-serie.

Por otro lado, en el caso de dispositivos de captura de imagen, como la cámara USB (*/dev/video0*), el acceso está gestionado por el grupo *video*. Si el sistema reconoce la cámara, pero el programa devuelve un error de tipo “Permiso denegado”, no es necesario ejecutar la aplicación con *sudo*. La solución adecuada consiste en añadir el usuario al grupo correspondiente:

- `sudo usermod -aG video $USER`

Tras la ejecución de estos comandos, es necesario cerrar la sesión o reiniciar el sistema para que los cambios en los permisos se apliquen correctamente.

Este enfoque permite ejecutar los scripts de adquisición de datos de forma segura y sin privilegios elevados, evitando problemas derivados de restricciones de acceso y garantizando un entorno de ejecución coherente. Además, evita el uso innecesario de `sudo`, que puede provocar inconsistencias en el acceso a librerías instaladas en el entorno del usuario.

Por otro lado, la instalación de determinadas librerías de Python en *Ubuntu 24.04* exigió el uso de la opción `--break-system-packages`. Esta opción fue necesaria debido a la política de protección del entorno Python del sistema implementada en versiones recientes de *Ubuntu*. En consecuencia, las librerías de Python se instalaron en el entorno global con comandos del tipo:

- `pip3 install --break-system-packages [nombre_paquete]`.

Este procedimiento permitió instalar paquetes necesarios para el proyecto sin utilizar entornos virtuales, manteniendo una configuración única y accesible desde cualquier directorio del sistema.

3.3.5 Instalación y Configuración de la IMU Phidget

Para la integración de la unidad de medida inercial *Phidget Spatial Precision 3/3/3*, fue necesario instalar tanto la librería de sistema como la librería Python asociada. Para ello se siguió la guía de instalación estipulada en la propia página web de *Phidget*. En primer lugar, se instaló la biblioteca del sistema:

- `sudo apt install -y libphidget22`.

A continuación, se instaló la librería Python oficial:

- `pip3 install --break-system-packages Phidget22`.

Además, para asegurar el acceso correcto al dispositivo USB sin necesidad de emplear `sudo`, fue necesario crear una regla `udev`. Para ello, se generó el archivo:

- `sudo nano /etc/udev/rules.d/99-libphidget22.rules`.

En dicho documento se estableció el siguiente contenido, tal y como marca la guía oficial de *phidget*:

- `SUBSYSTEMS=="usb", ACTION=="add", ATTRS{idVendor}=="06c2", MODE="0666"`

Posteriormente, se recargaron las reglas del sistema mediante:

- `sudo udevadm control --reload-rules`
- `sudo udevadm trigger`

Se reconectó físicamente la IMU para asegurar que la configuración fuese aplicada correctamente.

Posteriormente para comprobar que la Raspberry detectaba la IMU se realizaron dos pruebas. En primer lugar, se comprobó que el sistema detectaba la IMU conectada mediante `lsusb`.

Más tarde, se ejecutó un script de prueba en Python basado en la librería *Phidget22*, que permitió verificar la lectura de datos reales del sensor. El script establecía la conexión con el dispositivo, registraba una función de *callback* y mostraba por pantalla valores de aceleración, velocidad angular y campo magnético. Esta prueba confirmó tanto la correcta instalación de la librería como la operatividad del dispositivo.

Es importante señalar que, a diferencia de otros sensores del sistema, la IMU *Phidget* no depende de una ruta del tipo `/dev/ttyUSB0` o `/dev/ttyACM0`, ya que su acceso se realiza directamente a través de la librería oficial del fabricante.

3.3.6 Instalación y comprobación del receptor GNSS

Para el receptor GNSS se empleó un dispositivo USB con salida serie compatible con tramas NMEA. En este caso, fue necesario instalar la librería Python *pynmea2*, utilizada para interpretar los mensajes de posicionamiento.

La instalación se llevó a cabo mediante:

- `pip3 install --break-system-packages pynmea2`

Antes de desarrollar el script de lectura, se procedió a identificar el puerto serie asignado por el sistema al dispositivo. Para ello, se utilizaron los siguientes comandos:

- `ls /dev/ttyACM* /dev/ttyUSB* 2>/dev/null`
- `lsusb`

Esta comprobación permitió determinar si el receptor aparecía como `/dev/ttyACM0` o como `/dev/ttyUSB0`, evitando asumir de antemano un nombre de dispositivo incorrecto. En el desarrollo realizado, el GNSS fue reconocido como `/dev/ttyACM0`, y `lsusb` confirmó la detección del dispositivo correspondiente al fabricante u-blox.

Una vez identificado el puerto, se desarrolló un script de prueba en Python utilizando *pyserial* y *pynmea2*. El programa abría el puerto serie, leía las líneas NMEA recibidas y mostraba por pantalla aquellas que contenían información de posicionamiento. Con ello se verificó la capacidad de lectura de las tramas GNSS y la correcta interpretación de latitud y longitud.

Este procedimiento resultó esencial, ya que los dispositivos serie sí dependen del identificador `/dev/tty...`, el cual puede variar entre reinicios o según el orden de conexión de los periféricos USB.

3.3.7 Instalación y comprobación de la Cámara USB

Para la cámara USB se siguió un procedimiento diferente, basado en la detección de dispositivos de vídeo del sistema Linux. En primer lugar, se instalaron las herramientas necesarias para verificar la cámara y capturar imágenes desde Python:

- `sudo apt install -y v4l-utils python3-opencv.`

En este caso:

- *v4l-utils* proporciona herramientas de diagnóstico para dispositivos compatibles con Video4Linux.
- *python3-opencv* instala *OpenCV* para Python desde los repositorios del sistema.

La detección del dispositivo se comprobó mediante:

- `v4l2-ctl --list-devices`

Este comando permitió identificar que la cámara USB estaba asociada a los dispositivos `/dev/video0` y `/dev/video1`, siendo `/dev/video0` el canal principal de captura.

Inicialmente, se desarrolló un script en Python con *OpenCV* orientado a mostrar la imagen en una ventana mediante `cv2.imshow()`. Sin embargo, dado que la plataforma utilizada era Ubuntu Server sin interfaz gráfica, esta estrategia no resultó válida, ya que el sistema no disponía de entorno visual para representar la imagen.

Por este motivo, se desarrolló un segundo script de prueba específico, orientado a capturar un fotograma y almacenarlo en disco como archivo *.jpg*. Este enfoque permitió validar la cámara sin necesidad de interfaz gráfica, confirmando que OpenCV era capaz de abrir el dispositivo `/dev/video0`, capturar un frame y guardarlo correctamente.

Esta solución fue especialmente adecuada para un sistema embebido sin interfaz gráfica, y permitió verificar de forma objetiva el correcto funcionamiento de la cámara antes de su futura integración en ROS 2.

3.3.8 Instalación y comprobación del LIDAR

Para la integración del sensor LiDAR, se realizó previamente un análisis de las dependencias software necesarias para su correcto funcionamiento. Entre las librerías requeridas se identificaron NumPy y SciPy, utilizadas para el procesamiento de datos numéricos, así como pyserial, empleada en la gestión de comunicaciones a través de puertos serie.

Sin embargo, al abordar esta fase del desarrollo, se comprobó que dichas librerías ya se encontraban instaladas en el sistema. Esto se debe a que habían sido previamente incorporadas como dependencias de otros sensores integrados en etapas anteriores del proyecto, como el receptor GNSS o los módulos de adquisición de datos.

En consecuencia, no fue necesario realizar ninguna instalación adicional específica para el LiDAR, ya que el entorno de trabajo disponía de todos los recursos necesarios para su funcionamiento. Este hecho pone de manifiesto la reutilización eficiente de dependencias dentro del sistema, reduciendo la complejidad de integración de nuevos dispositivos y facilitando el desarrollo incremental del proyecto.

De este modo, el LiDAR pudo ser integrado en el sistema aprovechando la infraestructura software ya existente, quedando preparado para su utilización dentro del flujo general de adquisición y procesamiento de datos.

3.3.9 Resumen Librerías Instaladas

A modo de síntesis se proporciona a continuación en las Tabla 3-8 y Tabla 3-9 un resumen de las librerías y dependencias instaladas tanto para los propios sensores como para la propia unidad de procesamiento separándolas en dos categorías: *apt* y *pip*.

Paquetes Instalados mediante *apt*

Paquete	Tipo de uso	Descripción técnica
<i>python3-pip</i>	Sistema	Gestor de paquetes para la instalación de librerías Python
<i>unzip</i>	Sistema	Herramienta para la descompresión de archivos
<i>curl</i>	Sistema	Utilidad para la descarga de recursos desde terminal
<i>wget</i>	Sistema	Herramienta alternativa para descarga de archivos
<i>git</i>	Desarrollo	Sistema de control de versiones para gestión de código
<i>usbutils</i>	Diagnóstico	Permite la detección de dispositivos USB mediante <i>lsusb</i>
<i>net-tools</i>	Red	Herramientas clásicas de red (<i>ifconfig</i> , <i>netstat</i>)
<i>iw</i>	Red	Configuración avanzada de interfaces inalámbricas
<i>wireless-tools</i>	Red	Utilidades complementarias para redes WiFi
<i>wpa_supplicant</i>	Red	Gestión de autenticación en redes WiFi protegidas
<i>libphidget22</i>	IMU	Librería base para comunicación con dispositivos <i>Phidget</i>
<i>v4l-utils</i>	Cámara	Herramientas de diagnóstico para dispositivos Video4Linux
<i>python3-opencv</i>	Cámara	Librería para captura y procesamiento de imágenes
<i>openssh-server</i>	Sistema	Servicio para acceso remoto mediante protocolo SSH

Tabla 3-8 Resumen de los diferentes paquetes instalados mediante *apt*.

Paquetes Instalados mediante *pip*

Librería	Tipo de uso	Descripción técnica
<i>Phidget22</i>	IMU	Librería Python para la adquisición de datos de la IMU <i>Phidget</i>
<i>pynmea2</i>	GNSS	Librería para interpretación de tramas NMEA

Tabla 3-9 Resumen de las diferentes librerías instaladas mediante *pip*.

Como se puede observar en las Tablas, Tabla 3-8 y Tabla 3-9, las librerías específicas de sensores se instalaron mediante *pip*, como *Phidget22* y *pynmea2*, mientras que otras dependencias como OpenCV se instalaron mediante el gestor de paquetes del sistema para garantizar compatibilidad y estabilidad.

3.3.10 Instalación y Configuración de ROS 2 Jazzy

Una vez comprobado el funcionamiento individual de los sensores, se procedió a la instalación de ROS 2 Jazzy, con el objetivo de disponer de una infraestructura de comunicación entre nodos y facilitar la futura integración multisensorial.

Durante el proceso de instalación de ROS 2 Jazzy en la Raspberry Pi, se detectaron problemas en la resolución de dependencias al intentar instalar los paquetes binarios mediante el gestor APT. En concreto, el sistema devolvía errores relacionados con paquetes “no instalables” o con conflictos de versiones.

Tras el análisis del problema, se identificó que la causa estaba asociada a la configuración de los repositorios de Ubuntu en la plataforma Raspberry Pi. Por defecto, la instalación de Ubuntu Server para arquitectura ARM no incluye todas las ramas de actualización estándar, en particular dos: *noble-updates* y *noble-backports* [53], [54]. La ausencia de estas ramas impide que el sistema acceda a versiones actualizadas de ciertos paquetes y dependencias necesarias para ROS 2, lo que provoca incompatibilidades durante la instalación. Para solucionar este problema, se procedió a la edición del archivo de configuración de repositorios:

- `sudo nano /etc/apt/sources.list.d/ubuntu.sources`

Modificando la sección correspondiente para incluir las ramas adicionales:

- `Suites: noble noble-updates noble-backports`

Manteniendo la sección de seguridad sin modificaciones:

- `Suites: noble-security`

Una vez realizados los cambios, se actualizaron los repositorios del sistema mediante un `sudo apt update`.

Tras la configuración de repositorios, se instaló ROS 2.

- `sudo apt install -y ros-jazzy-desktop`

Una vez completada la instalación, se verificó la presencia del entorno ROS en la ruta `/opt/ros/jazzy`, posteriormente se cargó el entorno en la sesión actual:

- `source /opt/ros/jazzy/setup.bash`

Con el objetivo de automatizar la carga del entorno de trabajo en cada nueva sesión de terminal, se procedió a modificar el archivo de configuración del usuario, `.bashrc`, el cual se ejecuta automáticamente al iniciar una nueva terminal en sistemas Linux.

Para ello, se editó dicho archivo mediante el comando:

- `nano /home/jagprietto/.bashrc`
- `source /opt/ros/jazzy/setup.bash`
- `source /home/jagprietto/sim_sensors_foxglove/install/setup.bash`

La primera línea permite cargar el entorno global de ROS 2, mientras que la segunda incorpora el *workspace* específico del proyecto, incluyendo los paquetes desarrollados.

Gracias a esta configuración, el entorno se inicializa automáticamente al abrir una nueva terminal, evitando errores del tipo “*package not found*” y eliminando la necesidad de ejecutar manualmente los comandos de configuración en cada sesión.

Esta mejora permitió aumentar la robustez del sistema y reducir errores operativos durante la fase experimental.

La correcta instalación se validó ejecutando:

- `ros2`
- `ros2 topic list`

Se obtuvo la respuesta esperada del sistema y la visualización de los tópicos básicos `/rosout` y `/parameter_events`.

3.3.11 Integración del sistema de visualización mediante Foxglove Bridge

Una vez completada la instalación y verificación del entorno ROS 2, se procedió a la integración de una herramienta de visualización que permitiese monitorizar en tiempo real los datos generados por el sistema. Para ello, se seleccionó Foxglove, una plataforma orientada a la visualización de datos en sistemas robóticos.

Con el objetivo de establecer la comunicación entre la Raspberry Pi y el entorno de visualización en el equipo de desarrollo, se utilizó el paquete Foxglove Bridge, el cual permite exponer los tópicos de ROS 2 mediante una conexión WebSocket [54].

Instalación de Foxglove Bridge

La instalación del paquete se realizó a través del gestor APT, aprovechando la integración directa con ROS 2 Jazzy:

- `sudo apt install -y ros-jazzy-foxglove-bridge`

Este paquete forma parte del ecosistema de ROS 2 y permite convertir los tópicos internos del sistema en un flujo de datos accesible externamente.

Ejecución del Bridge

Una vez instalado, el bridge se lanzó mediante el siguiente comando:

- `ros2 launch foxglove_bridge foxglove_bridge_launch.xml port:=8765`

Durante la ejecución, el sistema inicializa un servidor *WebSocket* que queda escuchando, por defecto, en el puerto 8765.

Asimismo, en la salida del terminal se observa la publicación automática de los tópicos básicos del sistema, como */rosout* o */parameter_events*, lo cual confirma el correcto funcionamiento del bridge y su integración con *ROS 2*.

Conexión con el entorno de visualización

Para establecer la visualización de datos, se utilizó la aplicación *Foxglove* instalada en el equipo de desarrollo. La conexión se realizó mediante una dirección *WebSocket* de la forma:

- `ws://<IP_RASPBERRY>:8765`

Donde *<IP_RASPBERRY>* corresponde a la dirección IP asignada a la Raspberry Pi dentro de la red local.

Para garantizar el correcto funcionamiento de esta conexión, fue necesario que tanto la Raspberry Pi como el equipo de desarrollo se encontraran conectados a la misma red WiFi, previamente configurada en el sistema mediante *Netplan*.

Verificación del sistema

Una vez establecida la conexión, se comprobó que *Foxglove* era capaz de detectar y mostrar los tópicos disponibles en *ROS 2*. En esta fase inicial, únicamente se visualizaron los tópicos por defecto del sistema, ya que todavía no se habían integrado los nodos correspondientes a los sensores.

No obstante, esta prueba permitió validar tres aspectos importantes: la correcta instalación de *Foxglove Bridge*, la conectividad en red entre dispositivos y por último la exposición de datos *ROS 2* mediante *WebSocket*.

Consideraciones de uso

Es importante destacar que el proceso *foxglove_bridge* debe permanecer en ejecución para mantener activa la comunicación. En caso de detener el proceso, la conexión con *Foxglove* se interrumpe automáticamente.

Asimismo, esta arquitectura permite desacoplar la adquisición de datos del sistema embarcado y su visualización, facilitando tareas de depuración, análisis y validación del sistema en tiempo real.

El diseño de la interfaz de visualización y la interpretación de los datos obtenidos se abordarán en el capítulo de validación del sistema.

Con el objetivo de representar de forma global el flujo de integración software del sistema, la Figura 3-25 muestra la cadena completa desde la adquisición de datos sensoriales hasta su visualización en el entorno externo.

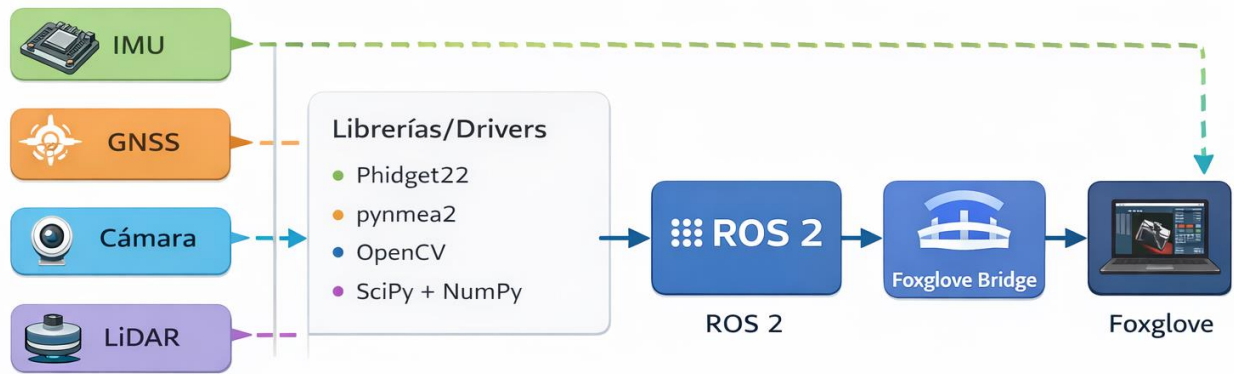


Figura 3-25 Arquitectura del flujo de datos software del sistema multisensor, desde la adquisición sensorial y su procesamiento mediante librerías específicas, hasta su distribución en ROS 2 y su visualización remota mediante Foxglove Bridge. Fuente: Elaboración asistida por herramientas de inteligencia artificial

3.4 Arquitectura del sistema

La arquitectura del sistema se diseñó con el objetivo de integrar la adquisición de datos multisensor, su procesamiento en una plataforma embarcada y su visualización en tiempo real en un equipo externo. Para ello, se adoptó una estructura modular basada en ROS 2, que permite desacoplar la captura de datos, su distribución interna y su representación, facilitando la escalabilidad y mantenibilidad del sistema.

El sistema se organiza en cuatro bloques principales: sensorial, procesamiento embarcado, comunicaciones internas mediante ROS 2 y visualización externa.

3.4.1 Bloques funcionales del sistema

Como se puede observar en la Tabla 3-10, la arquitectura general se divide en los siguientes bloques funcionales:

Bloque	Función principal
Sensores	Adquisición de datos físicos del entorno y del vehículo
Raspberry Pi	Procesamiento de datos y ejecución de nodos
ROS 2	Gestión de la comunicación mediante tópicos
Visualización externa	Representación de datos en tiempo real

Tabla 3-10 Resumen de los bloques funcionales del sistema

El bloque de sensores constituye la fuente de información del sistema, mientras que la Raspberry Pi actúa como unidad central de procesamiento. Sobre esta se ejecuta ROS 2, encargado de estructurar la comunicación entre módulos. Finalmente, Foxglove permite la visualización remota de los datos generados

3.4.2 Arquitectura Hardware

Desde el punto de vista hardware, la Raspberry Pi actúa como unidad central del sistema embarcado. A ella se conectan los distintos sensores empleados en el proyecto:

- Unidad de medida inercial (IMU) basada en tecnología Phidget
- Receptor GNSS para posicionamiento
- Cámara USB para captura de imágenes

- Sensor LiDAR 2D

Estos dispositivos se conectan mediante interfaces USB, bien directamente o a través de un hub USB alimentado, en función del número de periféricos utilizados simultáneamente.

La Raspberry Pi se conecta a la red WiFi configurada mediante *Netplan*, compartiendo red con el equipo externo en el que se ejecuta *Foxglove*, lo que permite la transmisión de datos en tiempo real.

De forma esquemática, la arquitectura hardware puede visualizarse en la Figura 3-26 mostrada a continuación.

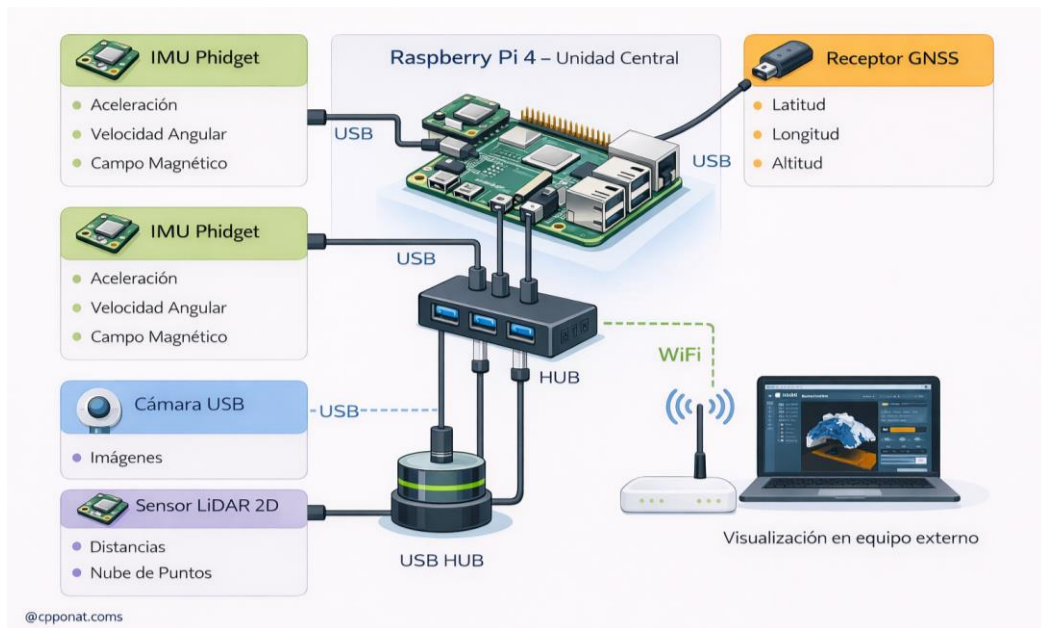


Figura 3-26 Esquema de la arquitectura hardware del sistema. Fuente: Elaboración asistida por herramientas de inteligencia artificial

3.4.3 Arquitectura Software

Desde el punto de vista software, la arquitectura se basa en una estructura modular en varias capas. En una primera capa se encuentran los scripts de adquisición, desarrollados en Python, encargados de comunicarse directamente con cada sensor mediante sus librerías específicas:

- *Phidget22* para la IMU
- *pynmea2* y acceso serie para el GNSS
- *OpenCV* para la cámara
- *SciPy* y *Numpy* para el LiDAR

En una segunda capa, estos datos se integran en *ROS 2*, donde cada fuente de información se representa como un nodo independiente que publica datos en tópicos.

En una tercera capa, el paquete *Foxglove Bridge* actúa como intermediario, exponiendo los tópicos de *ROS 2* mediante un servidor *WebSocket*.

Finalmente, en la capa externa, la aplicación *Foxglove* se conecta a dicho servidor y permite la visualización de los datos en tiempo real.

Esta arquitectura permite separar claramente la adquisición, la comunicación y la visualización, evitando dependencias entre módulos y facilitando el desarrollo incremental del sistema.

La Figura 3-25 muestra de forma esquemática la arquitectura modular en la que se basa el sistema.

3.4.4 Flujo de datos

El sistema gestiona distintos tipos de datos en función del sensor utilizado. A continuación, en la Tabla 3-11 se resume el flujo de información.

Sensor	Tipo de dato	Interfaz	Imagen
IMU Phidget	Aceleración, velocidad angular, campo magnético	USB (librería Phidget22)	
GNSS	Posición (latitud, longitud)	Serie (<i>/dev/ttyACM0</i>)	
Cámara USB	Imágenes (frames)	<i>/dev/video0</i> (OpenCV)	
LiDAR	Distancias / nube de puntos	USB o serie (según modelo)	

Tabla 3-11 Flujos de Información según sensor, tipo de dato, interfaz y estado.

Cada sensor genera datos de naturaleza distinta, por lo que se emplean interfaces de acceso específicas. Una vez adquiridos, estos datos son procesados y publicados en *ROS 2*, donde quedan disponibles para su consumo por otros nodos o herramientas externas.

Posteriormente, *Foxglove Bridge* convierte estos datos en un flujo accesible mediante *WebSocket*, permitiendo su transmisión al equipo de visualización. A modo ilustrativo la Figura 3-27 muestra la interfaz de *Foxglove* con la transmisión de datos de los sensores

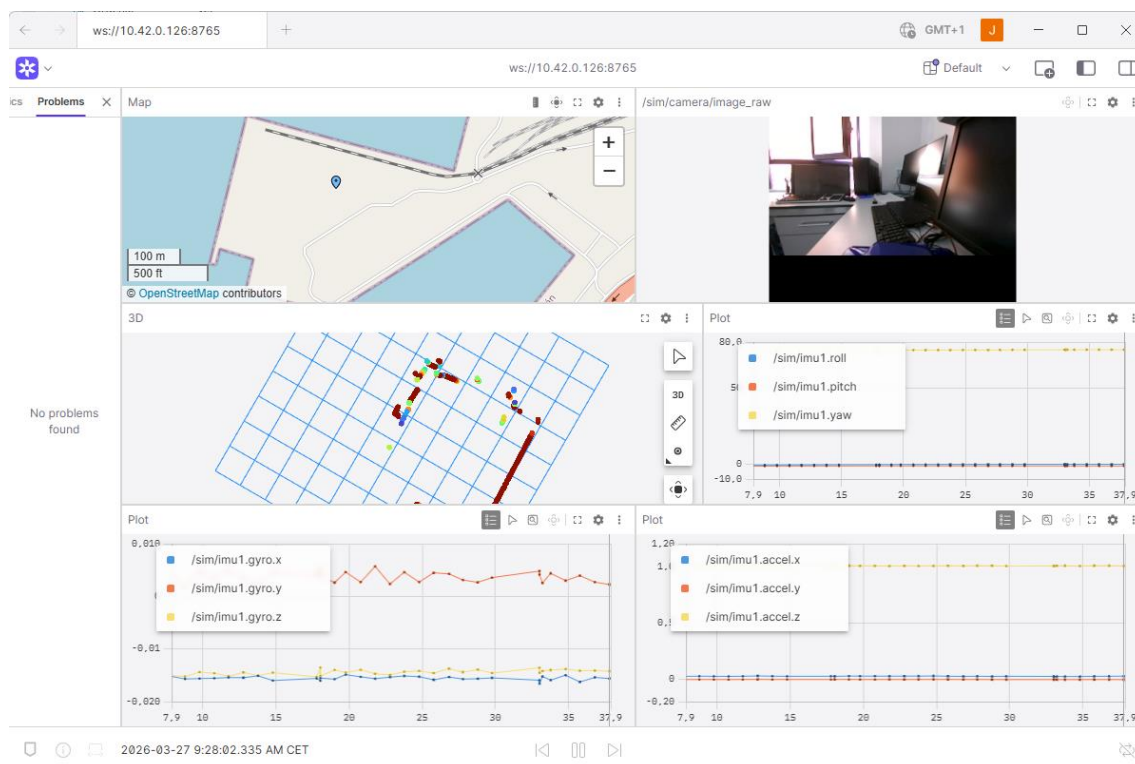


Figura 3-27 Interfaz de Foxglove con flujo de datos de los distintos sensores empleados

3.4.5 Justificación de la Arquitectura Adoptada

La arquitectura modular propuesta ha sido diseñada atendiendo a criterios de escalabilidad, mantenibilidad y eficiencia computacional, aspectos fundamentales en el desarrollo de sistemas multisensor embarcados.

En primer lugar, la descomposición del sistema en módulos independientes permite el desarrollo, validación y depuración individual de cada sensor, reduciendo la complejidad global durante las fases iniciales del proyecto. Este enfoque facilita la identificación de errores y permite aislar problemas específicos sin comprometer el funcionamiento del resto del sistema.

En segundo lugar, la adopción de *ROS 2* como middleware de comunicación proporciona una estructura distribuida basada en nodos y tópicos, que favorece la integración de sensores heterogéneos de forma desacoplada. Esta arquitectura no solo mejora la escalabilidad del sistema, sino que permite reutilizar componentes software y adaptarlos a diferentes configuraciones sin necesidad de rediseño completo.

Asimismo, la separación entre adquisición, procesamiento y visualización constituye una decisión clave desde el punto de vista de la eficiencia del sistema. La utilización de *Foxglove Bridge* para la visualización remota permite descargar a la Raspberry Pi de tareas gráficas, optimizando el uso de recursos computacionales y garantizando un funcionamiento más estable en una plataforma con capacidades limitadas.

Desde el punto de vista de la gestión temporal, la arquitectura basada en nodos independientes y temporizadores permite adaptar la frecuencia de adquisición a las características de cada sensor. Esto evita bloqueos en el sistema y mejora la robustez frente a cuellos de botella, especialmente en escenarios donde coexisten sensores con diferentes tasas de muestreo.

Por otro lado, el diseño adoptado favorece la integración progresiva de nuevos dispositivos, tanto a nivel hardware como software. La incorporación de un nuevo sensor se limita a la implementación de

un nuevo nodo y su correspondiente publicación en *ROS 2*, sin necesidad de modificar la estructura existente. Esta característica resulta especialmente relevante en entornos de investigación y prototipado.

Finalmente, la arquitectura propuesta se alinea con las tendencias actuales en sistemas robóticos, donde se priorizan soluciones distribuidas, modulares y orientadas a la interoperabilidad. Esto no solo valida las decisiones adoptadas en el contexto del proyecto, sino que garantiza su compatibilidad con desarrollos futuros más complejos.

3.4.6 Consideraciones y limitaciones del sistema

Durante el diseño e implementación del sistema se identificaron diversas consideraciones prácticas que han condicionado tanto la configuración como las decisiones adoptadas a nivel de arquitectura. El análisis de estos aspectos ha permitido extraer conclusiones relevantes en términos de robustez, escalabilidad y fiabilidad del sistema.

En primer lugar, la conexión simultánea de múltiples dispositivos USB puso de manifiesto las limitaciones de la Raspberry Pi en cuanto a suministro eléctrico. Inicialmente, se intentó conectar todos los sensores a través de un único hub USB, lo que provocó problemas durante el arranque del sistema, incluyendo fallos en la detección de dispositivos y comportamientos inestables. Como solución, se optó por distribuir la carga conectando parte de los sensores directamente a la Raspberry Pi y el resto a través del hub USB. Esta configuración permitió estabilizar el sistema y garantizar un funcionamiento correcto, evidenciando la importancia de una adecuada gestión de la alimentación en sistemas multisensor.

En segundo lugar, se identificó la naturaleza dinámica de los dispositivos serie en sistemas Linux. Sensores como el receptor GNSS o el LiDAR pueden ser asignados a diferentes identificadores (`/dev/ttyACM0`, `/dev/ttyUSB0`) en función del orden de conexión o del arranque del sistema. Este comportamiento introduce una fuente de incertidumbre que debe ser gestionada mediante mecanismos más robustos, como reglas *udev* o la asignación fija de puertos físicos.

Asimismo, durante la fase de integración mecánica, se detectó que la ubicación de las unidades de medida inercial (IMU) influía significativamente en la calidad de los datos adquiridos. El diseño inicial contemplaba soportes específicos situados en los extremos del vehículo; sin embargo, esta configuración introducía vibraciones que degradaban las mediciones. Como consecuencia, se optó por eliminar dichos soportes y reubicar las IMU en una zona más centrada del sistema, utilizando fijación mediante velcro adhesivo. Esta solución permitió reducir el ruido en las medidas y mejorar la estabilidad de los datos, poniendo de manifiesto la importancia del diseño mecánico en sistemas de percepción.

Por otro lado, la utilización de *Ubuntu 24.04 Server* sin entorno gráfico condicionó la estrategia de validación de determinados dispositivos, especialmente en el caso de la cámara. La imposibilidad de emplear herramientas de visualización directa obligó a desarrollar métodos alternativos, como la captura y almacenamiento de imágenes en disco, lo que permitió verificar el funcionamiento del sistema de forma indirecta.

En cuanto a la comunicación, la integración con Foxglove evidenció la dependencia de la conectividad de red para la monitorización del sistema. La comunicación mediante *WebSocket* requiere que tanto la Raspberry Pi como el equipo de visualización se encuentren en la misma red, lo que motivó la configuración de una conexión WiFi automática mediante *Netplan* y el uso de una dirección IP estática.

Finalmente, de manera global, se concluye que el desarrollo de un sistema multisensor embarcado no solo implica la integración de hardware y software, sino también la gestión de aspectos físicos y de sistema como la alimentación, la disposición de los sensores, la identificación de dispositivos y la conectividad. La arquitectura modular basada en *ROS 2* ha permitido mitigar parte de estas limitaciones, facilitando la escalabilidad del sistema y su adaptación a futuras ampliaciones.

3.5 Implementación Software del Sistema Multisensor

3.5.1 Diseño del nodo ROS

Una vez definida la arquitectura del sistema, se procedió al desarrollo de un nodo ROS 2 encargado de la adquisición, procesamiento y publicación de datos multisensor. Este nodo, denominado *multi_sensor_sim*, ha sido implementado en Python y constituye el núcleo software del sistema.

El diseño adoptado se basa en una arquitectura centralizada, en la que un único nodo gestiona la interacción con todos los sensores, encapsulando la lógica de adquisición y publicación de datos. Esta decisión se tomó con el objetivo de simplificar la implementación y facilitar la integración inicial del sistema, permitiendo validar el funcionamiento conjunto de todos los dispositivos en una única estructura.

El nodo hereda de la clase Node de ROS 2, lo que permite su integración directa dentro del ecosistema basado en comunicación mediante tópicos. Durante su inicialización, se configuran los distintos parámetros del sistema, incluyendo los sistemas de referencia, las frecuencias de ejecución y la configuración específica de cada sensor.

Un fragmento representativo de la inicialización del nodo se muestra a continuación en la Figura 3-28

```
class MultiSensorSim(Node):
    def __init__(self):
        super().__init__("multi_sensor_sim")

        self.pub_imu1 = self.create_publisher(ImuDetails, "/sim/imu1", 10)
        self.pub_gps = self.create_publisher(NavSatFix, "/sim/gps", 10)
        self.pub_scan = self.create_publisher(LaserScan, "/sim/lidar/scan", 10)
        self.pub_img = self.create_publisher(Image, "/sim/camera/image_raw", 10)
```

Figura 3-28 Fragmento de la inicialización del nodo dentro del script *multi_sensor_sim*

Este diseño permite centralizar la gestión de los sensores y garantizar una estructura coherente de publicación de datos dentro del sistema.

3.5.2 Gestión Temporal y ejecución del sistema

La ejecución del sistema se basa en un modelo orientado a eventos mediante temporizadores independientes, lo que permite gestionar de forma desacoplada la adquisición de cada sensor.

En el nodo desarrollado, se implementan múltiples temporizadores asociados a distintas frecuencias de ejecución, tal y como se muestra en el siguiente fragmento de código en la Figura 3-29

```
self.timer_main = self.create_timer(1.0 / self.rate_hz, self.tick_main)
self.timer_cam = self.create_timer(1.0 / self.camera_rate_hz, self.tick_camera)
self.timer_lidar = self.create_timer(1.0 / self.lidar_rate_hz, self.tick_lidar)
```

Figura 3-29 Fragmento de inicialización de temporizadores dentro del script *multi_sensor_sim*

Este enfoque permite adaptar la frecuencia de adquisición a las características de cada sensor, evitando bloqueos y optimizando el uso de los recursos disponibles en la Raspberry Pi.

El temporizador principal (*tick_main*) se encarga de la lectura de sensores como la IMU y el GNSS, así como de la publicación de la odometría del sistema. Por otro lado, los temporizadores específicos gestionan sensores con mayores requerimientos de procesamiento, como la cámara o el LiDAR.

Esta arquitectura permite un funcionamiento eficiente en sistemas con recursos limitados, aunque introduce una limitación importante: la ausencia de sincronización estricta entre sensores. No obstante, para el alcance del presente proyecto, esta aproximación resulta suficiente y simplifica significativamente la implementación.

3.5.3 Adquisición y procesamiento de datos

La adquisición de datos en el sistema desarrollado se realiza de forma centralizada dentro del nodo *multi_sensor_sim*, mediante el uso de drivers específicos y librerías adaptadas a cada tipo de sensor. Este enfoque permite abstraer la interacción con el hardware y unificar el flujo de datos dentro del sistema ROS 2.

Cada sensor presenta características y métodos de adquisición distintos, lo que ha requerido el uso de estrategias específicas en cada caso.

En el caso de las unidades de medida inercial (IMU), los datos de aceleración y velocidad angular se obtienen a través del driver correspondiente, siendo posteriormente estructurados en mensajes personalizados. Además, se incorporan estimaciones de orientación, ya sea a partir del propio sensor o mediante el modelo cinemático del sistema. Tal y como se observa en la Figura 3-30, los datos se encapsulan antes de su publicación.

```
imu_data = self.imu_driver.get_data(st)

msg1.accel.x = float(a[0])
msg1.gyro.z = float(g[2])
msg1.yaw = math.degrees(st["yaw"])
```

Figura 3-30 Fragmento de la IMU dentro del script *multi_sensor_sim*

En el caso del receptor GNSS, la adquisición se realiza mediante comunicación serie utilizando la librería *pyserial*, procesando directamente las tramas *NMEA* recibidas por el dispositivo. El sistema extrae la información de latitud, longitud y altitud a partir de dichas tramas, realizando la conversión necesaria desde el formato *NMEA* a coordenadas geográficas, tal y como se puede ver a continuación en la Figura 3-31.

```
ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1)
line = ser.readline().decode('ascii', errors='ignore').strip()

if line.startswith('$GPGGA'):
    lat_raw = float(parts[2])
    lon_raw = float(parts[4])
```

Figura 3-31 Fragmento del GNSS dentro del script *multi_sensor_sim*

Este proceso permite obtener datos reales de posicionamiento en tiempo real. En caso de fallo en la comunicación, el sistema implementa un mecanismo de *fallback*, garantizando la continuidad de la ejecución.

Para el sensor LiDAR, la adquisición de datos se realiza mediante un driver específico que proporciona medidas de distancia en función del ángulo. Estos datos son posteriormente transformados en mensajes del tipo *LaserScan*, estándar en ROS 2, permitiendo su uso directo en aplicaciones de navegación y percepción, en la Figura 3-32 se muestra un fragmento a modo ilustrativo de la parte del script del LiDAR.

```
ranges_list, intensities_list = self.lidar_driver.get_scan(...)

scan.ranges = [float(r) for r in ranges_list]
scan.intensities = [float(i) for i in intensities_list]
```

Figura 3-32 Fragmento del LiDAR dentro del script *multi_sensor_sim*

En el caso de la cámara, la adquisición se realiza mediante el uso de *OpenCV* o un driver simulado, generando imágenes en formato RGB que son convertidas a mensajes *sensor_msgs/Image* para su publicación en ROS 2. A modo ilustrativo la Figura 3-33 muestra un fragmento del código asociado.

```
msg = Image()
msg.height = self.img_h
msg.width = self.img_w
msg.encoding = "rgb8"
msg.data = img.tobytes()
```

Figura 3-33 Fragmento de la Cámara USB dentro del script *multi_sensor_sim*

Una vez adquiridos, todos los datos son normalizados y estructurados para garantizar su coherencia dentro del sistema. Este proceso incluye la conversión de unidades, adaptación a formatos estándar de ROS 2 y la asignación de sistemas de referencia mediante cabeceras (*header*).

Este enfoque permite integrar sensores heterogéneos dentro de un flujo de datos unificado, facilitando su posterior procesamiento, visualización y análisis.

A modo de síntesis, la Tabla 3-12 resume los distintos métodos de adquisición empleados para cada sensor, así como las librerías utilizadas y el tipo de datos generados en el sistema.

Sensor	Método de adquisición	Librería	Tipo de dato
IMU	Driver directo	Phidget22	Aceleración, orientación
GNSS	Comunicación serie	pyserial, pynmea2	Latitud, longitud
LiDAR	Driver LiDAR	Propio	Distancias
Cámara	Captura de imagen	OpenCV	Imagen RGB

Tabla 3-12 Resumen de los métodos de adquisición de datos multisensor, incluyendo las librerías empleadas y el tipo de información generada por cada sensor

3.5.4 Publicación en ROS 2

Una vez adquiridos y procesados los datos procedentes de los distintos sensores, estos son publicados en el ecosistema *ROS 2* mediante el uso de *publishers*, permitiendo su distribución a otros nodos del sistema y a herramientas externas de visualización.

El nodo *multi_sensor_sim* implementa múltiples *publishers*, cada uno asociado a un tipo de sensor y a un tópico específico. Esta estructura permite desacoplar completamente la adquisición de datos de su consumo, facilitando la escalabilidad del sistema y la integración con otras aplicaciones.

Tal y como se observa en la Figura 3-34, cada *publisher* se inicializa indicando el tipo de mensaje y el tópico correspondiente

```

self.pub_imu1 = self.create_publisher(ImuDetails, "/sim/imu1", 10)
self.pub_gps = self.create_publisher(NavSatFix, "/sim/gps", 10)
self.pub_scan = self.create_publisher(LaserScan, "/sim/lidar/scan", 10)
self.pub_img = self.create_publisher(Image, "/sim/camera/image_raw", 10)

```

Figura 3-34 Fragmento de los distintos *publishers* de cada sensor dentro del script *multi_sensor_sim*

Los datos generados por cada sensor se encapsulan en mensajes *ROS 2* estándar (o personalizados en el caso de la IMU) y se publican periódicamente en los tópicos correspondientes. Este enfoque garantiza la interoperabilidad del sistema, permitiendo que otros nodos o herramientas puedan suscribirse a los datos sin necesidad de conocer los detalles internos de la adquisición.

Además, el uso de mensajes estándar como *sensor_msgs/Image*, *sensor_msgs/LaserScan* o *sensor_msgs/NavSatFix* facilita la integración con el ecosistema *ROS 2*, incluyendo herramientas de visualización, simulación y navegación.

A continuación, la Tabla 3-13 recoge los principales tópicos generados por el sistema, junto con el tipo de mensaje asociado, la frecuencia de publicación y una breve descripción de su contenido.

Tópico	Tipo de mensaje	Frecuencia aprox.	Descripción
/sim/imu1	ImuDetails	~50 Hz	Datos inerciales de la IMU principal
/sim/imu2	ImuDetails	~50 Hz	Datos de la segunda IMU
/sim/gps	NavSatFix	~1 Hz	Posicionamiento GNSS (latitud, longitud, altitud)
/sim/lidar/scan	LaserScan	~5–10 Hz	Distancias medidas por el LiDAR
/sim/camera/image_raw	Image	~30 Hz	Imágenes capturadas por la cámara
/sim/odom	Odometry	variable	Estimación del estado del sistema

Tabla 3-13 Relación de tópicos *ROS 2* generados por el nodo multisensor, incluyendo tipo de mensaje, frecuencia de publicación y descripción funcional

3.5.5 Sistema de referencia y transformaciones

En sistemas robóticos multisensor, la correcta gestión de los sistemas de referencia resulta fundamental para interpretar de forma coherente la información procedente de los distintos sensores. Cada dispositivo mide magnitudes en su propio sistema de coordenadas, por lo que es necesario establecer relaciones espaciales entre ellos.

Para este propósito, se ha utilizado la librería *tf2_ros*, que permite gestionar transformaciones entre distintos marcos de referencia (*frames*) dentro del ecosistema *ROS 2*.

En el sistema desarrollado, se define una estructura jerárquica de referencias centrada en el vehículo, donde el marco principal (*base_link*) representa el sistema de coordenadas del robot. A partir de este, se definen los marcos asociados a cada sensor, permitiendo expresar todas las medidas en un sistema común.

Concretamente, se establece:

- Un marco global (*world*), que actúa como referencia fija del entorno
- Un marco del vehículo (*base_link*), que representa la posición y orientación del robot
- Marcos específicos para cada sensor (*imu_link*, *camera_link*, *lidar_link*, etc.)

La transformación entre el sistema global y el vehículo se actualiza dinámicamente en función del estado estimado del sistema, mientras que las transformaciones entre el vehículo y los sensores se consideran estáticas, ya que dependen de la configuración física del montaje.

Un ejemplo de publicación de transformaciones dinámicas se muestra en la

```
t = TransformStamped()
t.header.frame_id = "world"
t.child_frame_id = "base_link"

t.transform.translation.x = self.state["x"]
t.transform.translation.y = self.state["y"]

self.tf_broadcaster.sendTransform(t)
```

Figura 3-35 Fragmento de la publicación de transformaciones dentro del script *multi_sensor_sim*

Este mecanismo permite que cualquier nodo del sistema pueda conocer en todo momento la posición relativa entre los distintos elementos, facilitando tareas como la fusión sensorial, la visualización o la navegación.

La Figura 3-36 representa la estructura de transformaciones implementada en el sistema, mostrando la relación jerárquica entre los distintos marcos de referencia.

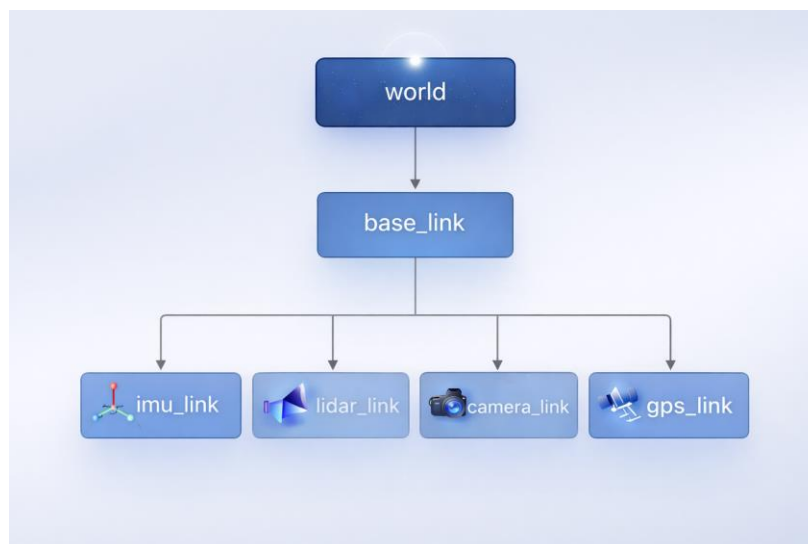


Figura 3-36 Árbol de transformaciones del sistema basado en tf2, donde se definen los marcos de referencia del vehículo y de los sensores. Fuente: Elaboración asistida por herramientas de inteligencia artificial

La correcta definición de estos sistemas de referencia resulta esencial en aplicaciones de percepción y navegación, ya que permite integrar datos procedentes de sensores heterogéneos en un mismo marco espacial. Sin este mecanismo, la información generada por cada sensor sería inconsistente y difícilmente combinable.

En este proyecto no se ha implementado una calibración extrínseca precisa entre sensores, asumiendo posiciones relativas aproximadas en función del montaje físico, lo cual es suficiente para los objetivos planteados.

3.5.6 Gestión de frecuencias y sincronización

Uno de los aspectos más relevantes en sistemas multisensor es la correcta gestión de las frecuencias de adquisición, ya que cada sensor presenta características temporales distintas en función de su tecnología y propósito.

En el sistema desarrollado, se ha optado por una arquitectura basada en temporizadores independientes dentro del nodo ROS 2, lo que permite desacoplar la frecuencia de actualización de cada sensor. Este enfoque se implementa mediante la creación de múltiples temporizadores asociados a distintas tareas, tal y como se ha descrito previamente.

Cada sensor opera a una frecuencia específica:

- Las IMU proporcionan datos a alta frecuencia (del orden de decenas de Hz), necesarios para capturar la dinámica del vehículo
- El GNSS trabaja a baja frecuencia (≈ 1 Hz), debido a las limitaciones del sistema de posicionamiento
- El LiDAR opera a frecuencia media ($\approx 5\text{--}10$ Hz), en función del modelo y configuración
- La cámara proporciona imágenes a mayor frecuencia (≈ 30 Hz), condicionada por la capacidad de procesamiento

Esta heterogeneidad hace inviable una sincronización estricta sin introducir una complejidad adicional significativa. Por este motivo, se ha adoptado un modelo de ejecución asíncrono, donde cada sensor publica datos de forma independiente.

Este enfoque presenta las siguientes ventajas:

- Permite adaptar el sistema a sensores con distintas características temporales
- Reduce la probabilidad de bloqueos en la ejecución
- Mejora la eficiencia en sistemas con recursos limitados

No obstante, también introduce una limitación importante, que es la falta de sincronización temporal entre los datos. Esto puede provocar pequeños desalineamientos entre las medidas de distintos sensores, especialmente en aplicaciones de fusión sensorial avanzada.

Sin embargo, para el alcance del presente proyecto, esta aproximación resulta adecuada, ya que permite simplificar la implementación sin comprometer significativamente la validez de los resultados.

A modo de resumen, la Tabla 3-14 recoge las frecuencias de operación aproximadas de los distintos sensores integrados en el sistema.

Sensor	Frecuencia aproximada	Tipo de dato	Observaciones
IMU	~ 50 Hz	Aceleración, velocidad angular	Alta frecuencia para dinámica
GNSS	~ 1 Hz	Posición geográfica	Limitado por hardware
LiDAR	$\sim 5\text{--}10$ Hz	Distancias	Dependiente del barrido

Sensor	Frecuencia aproximada	Tipo de dato	Observaciones
Cámara	~30 Hz	Imagen RGB	Limitado por procesamiento

Tabla 3-14 Frecuencias de adquisición de los sensores del sistema, mostrando la heterogeneidad temporal característica de arquitecturas multisensor

3.5.7 Diseño modular del sistema

El diseño software del sistema se ha desarrollado siguiendo una arquitectura modular, en la que cada componente funcional se encuentra desacoplado del resto, permitiendo una mayor flexibilidad, mantenibilidad y escalabilidad.

En este enfoque, la lógica específica de cada sensor se encapsula en módulos independientes o drivers, encargados de gestionar la comunicación con el hardware y proporcionar los datos en un formato estructurado. Estos módulos son posteriormente integrados en el nodo principal, que actúa como elemento coordinador del sistema.

El nodo *multi_sensor_sim* centraliza la adquisición y publicación de datos, pero mantiene separada la lógica de cada sensor, lo que permite modificar o sustituir componentes sin afectar al resto del sistema. Esta separación de responsabilidades resulta clave en el desarrollo de sistemas complejos, ya que reduce la dependencia entre módulos y facilita la detección de errores.

Además, el uso de ROS 2 como *middleware* refuerza esta modularidad, ya que la comunicación entre componentes se realiza mediante tópicos, eliminando la necesidad de interacciones directas entre módulos. De este modo, cualquier nodo externo puede suscribirse a los datos generados sin necesidad de conocer la implementación interna del sistema.

Desde el punto de vista del diseño, la arquitectura puede interpretarse como una estructura en cuatro capas:

- Capa de adquisición: drivers y librerías específicas de cada sensor
- Capa de procesamiento: normalización y estructuración de datos
- Capa de comunicación: publicación mediante tópicos ROS 2
- Capa de visualización: herramientas externas como Foxglove

Esta organización permite una clara separación de responsabilidades, facilitando tanto el desarrollo como la futura ampliación del sistema.

La Figura 3-37 muestra la arquitectura modular del sistema software, destacando la separación en capas y el flujo de información entre los distintos componentes.



Figura 3-37 Arquitectura modular del sistema multisensor, organizada en capas desde la adquisición de datos hasta su visualización. Fuente: Elaboración asistida por herramientas de inteligencia artificial

Como limitación, el uso de un nodo centralizado puede convertirse en un cuello de botella en sistemas de mayor complejidad, donde sería más adecuado adoptar una arquitectura distribuida basada en múltiples nodos independientes. No obstante, para el alcance del presente proyecto, esta solución resulta adecuada y permite simplificar significativamente la implementación.

3.5.8 Integración con Foxglove

Con el objetivo de visualizar y analizar en tiempo real los datos generados por el sistema multisensor, se ha integrado la herramienta Foxglove, una plataforma especializada en la monitorización de sistemas robóticos basada en ROS.

La comunicación entre la Raspberry Pi y el entorno de visualización se realiza mediante el paquete Foxglove Bridge, que permite exponer los tópicos de ROS 2 a través de un servidor *WebSocket*. Este enfoque posibilita el acceso remoto a los datos del sistema sin necesidad de ejecutar herramientas gráficas en el sistema embarcado, optimizando así el uso de recursos.

El bridge se ejecuta en la Raspberry Pi, generando un servidor accesible mediante una dirección del tipo:

- `ws://<IP_RASPBERRY>:8765`

En el caso del sistema desarrollado, la conexión se estableció utilizando la dirección IP estática configurada previamente, lo que garantiza la estabilidad del acceso entre sesiones.

Una vez iniciado el bridge, *Foxglove* permite conectarse al sistema mediante el protocolo *WebSocket*, facilitando la suscripción a los distintos tópicos publicados en ROS 2.

La Figura 3-38 muestra el proceso de conexión al sistema mediante el protocolo *WebSocket* en Foxglove, utilizando la dirección IP de la Raspberry Pi.

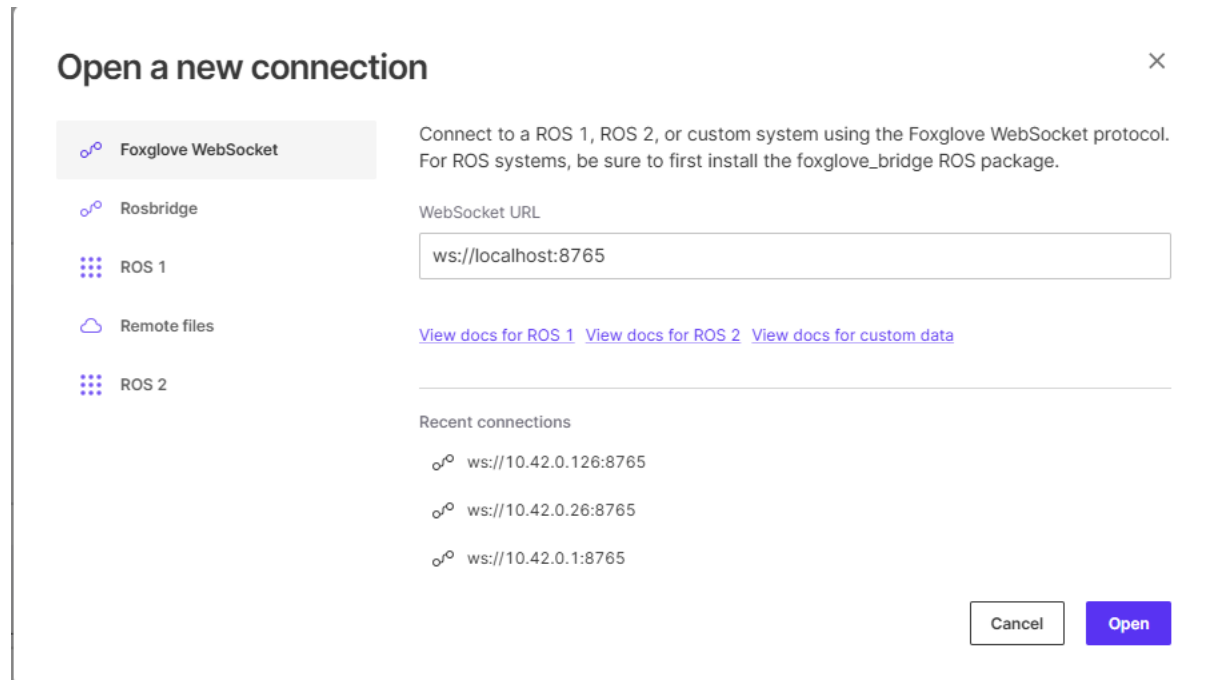


Figura 3-38 Configuración de la conexión WebSocket en Foxglove para el acceso remoto a los tópicos ROS 2 del sistema embarcado

Una vez establecida la conexión, la herramienta permite visualizar en tiempo real los datos publicados por el sistema, organizándolos en distintos paneles configurables. Entre las principales funcionalidades utilizadas destacan:

- Visualización de imágenes procedentes de la cámara
- Representación de datos LiDAR en forma de nube de puntos
- Monitorización de variables de la IMU mediante gráficos temporales
- Visualización de la posición GNSS sobre mapas

Esta capacidad de visualización resulta fundamental para la validación del sistema, ya que permite comprobar el correcto funcionamiento de cada sensor y analizar el comportamiento global del sistema en tiempo real.

La Figura 3-39 muestra la interfaz de Foxglove durante la ejecución del sistema, donde se visualizan simultáneamente los datos procedentes de los distintos sensores.

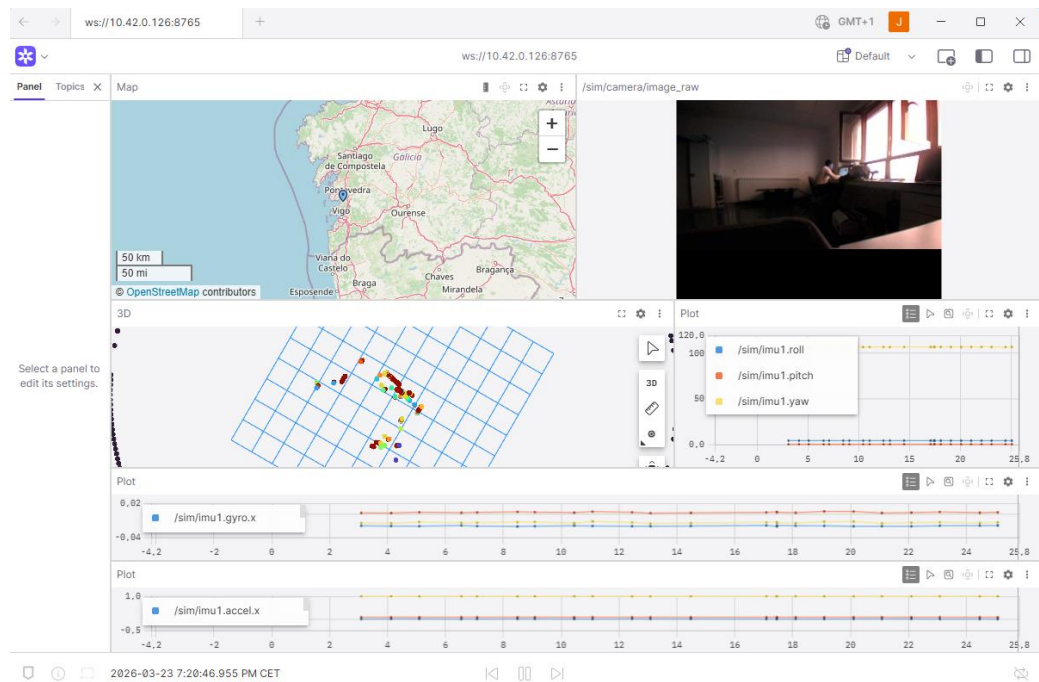


Figura 3-39 Visualización en tiempo real de los datos multisensor en Foxglove, incluyendo información de cámara, LiDAR, GNSS e IMU

La integración con Foxglove permite desacoplar completamente la adquisición de datos de su visualización, facilitando tareas de depuración, análisis y validación sin sobrecargar el sistema embarcado. Este enfoque resulta especialmente adecuado en plataformas con recursos limitados, como la Raspberry Pi, donde la ejecución de interfaces gráficas puede comprometer el rendimiento del sistema.

3.5.9 Síntesis del desarrollo software

Con el objetivo de sintetizar el proceso completo de desarrollo software del sistema multisensor, en este apartado se presenta un resumen estructurado dentro de la Tabla 3-15 de las principales fases del proyecto, incluyendo los objetivos de cada etapa, las herramientas empleadas y los resultados obtenidos.

Fase	Objetivo	Herramientas empleadas	Resultado obtenido
Preparación del sistema operativo	Instalar y configurar el entorno base para la ejecución del sistema	Raspberry Pi Imager, Raspberry Pi OS, terminal Linux	Sistema operativo funcional y preparado para desarrollo
Configuración de red y acceso remoto	Permitir la conexión remota y la comunicación con el sistema	Netplan, SSH, red WiFi	Acceso remoto estable y comunicación con estación base
Integración de sensores	Conectar y verificar el funcionamiento individual de cada sensor	Drivers específicos (<i>Phidget</i> , GNSS, cámara, LiDAR), Python	Sensores operativos y adquisición de datos individual
Instalación de ROS 2	Implementar un entorno de comunicación y gestión de nodos	ROS 2 Jazzy, paquetes ROS	Sistema preparado para arquitectura distribuida
Desarrollo del nodo multisensor	Integrar la adquisición de todos los sensores en un único nodo	Python, <i>rclpy</i> , librerías específicas	Nodo funcional con publicación de datos multisensor

Fase	Objetivo	Herramientas empleadas	Resultado obtenido
Visualización con <i>Foxglove</i>	Permitir la monitorización en tiempo real de los datos	<i>Foxglove Bridge</i> , <i>WebSocket</i>	Visualización remota de datos en tiempo real
Validación del sistema	Evaluar el funcionamiento global del sistema integrado	Pruebas experimentales, <i>Foxglove</i> , análisis de datos	Sistema validado en entorno real y detección de limitaciones

Tabla 3-15 Síntesis del proceso de desarrollo software del sistema multisensor, incluyendo fases, objetivos, herramientas empleadas y resultados obtenidos

Como se observa, el desarrollo del sistema ha seguido una secuencia estructurada que abarca desde la configuración inicial del entorno hasta la validación funcional del sistema multisensor, garantizando la correcta integración de los componentes hardware y software. Esta organización ha permitido abordar el desarrollo de forma progresiva, facilitando la detección de errores y la validación incremental del sistema.

3.5.10 Ventajas de la implementación adoptada

La implementación software desarrollada presenta diversas ventajas desde el punto de vista técnico y funcional, especialmente en el contexto de sistemas multisensor embarcados.

En primer lugar, permite la integración simultánea de múltiples sensores heterogéneos dentro de una misma plataforma, gestionando datos de distinta naturaleza (inerciales, visuales, geométricos y de posicionamiento) de forma unificada. Esta capacidad resulta fundamental en sistemas de percepción avanzados, donde la complementariedad entre sensores mejora la calidad de la información disponible.

En segundo lugar, el uso de *ROS 2* como middleware proporciona una arquitectura robusta, modular y escalable, basada en comunicación mediante tópicos. Este enfoque facilita la interoperabilidad con otras herramientas del ecosistema *ROS*, así como la posible ampliación del sistema mediante la incorporación de nuevos nodos o sensores sin necesidad de modificar la estructura existente.

Otra ventaja relevante es la separación entre adquisición de datos y visualización, implementada mediante *Foxglove Bridge*. Este diseño permite descargar a la Raspberry Pi de tareas gráficas, delegando la visualización en un equipo externo con mayor capacidad, lo que mejora el rendimiento global del sistema.

Asimismo, la utilización de un nodo centralizado ha permitido simplificar la implementación y acelerar la fase de desarrollo, facilitando la integración progresiva de sensores y la validación del sistema en etapas tempranas del proyecto.

Por último, la arquitectura implementada permite operar tanto con sensores reales como con datos simulados, lo que aporta una gran flexibilidad durante el desarrollo y facilita la depuración del sistema.

3.5.11 Consideraciones finales

La implementación software desarrollada ha permitido validar la viabilidad de un sistema multisensor embarcado basado en *ROS 2*, capaz de integrar de forma simultánea diferentes fuentes de información heterogéneas.

A lo largo del desarrollo, se ha comprobado que la utilización de una arquitectura modular, junto con un nodo centralizado, facilita la gestión de la adquisición de datos y simplifica la integración progresiva de sensores. Este enfoque ha permitido validar el sistema de forma incremental, reduciendo la complejidad en las fases iniciales del proyecto.

Uno de los aspectos más relevantes identificados es la importancia de la gestión temporal en sistemas multisensor. La utilización de temporizadores independientes ha permitido adaptar la frecuencia de

adquisición a las características de cada dispositivo, garantizando un funcionamiento estable incluso en una plataforma con recursos limitados como la Raspberry Pi.

Asimismo, la separación entre adquisición de datos y visualización, mediante el uso de *Foxglove Bridge*, ha demostrado ser una solución eficaz para optimizar el rendimiento del sistema embarcado. Este enfoque permite mantener la eficiencia del sistema sin renunciar a capacidades avanzadas de monitorización en tiempo real.

Desde el punto de vista práctico, el desarrollo ha puesto de manifiesto la importancia de una correcta configuración del entorno software, así como la necesidad de gestionar adecuadamente aspectos como la conectividad, la comunicación entre dispositivos y la reutilización de dependencias.

No obstante, el sistema presenta ciertas limitaciones inherentes al diseño adoptado. En particular, el uso de un nodo centralizado puede suponer un cuello de botella en sistemas de mayor complejidad, y la ausencia de sincronización temporal estricta puede afectar a aplicaciones de fusión sensorial avanzada.

Como líneas de mejora futuras, se plantea la evolución hacia una arquitectura distribuida basada en múltiples nodos, así como la implementación de mecanismos de sincronización temporal y técnicas de fusión sensorial más avanzadas.

En conjunto, la arquitectura desarrollada constituye una base sólida y escalable para la implementación de sistemas de percepción más complejos, cumpliendo con los objetivos planteados en el presente trabajo.

En el siguiente capítulo se presenta la validación experimental del sistema, donde se analiza su comportamiento en condiciones reales de operación.

4 RESULTADOS Y VALIDACIÓN EXPERIMENTAL

4.1 Entorno Experimental

Las pruebas del sistema se llevaron a cabo en un entorno real utilizando una plataforma móvil sobre la que se integró un sistema multisensor embarcado, previamente descrito en el Capítulo 3: Desarrollo del TFG. La unidad de procesamiento empleada fue una Raspberry Pi 4, seleccionada por su bajo consumo energético, tamaño reducido y capacidad suficiente para la ejecución de ROS 2 en aplicaciones de robótica ligera.

Todos los componentes fueron montados sobre un vehículo móvil, configurando un sistema autónomo alimentado mediante una batería externa (*powerbank*), lo que permitió la realización de pruebas sin necesidad de conexión a red eléctrica.

Desde el punto de vista software, el sistema se ejecuta sobre *Ubuntu Server 24.04*, utilizando *ROS 2 Jazzy* como *middleware* de comunicación. La visualización de los datos se realizó mediante la herramienta *Foxglove*, ejecutada en un equipo externo.

La comunicación entre la Raspberry Pi y el equipo de visualización se estableció mediante una red local inalámbrica generada a través de un punto de acceso (*hotspot*). Ambos dispositivos se conectaron a dicha red, permitiendo la transmisión de datos en tiempo real mediante protocolo *WebSocket*.

Con el objetivo de garantizar la estabilidad de la conexión, se configuró una dirección IP estática en la Raspberry Pi, lo que permitió mantener constante la dirección de acceso al sistema durante las distintas sesiones de prueba.

Las pruebas se realizaron en condiciones reales de operación, lo que permitió evaluar el comportamiento del sistema teniendo en cuenta factores como la latencia de la comunicación inalámbrica, el consumo energético del sistema y las limitaciones propias del hardware embarcado.

La Figura 4-1 muestra el sistema multisensor embarcado durante la fase de validación, donde se observa la disposición de los sensores sobre el vehículo.



Figura 4-1 Plataforma experimental utilizada durante las pruebas, incluyendo la integración de los sensores y la unidad de procesamiento embarcada

4.2 Metodología de pruebas

La validación del sistema se llevó a cabo mediante pruebas experimentales en un entorno real, utilizando el vehículo móvil sobre el cual se integraron los distintos sensores. El objetivo principal fue verificar el funcionamiento del sistema multisensor en condiciones operativas, evaluando la adquisición, transmisión y visualización de los datos en tiempo real.

El procedimiento de puesta en marcha del sistema se realizó inicialmente mediante conexión directa a un monitor y teclado. Tras alimentar la Raspberry Pi mediante una batería externa (*powerbank*) instalada en el chasis del vehículo, se procedió al arranque del sistema operativo.

Una vez iniciado el sistema, se abrieron dos terminales independientes para la ejecución de los procesos principales:

Ejecución del nodo multisensor:

- `ros2 run sim_sensors multi_sensor_sim`

Ejecución del servicio de comunicación:

- `ros2 launch foxglove_bridge foxglove_bridge_launch.xml
port:=8765`

Tras la inicialización del sistema, la Raspberry Pi comienza a adquirir datos de los sensores y a publicarlos en los correspondientes tópicos de *ROS 2*, los cuales son accesibles externamente mediante el servidor *WebSocket*.

Una vez verificado el correcto funcionamiento del sistema, se procedió a desconectar el monitor y el teclado, permitiendo que la plataforma operase de forma autónoma. En esta configuración, el sistema continúa ejecutándose de manera independiente, transmitiendo los datos en tiempo real sin necesidad de interacción adicional.

Cabe destacar que este procedimiento puede simplificarse mediante el uso de acceso remoto por SSH, evitando la necesidad de conexión física a monitor y teclado. Gracias a la configuración de una dirección IP estática en la Raspberry Pi, es posible establecer la conexión directamente desde el equipo de desarrollo, facilitando el despliegue del sistema.

La visualización de los datos se realizó mediante un ordenador portátil conectado a la misma red local, utilizando la herramienta Foxglove. La conexión se estableció mediante una dirección WebSocket del tipo:

- `ws://10.42.0.126:8765`

Durante las pruebas, se configuró un panel de visualización que permite monitorizar de forma simultánea los distintos sensores, incluyendo:

- Posición GNSS sobre mapa
- Representación del LiDAR
- Flujo de vídeo de la cámara
- Variables inerciales de la IMU

Este entorno permitió evaluar el comportamiento global del sistema en tiempo real, facilitando la detección de posibles anomalías y la validación de su funcionamiento.

La Figura 4-2 muestra el entorno de visualización empleado durante las pruebas, donde se representan simultáneamente los datos procedentes de los distintos sensores.

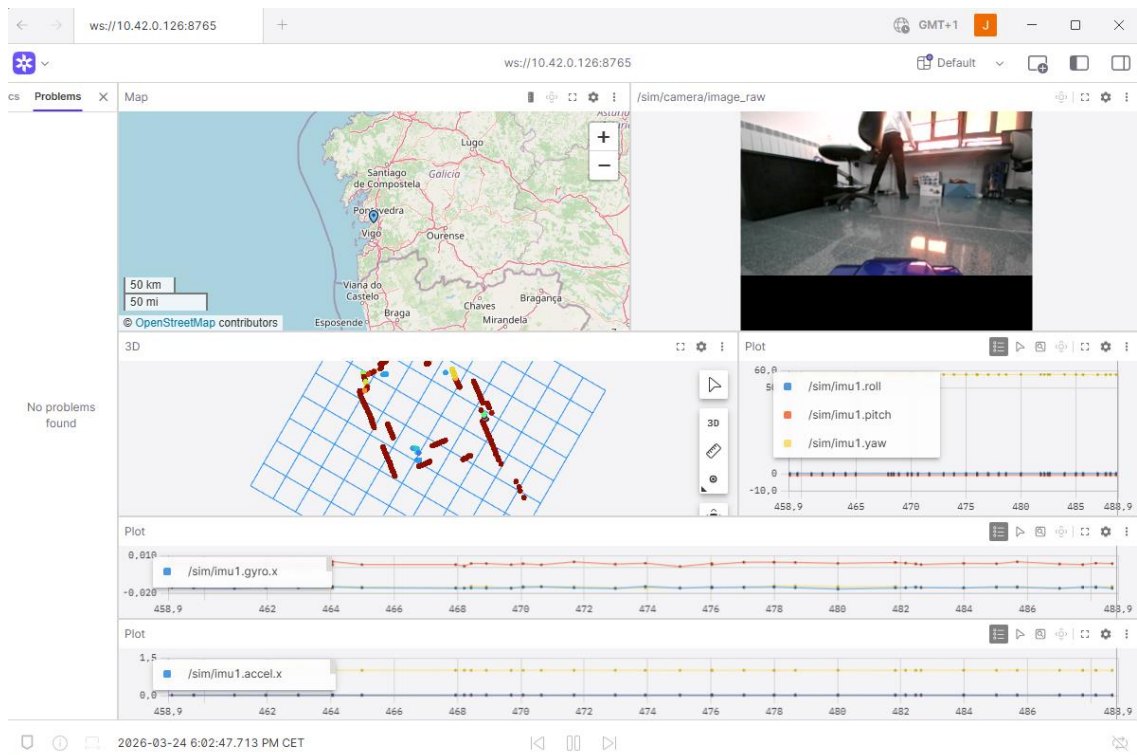


Figura 4-2 Interfaz de Foxglove durante la fase de validación, mostrando datos multisensor en tiempo real

4.3 Validación de sensores

4.3.1 Validación de la IMU

La validación de la unidad de medida inercial (IMU) se llevó a cabo mediante el análisis de los datos publicados en los tópicos correspondientes y su visualización en tiempo real a través de la herramienta Foxglove.

La IMU proporciona información sobre la aceleración lineal, la velocidad angular y la orientación del sistema, lo que permite caracterizar el comportamiento dinámico del vehículo. Para su validación,

se analizaron principalmente los ángulos de orientación (roll, pitch y yaw), así como las componentes de aceleración en los tres ejes.

Previamente a su integración en el sistema, el sensor fue calibrado utilizando la herramienta oficial del fabricante (*Phidget Control Panel*), lo que permitió compensar posibles errores sistemáticos asociados a los acelerómetros y giróscopos. Este proceso resulta fundamental para garantizar la fiabilidad de las medidas obtenidas [17].

Durante las pruebas, se analizaron los datos en condiciones tanto estáticas como dinámicas. En estado de reposo, se observó que las señales presentan un comportamiento estable, manteniéndose dentro de rangos coherentes con la ausencia de movimiento. En particular:

- Las aceleraciones en los ejes horizontales se mantienen próximas a cero.
- El eje vertical presenta un valor cercano a la aceleración de la gravedad ($\sim 9.81 \text{ m/s}^2$).
- Las velocidades angulares muestran valores reducidos con pequeñas fluctuaciones.
- Estas pequeñas variaciones, estimadas dentro de un rango bajo propio del ruido del sensor, son coherentes con el comportamiento esperado en dispositivos inerciales de este tipo.

En cuanto a la frecuencia de actualización, la IMU presenta una tasa elevada (del orden de decenas de Hz), lo que permite una representación prácticamente continua del estado del sistema sin retardos apreciables en la visualización.

Por otro lado, al introducir cambios en la orientación del sistema, se observan variaciones claras y coherentes en los ángulos (roll, pitch y yaw), lo que confirma la correcta sensibilidad del sensor y su capacidad para detectar cambios de orientación.

En conjunto, los resultados obtenidos permiten confirmar que la IMU se encuentra correctamente calibrada e integrada en el sistema, proporcionando datos consistentes y adecuados para su uso en aplicaciones de navegación y estimación del estado.

La Figura 4-3 muestra la evolución temporal de los datos de orientación y aceleración registrados por la IMU durante las pruebas experimentales.

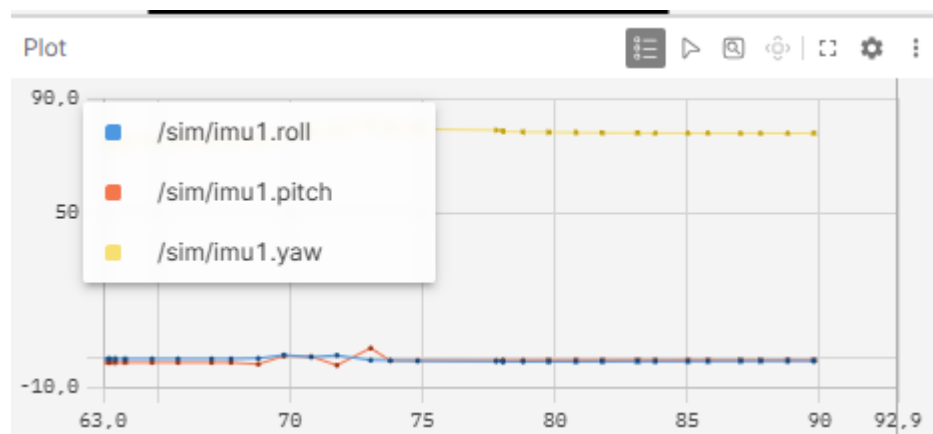


Figura 4-3 Representación temporal de los datos de la IMU en Foxglove, incluyendo roll, pitch y yaw, donde se observa un comportamiento estable en condiciones estáticas y una respuesta coherente ante cambios de orientación.

En conjunto, los resultados obtenidos permiten confirmar que la IMU se encuentra correctamente calibrada e integrada en el sistema, proporcionando datos consistentes y adecuados para su uso en aplicaciones de navegación y estimación del estado. No obstante, se observa la presencia de ruido en las señales, lo cual es característico de este tipo de sensores y debe ser tenido en cuenta en aplicaciones de fusión sensorial avanzada.

4.3.2 Validación del GNSS

La validación del sistema de posicionamiento GNSS se realizó mediante el análisis de los datos de latitud, longitud y altitud publicados en *ROS 2* y su visualización en tiempo real a través de la herramienta Foxglove.

El receptor GNSS proporciona información de posicionamiento geográfico en coordenadas globales, lo que permite situar el sistema sobre un mapa y realizar un seguimiento de su posición en tiempo real. Para la validación, se empleó el panel de visualización geográfica de *Foxglove*, donde los datos se representan directamente sobre un mapa.

Durante las pruebas en condiciones estáticas, se observó que la posición estimada no permanece completamente fija, sino que presenta pequeñas variaciones alrededor de un punto central. Estas variaciones corresponden a un error de posicionamiento estimado en el rango de varios metros (aproximadamente entre 2 y 5 m), valor característico de receptores GNSS de consumo en condiciones de cielo abierto.

En condiciones dinámicas, se comprobó que el sistema es capaz de actualizar la posición de forma continua, con una frecuencia aproximada de 1 Hz, permitiendo el seguimiento del vehículo en tiempo casi real. El retardo observado en la actualización de la posición es bajo (del orden de pocos segundos), estando principalmente condicionado por la comunicación inalámbrica.

La Figura 4-4 muestra la representación de la posición GNSS del sistema sobre el mapa durante la fase de validación.

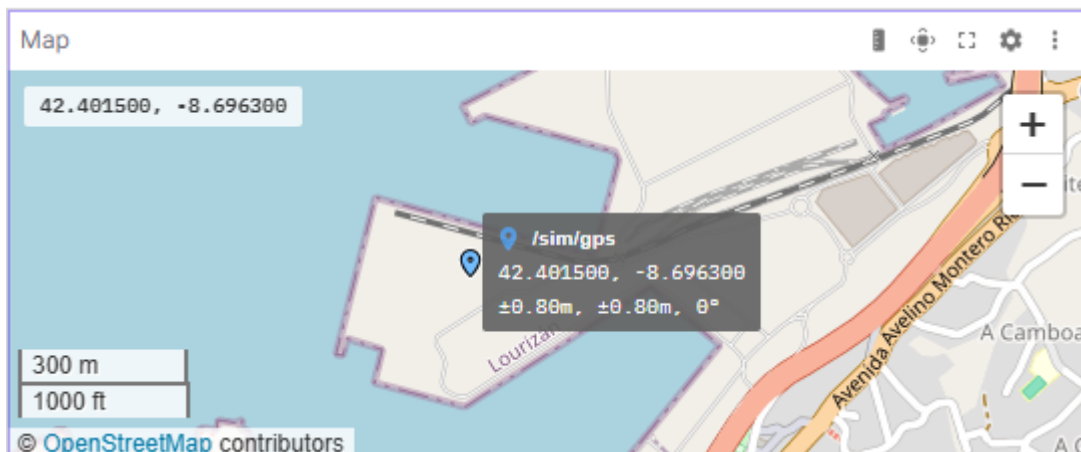


Figura 4-4 Visualización de la posición GNSS en Foxglove, donde se observa la estimación de la ubicación del sistema y las pequeñas variaciones asociadas al error de posicionamiento.

En condiciones dinámicas, se comprobó que el sistema es capaz de actualizar la posición de forma continua, permitiendo el seguimiento del vehículo en tiempo real. La evolución de la posición es coherente con el movimiento del sistema, lo que confirma la correcta adquisición y procesamiento de los datos GNSS.

Aunque en el presente proyecto no se ha realizado una evaluación cuantitativa del error de posicionamiento, los resultados obtenidos son consistentes con el comportamiento esperado de un receptor GNSS de estas características.

La integración del GNSS dentro del sistema *ROS 2* permite su combinación con otros sensores, como la IMU o el LiDAR, lo que resulta fundamental en aplicaciones de localización y navegación. En conjunto, los resultados obtenidos permiten concluir que el sistema GNSS se encuentra correctamente integrado, proporcionando datos válidos y adecuados para su uso dentro del sistema multisensor.

4.3.3 Validación de la Cámara

La validación del sistema de visión se realizó mediante el análisis del flujo de imágenes publicado en el tópico correspondiente y su visualización en tiempo real a través de la herramienta *Foxglove*.

La cámara USB integrada en el sistema permite la captura de imágenes en formato RGB, las cuales son procesadas y publicadas en *ROS 2* para su posterior transmisión al equipo externo. En *Foxglove*, estas imágenes se representan mediante un panel específico que muestra el flujo de vídeo en tiempo real.

La Figura 4-5 muestra el flujo de imágenes capturado por la cámara y visualizado en tiempo real mediante *Foxglove*.

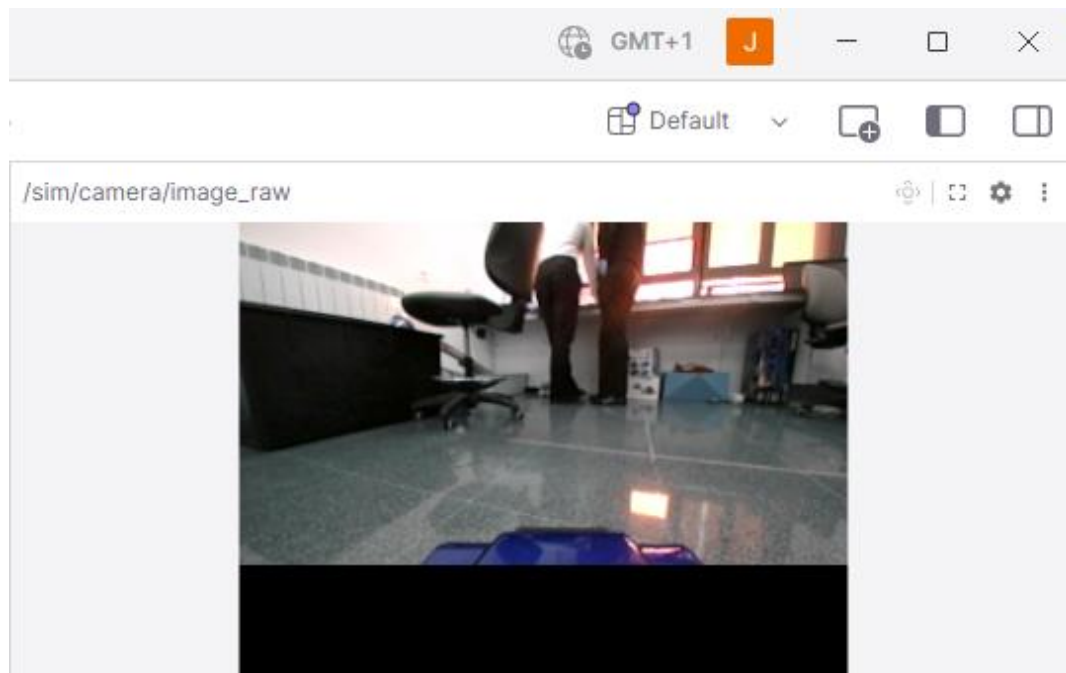


Figura 4-5 Visualización del flujo de imágenes procedente de la cámara USB en Foxglove, donde se observa la captura del entorno en tiempo real

Durante las pruebas, se comprobó que el sistema es capaz de capturar y transmitir imágenes, si bien se identificaron limitaciones significativas en la continuidad y latencia del flujo de vídeo.

En concreto, se observaron retardos elevados en la visualización, con valores estimados entre 20 y 30 segundos entre la captura y la representación de los fotogramas. Asimismo, el flujo de vídeo no se presenta de forma continua, sino mediante la recepción de imágenes discretas con interrupciones frecuentes, lo que da lugar a una visualización poco fluida. Este comportamiento se atribuye principalmente a varios factores:

- La limitada capacidad de procesamiento de la Raspberry Pi 4
- El elevado volumen de datos generado por la cámara
- El uso de comunicación inalámbrica mediante WiFi

A diferencia de otros sensores, la cámara genera una gran cantidad de información por segundo, lo que introduce un cuello de botella tanto en el procesamiento como en la transmisión de datos. No obstante, a pesar de estas limitaciones, el sistema mantiene la capacidad de capturar y transmitir información visual, lo que permite su uso en tareas de monitorización no crítica en tiempo real.

En conjunto, los resultados obtenidos permiten concluir que la cámara se encuentra correctamente integrada en el sistema, si bien presenta limitaciones importantes en términos de latencia y continuidad del flujo de vídeo. Estas limitaciones podrían mitigarse mediante el uso de técnicas de compresión de imagen, reducción de resolución o hardware de mayor capacidad.

4.3.4 Validación del LiDAR

La validación del sensor LiDAR se realizó mediante el análisis de los datos de distancia publicados en el tópic correspondiente y su representación en tiempo real a través de la herramienta *Foxglove*.

El LiDAR proporciona información sobre la distancia a los objetos del entorno mediante la emisión de haces láser en diferentes direcciones, generando un conjunto de medidas angulares que permiten reconstruir la geometría del entorno. Estos datos son publicados en *ROS 2* en forma de mensajes del tipo *LaserScan*, facilitando su integración con herramientas de visualización y procesamiento.

En *Foxglove*, los datos del LiDAR se representan mediante un panel tridimensional, donde es posible observar la nube de puntos generada por el sensor. La Figura 4-6 muestra la representación de los datos del LiDAR en *Foxglove*, donde se visualiza la nube de puntos generada a partir de las mediciones de distancia.

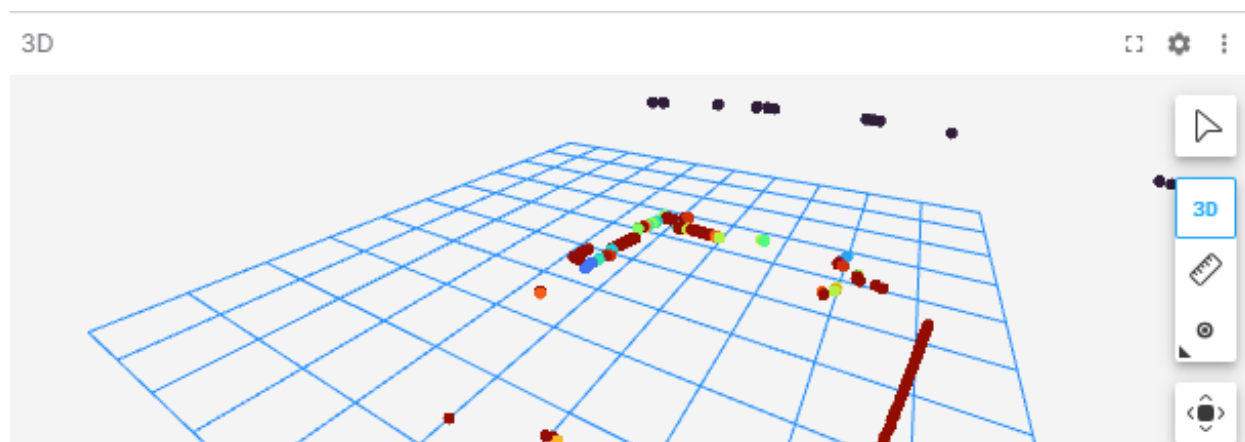


Figura 4-6 Visualización de los datos del sensor LiDAR en Foxglove, mostrando la representación del entorno mediante una nube de puntos en tiempo real

Durante las pruebas, se comprobó que el LiDAR es capaz de detectar objetos del entorno y representar su posición relativa con respecto al sensor de forma consistente.

La frecuencia de actualización del sensor se sitúa en torno a varios Hz (≈ 10 Hz), lo que permite una representación dinámica del entorno prácticamente en tiempo real. El retardo observado es bajo, generalmente inferior a unos pocos segundos, estando condicionado principalmente por la comunicación inalámbrica.

En condiciones estáticas, la nube de puntos se mantiene estable, reflejando de forma coherente la geometría del entorno. Por otro lado, al introducir movimiento en el sistema, se observa una actualización continua de los puntos, permitiendo la reconstrucción dinámica del entorno.

La densidad de puntos y la frecuencia de actualización resultan suficientes para aplicaciones básicas de percepción y detección de obstáculos.

En conjunto, los resultados obtenidos permiten concluir que el sensor LiDAR se encuentra correctamente integrado en el sistema, proporcionando información fiable y coherente sobre el entorno.

Con el objetivo de sintetizar el proceso de validación realizado para cada sensor, en la Tabla 4-1 se recogen las principales métricas observadas, el comportamiento esperado y las incidencias detectadas durante las pruebas.

Sensor	Métrica observada	Comportamiento esperado	Incidencia detectada	Conclusión
IMU	Señales estables en reposo, alta frecuencia	Alta estabilidad y respuesta rápida	Ruido leve en señales	Funcionamiento correcto
GNSS	Error 2–5 m, actualización ~1 Hz	Posicionamiento global continuo	Variación en posición en estático	Precisión acorde a GNSS de consumo
Cámara RGB	Flujo no continuo, latencia 20–30 s	Vídeo fluido en tiempo real	Retardos elevados y parones	Limitada para tiempo real
LiDAR	~10 Hz, nube de puntos estable	Representación continua del entorno	Ligera pérdida de fluidez ocasional	Funcionamiento adecuado

Tabla 4-1 Resumen del proceso de validación de los sensores, incluyendo comportamiento observado, incidencias detectadas y conclusiones

4.4 Análisis del rendimiento del Sistema

De forma general, se observa que la mayoría de los sensores (IMU, GNSS y LiDAR) presentan un comportamiento cercano al tiempo real, con retardos reducidos generalmente inferiores a unos pocos segundos, condicionados principalmente por la calidad de la conexión inalámbrica. En contraste, la cámara introduce retardos significativamente mayores, convirtiéndose en el principal factor limitante del sistema.

Durante la fase de validación, se analizó el rendimiento global del sistema multisensor, prestando especial atención a aspectos como la latencia en la transmisión de datos, la estabilidad de la comunicación y el impacto del hardware en el comportamiento del sistema.

Uno de los factores más relevantes observados fue la latencia en la visualización de los datos, especialmente en sensores que generan un mayor volumen de información, como la cámara. Mientras que sensores como la IMU o el GNSS presentan una respuesta prácticamente inmediata, la transmisión de imágenes introduce retardos perceptibles.

Este comportamiento se debe principalmente a las limitaciones de procesamiento de la Raspberry Pi 4, al elevado volumen de datos generado por la cámara y al uso de comunicación inalámbrica mediante WiFi.

A medida que el vehículo se aleja del origen de la red, se producen incrementos en la latencia, pérdidas puntuales de conexión y una reducción en la fluidez de la transmisión de datos.

Este comportamiento limita el alcance operativo del sistema y pone de manifiesto la dependencia de la infraestructura de comunicaciones en sistemas embarcados que transmiten datos en tiempo real.

Como posible mejora, se plantea el uso de adaptadores WiFi USB de mayor potencia o alcance, que permitirían mejorar la estabilidad de la conexión y ampliar la distancia operativa del sistema.

Desde el punto de vista físico, se identificó que la incorporación de múltiples sensores, junto con la batería externa (*powerbank*), incrementa el peso total del vehículo. Este aumento de masa tiene un impacto directo en la velocidad del vehículo, el consumo energético y la autonomía del sistema.

Durante las pruebas, se observó que el vehículo presenta una menor velocidad y un consumo más elevado en comparación con su funcionamiento sin carga adicional. Este comportamiento pone de manifiesto la importancia de considerar el equilibrio entre capacidades sensoriales y eficiencia energética en sistemas móviles.

A pesar de estas limitaciones, el sistema presenta un comportamiento estable en condiciones normales de operación, siendo capaz de adquirir, procesar y transmitir datos de múltiples sensores de forma simultánea.

La arquitectura implementada permite mantener la funcionalidad del sistema incluso en condiciones no ideales, como variaciones en la calidad de la conexión o limitaciones de procesamiento, lo que demuestra la robustez del diseño adoptado.

En conjunto, el análisis del rendimiento pone de manifiesto que, si bien el sistema presenta limitaciones inherentes al hardware y a la comunicación inalámbrica, estas no comprometen su funcionalidad, siendo adecuado para aplicaciones de monitorización y validación en tiempo real. No obstante, se identifican diversas oportunidades de mejora orientadas a optimizar la eficiencia del sistema y ampliar su capacidad operativa.

Con el objetivo de sintetizar tanto el comportamiento observado como los factores que afectan al rendimiento del sistema, en la Tabla 4-2 se presenta un resumen global que integra las características de los sensores, su impacto en el sistema y posibles líneas de mejora.

Sensor / Factor	Frecuencia estimada	Latencia observada	Impacto observado	Posible mejora
IMU	Alta (decenas de Hz)	Muy baja (< 1–2 s)	Datos estables, bajo ruido	Filtrado avanzado / fusión sensorial
GNSS	~1 Hz	Baja ($\approx$ 2–5 s)	Error de posicionamiento (2–5 m)	Uso de RTK / correcciones diferenciales
LiDAR	~10 Hz	Baja (< 2–3 s)	Nube de puntos estable y coherente	Optimización de procesamiento
Cámara RGB	Baja (no continua)	Muy alta (20–30 s)	Flujo no fluido, parones frecuentes	Compresión / reducción de resolución / mejor hardware
Conexión WiFi	—	Variable	Pérdida de señal con distancia	Adaptador WiFi USB de mayor alcance
Procesamiento (Raspberry Pi)	—	—	Limitaciones en procesamiento simultáneo	Uso de hardware más potente (Raspberry Pi 5)
Peso del sistema	—	—	Menor velocidad y autonomía	Optimización del diseño / reducción de masa

Tabla 4-2 Resumen del rendimiento del sistema multisensor, incluyendo características operativas, impacto observado y posibles mejoras.

Como se observa, el rendimiento del sistema está condicionado principalmente por la transmisión de datos y las limitaciones del hardware, siendo la cámara el elemento más restrictivo, mientras que sensores como la IMU y el LiDAR presentan un comportamiento cercano al tiempo real.

4.5 Problemas encontrados y soluciones

Durante el desarrollo e implementación del sistema multisensor se identificaron diversas dificultades técnicas, tanto a nivel software como hardware, cuya resolución fue necesaria para garantizar el correcto

funcionamiento del sistema. El análisis de estos problemas permite comprender las limitaciones del sistema y justificar las decisiones adoptadas durante el proyecto.

Problemas de software

Uno de los primeros problemas detectados estuvo relacionado con la instalación de ROS 2 Jazzy en la Raspberry Pi. Durante este proceso, el sistema presentaba errores de dependencias al intentar instalar los paquetes mediante el gestor APT, impidiendo completar la instalación correctamente.

Tras analizar el problema, se identificó que la causa estaba asociada a la configuración incompleta de los repositorios de Ubuntu en arquitectura ARM. En particular, la ausencia de las ramas *noble-updates* y *noble-backports* impedía acceder a versiones compatibles de ciertos paquetes.

Como solución, se modificó manualmente el archivo de repositorios, incluyendo dichas ramas, lo que permitió resolver los conflictos de dependencias y completar la instalación de ROS 2 con éxito.

Otro problema relevante fue la aparición de errores del tipo “package not found” al ejecutar nodos desarrollados en el workspace. Este comportamiento se debía a que el entorno de ROS 2 no se cargaba automáticamente en nuevas terminales.

Para solucionarlo, se modificó el archivo .bashrc, incorporando la carga automática del entorno de ROS 2 y del workspace del proyecto. Esta solución permitió garantizar la disponibilidad de los paquetes en cada sesión de trabajo.

Asimismo, se detectó la necesidad de establecer una dirección IP fija en la Raspberry Pi. En configuraciones basadas en DHCP, la dirección IP puede variar entre reinicios, lo que obliga a modificar manualmente la dirección de conexión en Foxglove.

Para evitar este problema, se configuró una IP estática mediante Netplan, lo que permitió mantener constante la dirección de acceso y simplificar el proceso de conexión durante las pruebas.

Problemas de hardware y conexión

En el ámbito hardware, se identificaron limitaciones relacionadas con la conexión simultánea de múltiples dispositivos USB. Inicialmente, se intentó conectar todos los sensores a través de un hub USB, lo que provocó problemas durante el arranque del sistema y fallos en el reconocimiento de dispositivos.

Este comportamiento se atribuye a limitaciones en el suministro eléctrico y en la gestión de dispositivos USB por parte de la Raspberry Pi.

Como solución, se adoptó una distribución mixta de los dispositivos, conectando algunos sensores directamente a la Raspberry Pi y otros a través del hub USB. Esta configuración permitió mejorar la estabilidad del sistema y evitar problemas de inicialización.

Problemas de comunicación

Durante las pruebas, se observó que la calidad de la conexión WiFi dependía significativamente de la distancia al punto de acceso. A medida que el sistema se alejaba del hotspot, se producían incrementos en la latencia y pérdidas puntuales de conexión.

Este comportamiento afecta directamente a la visualización en tiempo real de los datos, especialmente en sensores con alto volumen de información como la cámara.

Como posible solución, se plantea el uso de adaptadores WiFi USB de mayor alcance, que permitirían mejorar la estabilidad de la conexión y ampliar el rango operativo del sistema.

Problemas mecánicos y de integración

Durante el montaje del sistema, se detectaron limitaciones en el diseño inicial de los soportes para las unidades IMU. Las piezas diseñadas no proporcionaban una fijación suficientemente estable, lo que generaba vibraciones y afectaba negativamente a la calidad de las medidas.

Para solucionar este problema, se optó por eliminar los soportes específicos y fijar las IMU mediante velcro adhesivo directamente sobre una superficie más estable del vehículo. Esta solución permitió mejorar la estabilidad del sensor y reducir el ruido en las mediciones.

Problemas de rendimiento energético

Otro aspecto relevante fue el impacto del peso del sistema en el comportamiento del vehículo. La incorporación de sensores, junto con la batería externa, incrementa la masa total del sistema, lo que afecta a directamente a la velocidad del vehículo, el consumo energético y la autonomía del sistema.

Este comportamiento pone de manifiesto la necesidad de optimizar el diseño en aplicaciones reales, donde el equilibrio entre funcionalidad y eficiencia resulta crítico.

Conclusión del apartado

En conjunto, la resolución de estos problemas ha permitido desarrollar un sistema funcional y robusto, al mismo tiempo que ha puesto de manifiesto las limitaciones inherentes al hardware y al entorno de trabajo. Estas consideraciones aportan información valiosa para futuras mejoras y optimizaciones del sistema.

La Tabla 4-3 resume los principales problemas identificados durante el desarrollo del sistema, así como las soluciones adoptadas

Problema	Causa	Solución
Error instalación ROS 2	Repositorios incompletos	Añadir <i>noble-updates</i> y <i>backports</i>
<i>Package not found</i>	Entorno no cargado	Modificación de <i>.bashrc</i>
Cambio de IP	Uso de DHCP	IP estática con <i>Netplan</i>
Fallos USB	Sobrecarga del hub	Distribución mixta
Latencia WiFi	Distancia al hotspot	Adaptador WiFi USB
Vibraciones IMU	Soporte inestable	Fijación con velcro
Bajo rendimiento	Peso del sistema	Optimización futura

Tabla 4-3 Problemas identificados durante el desarrollo del sistema y soluciones adoptadas.

4.6 Conclusiones de Validación

La fase de validación ha permitido comprobar el correcto funcionamiento del sistema multisensor desarrollado, tanto a nivel individual de cada sensor como en su integración dentro de una arquitectura basada en ROS 2.

En primer lugar, se ha verificado que todos los sensores empleados (IMU, GNSS, cámara y LiDAR) son capaces de adquirir datos de forma consistente y coherente con su naturaleza física. Los resultados obtenidos muestran un comportamiento estable en condiciones estáticas, así como una respuesta adecuada ante variaciones dinámicas, lo que confirma la correcta integración de los dispositivos en el sistema.

Desde el punto de vista global, se ha demostrado que la arquitectura implementada permite la adquisición simultánea de múltiples fuentes de información heterogéneas, así como su publicación y transmisión en tiempo real mediante ROS 2. La utilización de *Foxglove* como herramienta de visualización ha permitido monitorizar de forma conjunta todos los sensores, facilitando la validación del sistema de forma integral.

La validación en un entorno real, mediante la integración del sistema en un vehículo móvil alimentado de forma autónoma, ha permitido evaluar el comportamiento del sistema en condiciones operativas, teniendo en cuenta factores como la latencia de la comunicación inalámbrica, el consumo energético y las limitaciones del hardware embarcado.

No obstante, durante las pruebas se han identificado ciertas limitaciones que condicionan el rendimiento del sistema. En particular, destacan:

- La elevada latencia en la transmisión de datos de la cámara, con retardos de hasta 20–30 segundos
- La dependencia de la calidad de la conexión WiFi, que introduce variaciones en la latencia (generalmente inferiores a 5 segundos en otros sensores)
- El impacto del peso del sistema en la autonomía y el comportamiento del vehículo
- Las limitaciones de procesamiento de la Raspberry Pi 4

A pesar de estas restricciones, el sistema ha demostrado un comportamiento estable y funcional, cumpliendo con los objetivos planteados en el proyecto y permitiendo la monitorización en tiempo real de los sensores.

Asimismo, el desarrollo ha puesto de manifiesto la importancia de aspectos frecuentemente subestimados en sistemas embarcados, como la gestión de la alimentación, la distribución de dispositivos USB o la estabilidad de la comunicación. Estos factores, aunque externos al diseño puramente software, tienen un impacto directo en la fiabilidad del sistema.

En conjunto, los resultados obtenidos validan la arquitectura propuesta y demuestran la viabilidad de un sistema multisensor embarcado basado en ROS 2 para aplicaciones de percepción y monitorización. La solución desarrollada constituye una base sólida para futuras ampliaciones, como la incorporación de técnicas de fusión sensorial, mejora de la infraestructura de comunicaciones o la utilización de hardware más avanzado.

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Consecución de los objetivos

En este Trabajo Fin de Grado se ha alcanzado con éxito el desarrollo de un sistema multisensor embarcado basado en una Raspberry Pi, capaz de integrar diferentes dispositivos de adquisición de datos, como una IMU, un receptor GNSS, una cámara y un sensor LiDAR. La integración de estos sensores en una única plataforma ha permitido obtener información de distinta naturaleza de forma simultánea, cumpliendo con el objetivo principal del proyecto.

Para ello, se ha implementado una arquitectura software basada en *ROS 2*, que permite estructurar el sistema de forma modular mediante el uso de nodos y tópicos. Esta arquitectura ha facilitado la integración de múltiples sensores heterogéneos, permitiendo desacoplar la adquisición, el procesamiento y la visualización de los datos. En este sentido, el uso de *ROS 2* ha resultado clave para garantizar la escalabilidad del sistema, permitiendo la incorporación de nuevos sensores sin modificar la estructura principal, además de proporcionar un entorno robusto de comunicación, fundamental en sistemas multisensor embarcados.

Asimismo, se ha desarrollado un nodo en *ROS 2* encargado de la adquisición, procesamiento y publicación de los datos procedentes de los distintos sensores, lo que permite gestionar múltiples fuentes de información de forma simultánea y garantizar un flujo continuo de datos, cumpliendo con los requisitos planteados al inicio del proyecto.

Por otro lado, se ha logrado implementar un sistema de visualización en tiempo real mediante la herramienta *Foxglove*, permitiendo monitorizar de forma simultánea los datos de todos los sensores. La utilización de *Foxglove Bridge* ha permitido exponer los datos mediante *WebSocket*, facilitando la conexión con un equipo externo y cumpliendo el objetivo de desacoplar la adquisición de datos de su visualización.

Finalmente, el sistema ha sido validado en un entorno real mediante su integración en un vehículo móvil alimentado de forma autónoma. Esta validación ha permitido comprobar su funcionamiento en condiciones operativas, confirmando la viabilidad práctica de la solución desarrollada.

5.2 Conclusión final

En conjunto, los objetivos planteados en este Trabajo Fin de Grado han sido alcanzados satisfactoriamente, dando lugar a un sistema multisensor embarcado funcional, capaz de adquirir, procesar y visualizar datos en tiempo real mediante una arquitectura modular basada en *ROS 2*.

Más allá del cumplimiento de los objetivos, el desarrollo del proyecto ha permitido abordar de forma práctica los principales retos asociados a la integración de sistemas embarcados, incluyendo la gestión de múltiples sensores, la comunicación en tiempo real y las limitaciones del hardware disponible.

El sistema desarrollado demuestra que es posible implementar soluciones de percepción multisensor de bajo coste con un alto grado de funcionalidad, lo que resulta especialmente relevante en contextos donde el coste, la flexibilidad y la rapidez de despliegue son factores críticos.

Asimismo, el trabajo realizado pone de manifiesto la importancia de aspectos como la gestión energética, la estabilidad de las comunicaciones o la integración física de los sensores, los cuales tienen un impacto directo en el rendimiento del sistema y han condicionado las decisiones de diseño adoptadas.

El grado de cumplimiento de los objetivos planteados se sintetiza en la Tabla 5-1

Objetivo específico	Grado de cumplimiento	Evidencia en el trabajo
Seleccionar e integrar sensores (LiDAR, cámara, IMU y GNSS)	Alto	Capítulo 3.1: Plataformas y Sensores del Sistema
Diseñar soportes mediante CAD	Alto	Capítulo 3.2: Diseño e integración de soportes para sensores
Fabricar los soportes mediante impresión 3D	Alto	Figura 3-17 y Figura 3-19
Configurar la Raspberry Pi como unidad de procesamiento	Alto	Capítulo 3.3: Desarrollo del sistema embarcado
Integrar todos los componentes en la plataforma móvil	Alto	Figura 3-20
Implementar arquitectura basada en ROS 2	Alto	Capítulo 3.4 y 3.5: Arquitectura del sistema e Implementación Software del Sistema Multisensor
Desarrollar nodo de adquisición multisensor	Alto	Figura 3-28 y Figura 3-34
Visualizar datos en tiempo real	Alto	Figura 3-38 y Figura 3-39
Evaluar el comportamiento del sistema	Alto	Capítulo 4: Resultados

Tabla 5-1 Evaluación del grado de cumplimiento de los objetivos

5.3 Aplicación en el ámbito militar

En el contexto actual, caracterizado por la evolución de los conflictos modernos, el uso de sistemas no tripulados para tareas de reconocimiento ha adquirido una importancia creciente. En particular, el conflicto en Ucrania ha puesto de manifiesto el papel clave de plataformas terrestres y aéreas de bajo coste, utilizadas para la obtención de información en entornos de riesgo.

En este escenario, se observa una tendencia hacia el empleo de sistemas modulares y adaptables, en muchos casos desarrollados a partir de soluciones comerciales o de bajo coste, similares en concepto al sistema desarrollado en este proyecto. Estos sistemas permiten desplegar capacidades de reconocimiento sin comprometer la seguridad del personal, reduciendo el riesgo en operaciones tácticas.

En el ámbito de la Infantería de Marina, este tipo de plataformas presenta un especial interés, ya que permiten realizar tareas de exploración, vigilancia y adquisición de información en entornos complejos, como zonas urbanas o áreas costeras, donde la exposición directa del personal puede suponer un riesgo elevado.

Asimismo, el uso de tecnologías como la comunicación mediante fibra óptica, observada en algunos sistemas empleados en conflictos actuales, pone de manifiesto la importancia de reducir la latencia y mejorar la robustez de las comunicaciones en escenarios operativos. Este tipo de soluciones representa una posible evolución de sistemas como el desarrollado en este trabajo.

Por tanto, el proyecto realizado, aunque en un contexto académico, se alinea con las tendencias actuales en el desarrollo de sistemas no tripulados, demostrando el potencial de plataformas multisensor embarcadas como herramienta de apoyo en operaciones de reconocimiento.

5.4 Líneas Futuras

En cuanto a las líneas futuras, se propone, en primer lugar, la optimización del rendimiento del sistema mediante el uso de hardware más avanzado, como la Raspberry Pi 5, así como la implementación de técnicas de compresión de imagen y el ajuste de las frecuencias de adquisición. Estas mejoras permitirían reducir la latencia observada, especialmente en la transmisión de vídeo.

Por otro lado, dado que la conectividad ha demostrado ser un factor crítico durante el desarrollo del proyecto, se plantea la incorporación de soluciones de comunicación más robustas, como adaptadores WiFi de mayor alcance o tecnologías alternativas de baja latencia. Asimismo, podría explorarse el uso de comunicaciones cableadas o mediante fibra óptica en entornos específicos, con el objetivo de mejorar la estabilidad y fiabilidad del sistema.

Además, una línea de desarrollo especialmente relevante consiste en la implementación de algoritmos de fusión sensorial que permitan combinar la información procedente de la IMU, el GNSS y el LiDAR. Esto permitiría mejorar la precisión en la estimación del estado del sistema y avanzar hacia aplicaciones de navegación autónoma.

En esta misma línea, se plantea la evolución del sistema hacia una plataforma completamente autónoma, mediante la incorporación de algoritmos de navegación, planificación de trayectorias y evitación de obstáculos. Este avance permitiría pasar de un sistema de monitorización a un sistema con capacidad de operación autónoma.

Finalmente, a nivel físico, se propone el rediseño del sistema con el objetivo de reducir el peso y mejorar la distribución de los componentes, optimizando así tanto la autonomía como el comportamiento dinámico del vehículo.

6 BIBLIOGRAFÍA

En esta sección figurarán todas las referencias, sean recursos web, libros, artículos, etc., incluyendo la información de autores, título de la obra, nombre de la publicación, año, edición y enlace más fecha de último acceso en el caso de referencias a recursos online.

[1] D. C. Guastella y G. Muscato, «Learning-Based Methods of Perception and Navigation for Ground Vehicles in Unstructured Environments: A Review», *Sensors*, vol. 21, n.º 1, p. 73, ene. 2021, doi: 10.3390/s21010073.

[2] Q. Liu, Z. Li, S. Yuan, Y. Zhu, y X. Li, «Review on Vehicle Detection Technology for Unmanned Ground Vehicles», *Sensors*, vol. 21, n.º 4, p. 1354, ene. 2021, doi: 10.3390/s21041354.

[3] M. Nowakowski y J. Kurylo, «Usability of Perception Sensors to Determine the Obstacles of Unmanned Ground Vehicles Operating in Off-Road Environments», *Appl. Sci.*, vol. 13, n.º 8, p. 4892, ene. 2023, doi: 10.3390/app13084892.

[4] D. J. Yeong, G. Velasco-Hernandez, J. Barry, y J. Walsh, «Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review», *Sensors*, vol. 21, n.º 6, p. 2140, ene. 2021, doi: 10.3390/s21062140.

[5] «Mobile Robotics: A Practical Introduction | Springer Nature Link». Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://link.springer.com/book/10.1007/978-1-4471-0025-6>

[6] «RPLIDAR C1 Fusion Lidar | High-Res & High-Perf | SLAMTEC». Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://www.slamtec.com/en/c1/spec>

[7] «A Review of Multi-Sensor Fusion SLAM Systems Based on 3D LIDAR». Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://www.mdpi.com/2072-4292/14/12/2835>

[8] C. Ersü, E. Petlenkov, y K. Janson, «A Systematic Review of Cutting-Edge Radar Technologies: Applications for Unmanned Ground Vehicles (UGVs)», *Sensors*, vol. 24, n.º 23, p. 7807, ene. 2024, doi: 10.3390/s24237807.

[9] «Camera, LiDAR, and IMU Spatiotemporal Calibration: Methodological Review and Research Perspectives». Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://www.mdpi.com/1424-8220/25/17/5409>

[10] «State-of-the-Art Navigation Systems and Sensors for Unmanned Underwater Vehicles (UUVs)». Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://www.mdpi.com/2673-3161/6/1/10>

[11] «Sonar», NOAA Ocean Exploration. Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://oceanexplorer.noaa.gov/technology/sonar/>

[12] «GPS General Introduction - Navipedia». Accedido: 5 de abril de 2026. [En línea]. Disponible en: https://gssc.esa.int/navipedia/index.php?title=GPS_General_Introduction

[13] «An intuitive approach to the GNSS positioning - Navipedia». Accedido: 5 de abril de 2026. [En línea]. Disponible en: https://gssc.esa.int/navipedia/index.php/An_intuitive_approach_to_the_GNSS_positioning

[14] «GNSS Basic Observables - Navipedia». Accedido: 5 de abril de 2026. [En línea]. Disponible en: https://gssc.esa.int/navipedia/index.php?title=GNSS_Basic_Observables

[15] A. Harindranath y M. Arora, «A systematic review of user - conducted calibration methods for MEMS-based IMUs», *Measurement*, vol. 225, p. 114001, feb. 2024, doi: 10.1016/j.measurement.2023.114001.

- [16] V. Suvorkin, M. Garcia-Fernandez, G. González-Casado, M. Li, y A. Rovira-Garcia, «Assessment of Noise of MEMS IMU Sensors of Different Grades for GNSS/IMU Navigation», *Sensors*, vol. 24, n.º 6, p. 1953, ene. 2024, doi: 10.3390/s24061953.
- [17] «PhidgetSpatial Precision 3/3/3 - MOT0110_0 - Phidgets». Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://www.phidgets.com/?prodid=1205>
- [18] «AIS transponders». Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://www.imo.org/en/ourwork/safety/pages/ais.aspx>
- [19] «Positioning Systems for Unmanned Underwater Vehicles: A Comprehensive Review». Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://www.mdpi.com/2076-3417/14/21/9671>
- [20] «DVLs (Doppler Velocity Logs) at Teledyne Marine». Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://www.teledynemarine.com/products/product-line/navigation-positioning/doppler-velocity-logs>
- [21] H. Qian, M. Wang, M. Zhu, y H. Wang, «A Review of Multi-Sensor Fusion in Autonomous Driving», *Sensors*, vol. 25, n.º 19, p. 6033, ene. 2025, doi: 10.3390/s25196033.
- [22] DAVID L. HALL y JAMES LLINAS, «An Introduction to Multisensor Data Fusion».
- [23] Alaa Khamis, Fakhreddine O. Karray, Bahador Khaleghi, y Saiedeh N. Razavi, «Multisensor data fusion: A review of the state-of-the-art».
- [24] Greg Welch y Gary Bishop, «An Introduction to the Kalman Filter», *Dep. Comput. Sci. Univ. N. C. Chap. Hill Chap. Hill NC 27599-3175*.
- [25] «Practical 3-D Beam Pattern Based Channel Modeling for Multi-Polarized Massive MIMO Systems». Accedido: 6 de abril de 2026. [En línea]. Disponible en: <https://www.mdpi.com/1424-8220/18/4/1186>
- [26] «DEVCOM Ground Vehicle Systems Center - DEVCOM HQ». Accedido: 6 de abril de 2026. [En línea]. Disponible en: <https://devcom.army.mil/centerslabs/gvsc/>
- [27] «Perception Sensing Advancements for Autonomous Ground Systems – Army SBIR|STTR Program». Accedido: 6 de abril de 2026. [En línea]. Disponible en: <https://armysbir.army.mil/topics/perception-sensing-advancements-for-autonomous-ground-systems/>
- [28] M. Hernández Suárez LLanos, «Tres siglos», *Implant. Veh. No Tripul. En El Ambito Nav..*
- [29] F. Tornell Velazquez, «Tres siglos», *Vehículos No Tripul. Para Los Buques Armada*, p. 10.
- [30] Armada, «El Instituto Hidrográfico de la Marina finaliza la primera campaña hidrográfica expedicionaria con vehículo de superficie no tripulado en aguas del Cantábrico. - Noticias de la Armada - Armada - Ministerio de Defensa - Gobierno de España». Accedido: 6 de abril de 2026. [En línea]. Disponible en: https://armada.defensa.gob.es/ArmadaPortal/page/Portal/ArmadaEspa%F1nola/conocenosnoticias/prefLang-es/00noticias--2023--06--NT-128-CAMPANNA-HIDROGRAFICA-es?_pageAction=selectItem&_pageNum=6&_selectedNodeID=5783132¶mNo=000000
- [31] «Naval Oceanography: At The Leading Edge of Unmanned Systems», United States Navy. Accedido: 6 de abril de 2026. [En línea]. Disponible en: <https://www.navy.mil/Press-Office/News-Stories/display-news/Article/3328115/naval-oceanography-at-the-leading-edge-of-unmanned-systems/>
- [32] G. Gallego *et al.*, «Event-Based Vision: A Survey», *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, n.º 1, pp. 154-180, ene. 2022, doi: 10.3929/ethz-b-000521357.

- [33] «Advancements in Key Parameters of Frequency-Modulated Continuous-Wave Light Detection and Ranging: A Research Review». Accedido: 6 de abril de 2026. [En línea]. Disponible en: <https://www.mdpi.com/2076-3417/14/17/7810>
- [34] Z. Han *et al.*, «4D Millimeter-Wave Radar in Autonomous Driving: A Survey», 14 de junio de 2023, *arXiv*: arXiv:2306.04242. doi: 10.48550/arXiv.2306.04242.
- [35] «Edge-AI: A systematic review on architectures, applications, and challenges - ScienceDirect». Accedido: 6 de abril de 2026. [En línea]. Disponible en: <https://www.sciencedirect.com/science/article/pii/S1084804525002723>
- [36] S. Thrun, W. Burgard, y D. Fox, *Probabilistic Robotics*. en Intelligent Robotics and Autonomous Agents series. Cambridge, MA, USA: MIT Press, 2005.
- [37] «A Review of Simultaneous Localization and Mapping for the Robotic-Based Nondestructive Evaluation of Infrastructures». Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://www.mdpi.com/1424-8220/25/3/712>
- [38] «A Review of Research on SLAM Technology Based on the Fusion of LiDAR and Vision». Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://www.mdpi.com/1424-8220/25/5/1447>
- [39] Armada, «El Centro de Experimentación y Vehículos No Tripulados de la Armada recibe el vehículo no tripulado de superficie SEAD23 - Noticias de la Armada - Armada - Ministerio de Defensa - Gobierno de España». Accedido: 5 de abril de 2026. [En línea]. Disponible en: https://armada.defensa.gob.es/ArmadaPortal/page/Portal/ArmadaEspannola/conocenosnoticias/prefLang-es/00noticias--2025--12--NW-115-SEAD23-es?_pageAction=selectItem&_pageNum=0&_selectedNodeID=6025127¶mNo=000000
- [40] Armada, «El BAM “Audaz” integra vehículos no tripulados en la mar, en preparación para el ejercicio “Dynamic Messenger 22” - Noticias de la Armada - Armada - Ministerio de Defensa - Gobierno de España». Accedido: 6 de abril de 2026. [En línea]. Disponible en: https://armada.defensa.gob.es/ArmadaPortal/page/Portal/ArmadaEspannola/conocenosnoticias/prefLang-es/00noticias--2022--07--NW-065-AUDAZ-es?_pageAction=selectItem&_pageNum=42&_selectedNodeID=5001090¶mNo=000000
- [41] «ISO 13849-1:2023», ISO. Accedido: 6 de abril de 2026. [En línea]. Disponible en: <https://www.iso.org/standard/73481.html>
- [42] «ISO 21448:2022», ISO. Accedido: 6 de abril de 2026. [En línea]. Disponible en: <https://www.iso.org/standard/77490.html>
- [43] «ISO 26262 road vehicles functional safety», ISO. Accedido: 6 de abril de 2026. [En línea]. Disponible en: <https://www.iso.org/publication/PUB200262.html>
- [44] «ISO/SAE 21434:2021», ISO. Accedido: 6 de abril de 2026. [En línea]. Disponible en: <https://www.iso.org/standard/70918.html>
- [45] «IEC 61508-1:2010 | IEC». Accedido: 6 de abril de 2026. [En línea]. Disponible en: <https://webstore.iec.ch/en/publication/5515>
- [46] «DO-178() – RTCA». Accedido: 6 de abril de 2026. [En línea]. Disponible en: <https://www.rtca.org/do-178/>
- [47] «TRAXXAS 82256-4 MANUAL Pdf Download», ManualsLib. Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://www.manualslib.com/manual/3654388/Traxxas-82256-4.html>
- [48] R. P. Ltd, «Raspberry Pi 4 Model B specifications», Raspberry Pi. Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>

- [49] «Welcome to ELP – USB Camera Module Factory». Accedido: 6 de abril de 2026. [En línea]. Disponible en: <https://www.svpro.cc/>
- [50] «Camera software - Raspberry Pi Documentation». Accedido: 6 de abril de 2026. [En línea]. Disponible en: https://www.raspberrypi.com/documentation/computers/camera_software.html
- [51] «u-blox 7 Receiver Description».
- [52] «Netplan documentation», Netplan. Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://netplan.readthedocs.io/en/stable/netplan-apply/>
- [53] «Ubuntu (deb packages) — ROS 2 Documentation: Jazzy documentation». Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://docs.ros.org/en/jazzy/Installation/Ubuntu-Install-Debs.html>
- [54] «ROS 2», Foxglove Docs. Accedido: 5 de abril de 2026. [En línea]. Disponible en: <https://docs.foxglove.dev/docs/getting-started/frameworks/ros2>

ANEXO I: IMPLICACIONES SOCIALES, Y/O ECONÓMICAS, Y/O AMBIENTALES

El presente Trabajo Fin de Grado aborda el desarrollo de un sistema multisensor embarcado basado en una plataforma de bajo coste, integrando diferentes tecnologías de adquisición, procesamiento y visualización de datos en tiempo real. A lo largo del desarrollo del proyecto se han tenido en cuenta diversos aspectos relacionados con el impacto social, económico, ambiental y de seguridad, en línea con los criterios establecidos por el sello internacional de calidad EUR-ACE.

Desde el punto de vista social, el proyecto contribuye al desarrollo de sistemas tecnológicos aplicables a la obtención de información en entornos de riesgo, reduciendo la exposición directa del personal humano. En este sentido, los sistemas multisensor embarcados y no tripulados representan una herramienta de gran utilidad en aplicaciones de reconocimiento, inspección y monitorización, permitiendo operar en entornos potencialmente peligrosos sin comprometer la seguridad del operador.

En el ámbito militar, y en particular en el contexto de la Infantería de Marina, este tipo de sistemas adquiere especial relevancia. Los conflictos actuales han evidenciado la creciente importancia de plataformas no tripuladas para tareas de reconocimiento y vigilancia. En escenarios como el conflicto en Ucrania, se ha observado el uso extendido de sistemas terrestres y aéreos de bajo coste, en muchos casos desarrollados a partir de soluciones comerciales o adaptaciones tecnológicas similares en concepto al sistema desarrollado en este proyecto. Estos sistemas permiten obtener información del entorno en tiempo real, mejorando la toma de decisiones y reduciendo el riesgo para el personal desplegado.

Asimismo, se ha puesto de manifiesto la necesidad de mejorar aspectos como la latencia y la robustez de las comunicaciones, recurriendo en algunos casos a soluciones avanzadas como enlaces mediante fibra óptica o sistemas de comunicación más estables. Este tipo de mejoras se alinea con las posibles evoluciones futuras del sistema desarrollado en este trabajo.

Desde el punto de vista económico, el proyecto demuestra la viabilidad de desarrollar sistemas multisensor funcionales utilizando plataformas de bajo coste como la Raspberry Pi. Este enfoque reduce significativamente los costes de desarrollo en comparación con soluciones industriales o militares tradicionales, facilitando el acceso a tecnologías avanzadas y permitiendo la experimentación y validación de conceptos sin necesidad de grandes inversiones.

Además, el uso de herramientas software de código abierto, como ROS 2, contribuye a minimizar los costes asociados a licencias y favorece la reutilización del sistema en futuros desarrollos. La arquitectura modular implementada permite ampliar el sistema sin necesidad de rediseños completos, optimizando así el coste a largo plazo.

En términos de seguridad y salud, el desarrollo de sistemas no tripulados permite reducir la exposición del personal a entornos peligrosos, lo que constituye uno de los principales beneficios de este tipo de tecnologías. Asimismo, durante el desarrollo del proyecto se han tenido en cuenta aspectos relacionados con la seguridad eléctrica y la correcta alimentación de los dispositivos, garantizando un funcionamiento estable del sistema.

Desde el punto de vista ambiental, el proyecto presenta un impacto reducido, al tratarse de un sistema de bajo consumo energético basado en una plataforma eficiente como la Raspberry Pi. El uso de baterías externas (*powerbank*) permite un funcionamiento autónomo sin necesidad de infraestructuras energéticas complejas.

Además, el enfoque basado en el uso de prototipos y validación experimental evita la necesidad de desarrollar sistemas de mayor escala en fases iniciales, reduciendo el consumo de recursos materiales. El uso de simulación y visualización remota mediante herramientas como Foxglove también contribuye a minimizar desplazamientos y optimizar el proceso de validación.

Por otro lado, el desarrollo del sistema ha puesto de manifiesto la importancia de diseñar soluciones tecnológicas eficientes, tanto en términos de consumo energético como de uso de recursos computacionales, lo que se alinea con los principios de sostenibilidad en ingeniería.

En conclusión, el proyecto desarrollado demuestra la capacidad de integrar aspectos técnicos con consideraciones sociales, económicas, ambientales y de seguridad, evidenciando la importancia de este tipo de criterios en el diseño de sistemas de ingeniería modernos. Estas consideraciones se han reflejado a lo largo de la memoria, especialmente en los apartados de diseño del sistema, arquitectura, validación y análisis de rendimiento.

ANEXO II: REFLEXIONES ÉTICAS Y SOCIALES

El desarrollo de sistemas multisensor embarcados y plataformas no tripuladas, como la presentada en este Trabajo Fin de Grado, plantea diversas reflexiones éticas y sociales, especialmente debido a su carácter de tecnología de doble uso. Este tipo de sistemas puede emplearse tanto en aplicaciones civiles (como monitorización del entorno, inspección de infraestructuras o asistencia en situaciones de emergencia) como en contextos militares, donde pueden utilizarse para tareas de reconocimiento, vigilancia o apoyo táctico.

Esta dualidad exige una reflexión crítica sobre el uso y las implicaciones de estas tecnologías, así como sobre los límites que deben establecerse en su desarrollo y aplicación. En el ámbito militar, el empleo de sistemas no tripulados permite reducir la exposición directa del personal a situaciones de riesgo, lo que supone una ventaja significativa desde el punto de vista de la seguridad. Sin embargo, al mismo tiempo, introduce nuevos desafíos éticos relacionados con la toma de decisiones, la responsabilidad en el uso de la tecnología y el posible impacto sobre terceros.

En el contexto actual, caracterizado por conflictos donde el uso de sistemas no tripulados ha aumentado significativamente, como es el caso de la guerra en Ucrania, se ha observado la utilización de plataformas de bajo coste para tareas de reconocimiento. En muchos casos, estos sistemas se desarrollan a partir de tecnologías accesibles o soluciones adaptadas, similares en concepto al sistema desarrollado en este proyecto. Esta realidad pone de manifiesto cómo tecnologías relativamente sencillas pueden adquirir un papel relevante en entornos operativos, lo que refuerza la necesidad de un uso responsable y regulado.

En particular, se ha evidenciado la importancia de aspectos como la latencia, la robustez de las comunicaciones o la autonomía del sistema, factores que condicionan directamente su aplicabilidad en entornos reales. La adopción de soluciones técnicas avanzadas, como enlaces de comunicación mediante fibra óptica o plataformas hardware más potentes, refleja la evolución de este tipo de sistemas hacia configuraciones cada vez más eficientes y adaptadas a escenarios operativos exigentes.

Desde el punto de vista ético, es fundamental que el desarrollo y utilización de estas tecnologías se realice dentro del marco del Derecho Internacional Humanitario. En este sentido, deben respetarse principios fundamentales como la distinción entre objetivos militares y civiles, la proporcionalidad en el uso de la fuerza y la necesidad operativa. Aunque el sistema desarrollado en este trabajo no tiene un carácter ofensivo, su naturaleza como plataforma de adquisición de información lo sitúa dentro de un contexto en el que estas consideraciones resultan relevantes.

Asimismo, el proyecto pone de manifiesto el valor de este tipo de tecnologías en tareas no ofensivas, como la obtención de información del entorno o el apoyo a la toma de decisiones. En el ámbito de la Infantería de Marina, sistemas como el desarrollado pueden contribuir a mejorar las capacidades de reconocimiento en entornos complejos, reduciendo la exposición del personal y aumentando la seguridad en operaciones.

Por otro lado, desde una perspectiva social, el desarrollo de tecnologías accesibles y de bajo coste plantea tanto oportunidades como riesgos. Por un lado, facilita la innovación y el acceso a herramientas avanzadas; por otro, puede dar lugar a un uso indebido si no existe un marco adecuado de regulación y control. Este aspecto refuerza la importancia de la formación técnica y ética de los ingenieros responsables de su desarrollo.

En conclusión, el presente proyecto, aunque centrado en un entorno académico y con fines de investigación y desarrollo, pone de manifiesto la relevancia de considerar los aspectos éticos y sociales en el diseño de sistemas tecnológicos. La responsabilidad en el desarrollo y uso de estas tecnologías resulta fundamental para garantizar que su aplicación contribuya al bienestar, la seguridad y el respeto de los principios fundamentales en cualquier ámbito de utilización.

ANEXO III: DECLARACIÓN SOBRE EL USO DE HERRAMIENTAS DE INTELIGENCIA ARTIFICIAL

En el desarrollo del presente Trabajo Fin de Grado se ha recurrido al uso de herramientas basadas en inteligencia artificial como apoyo puntual en tareas de redacción, creación de imágenes, revisión lingüística y mejora de la claridad expositiva del documento.

El uso de dichas herramientas se ha limitado exclusivamente a la reformulación de textos, corrección gramatical y estructuración de contenidos, con el objetivo de lograr una expresión más precisa, coherente y adecuada al contexto académico. En ningún caso se han empleado para la generación autónoma de contenido técnico, obtención de resultados experimentales, análisis de datos o elaboración de conclusiones.

Todos los contenidos técnicos incluidos en este trabajo, tales como el diseño del sistema, la implementación del código, la integración de sensores, la configuración del entorno ROS 2, así como la realización de pruebas y la interpretación de resultados, han sido desarrollados íntegramente por el autor.

Asimismo, se ha llevado a cabo una revisión crítica de todos los textos en los que se ha empleado apoyo de herramientas de inteligencia artificial, garantizando en todo momento la veracidad, coherencia y adecuación de la información presentada.

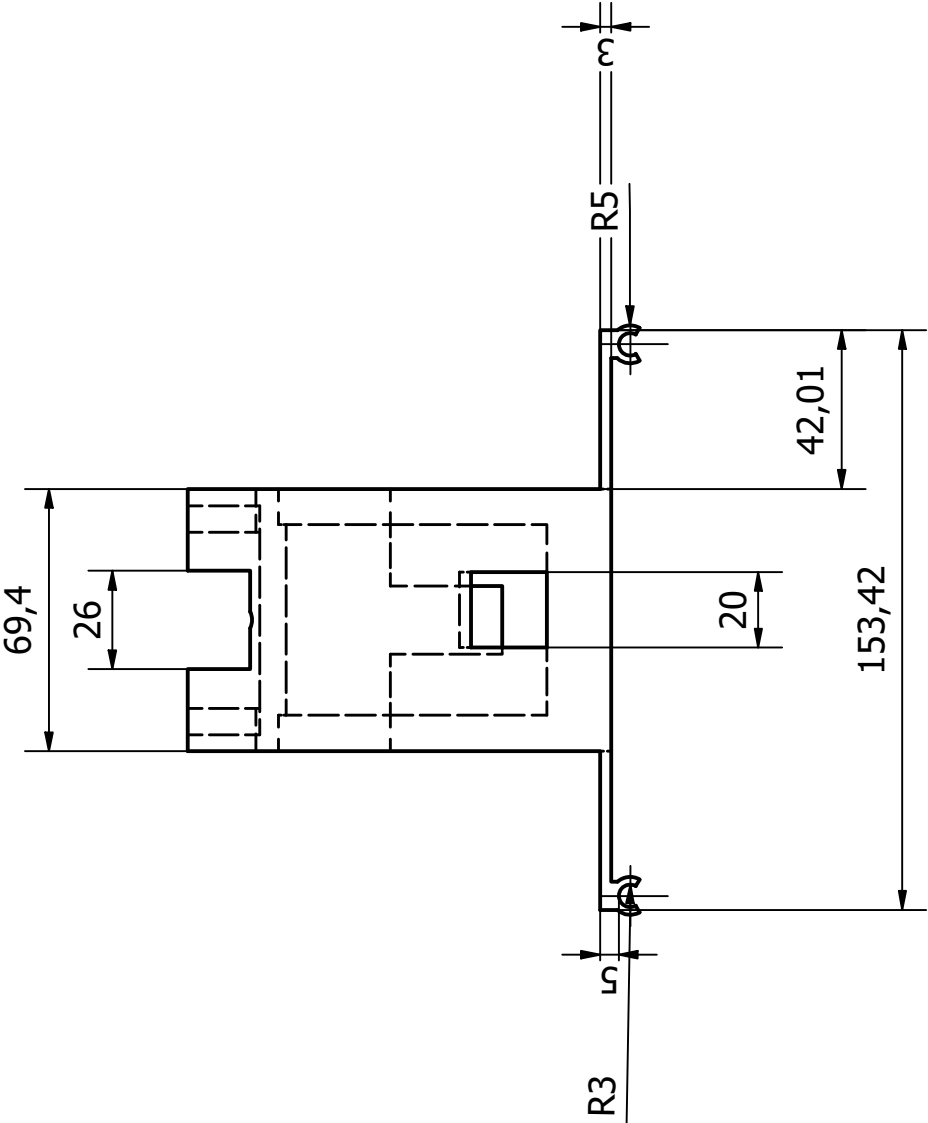
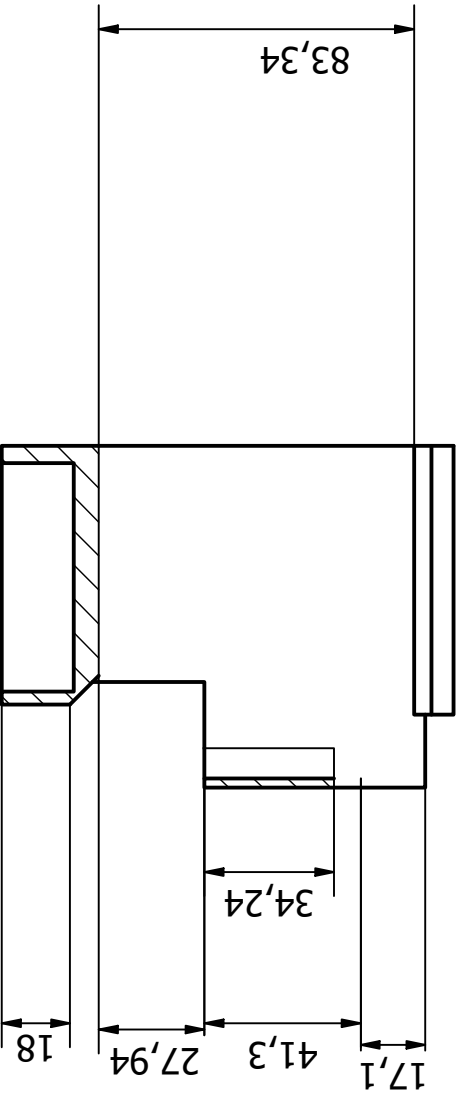
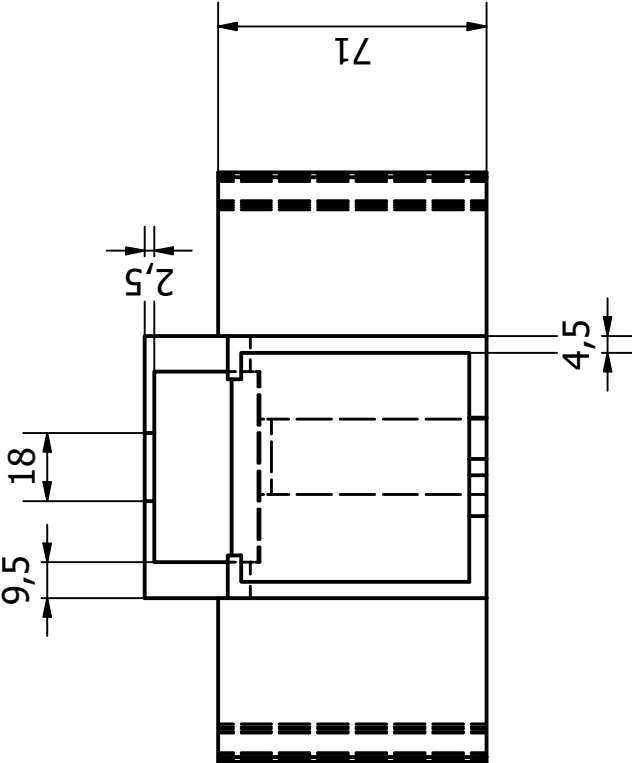
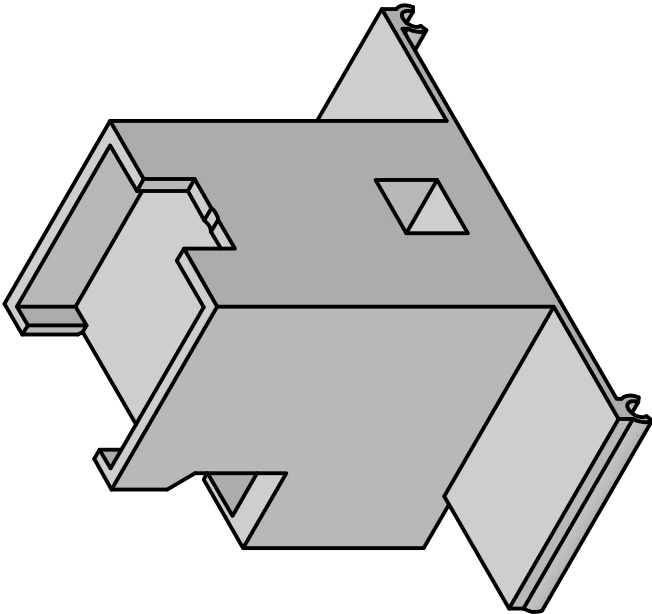
El uso de inteligencia artificial se ha realizado, por tanto, como una herramienta de apoyo complementaria, manteniendo en todo momento la autoría, responsabilidad y control sobre el contenido final del documento.

ANEXO IV: PLANIMETRÍA

En este anexo se presentan los planos técnicos correspondientes a las piezas diseñadas en el desarrollo del proyecto, generados mediante herramientas de diseño asistido por ordenador (CAD). Dichos planos incluyen la información geométrica y dimensional necesaria para la correcta fabricación y ensamblaje de los componentes.

La documentación gráfica recogida permite comprender en detalle la solución constructiva adoptada, así como su integración dentro del sistema global desarrollado

			1				
Nº de plano 1			Autor		2		
Formato A3			Designación		3		
Escala 1:1			Nº TFG		4		
Manuel Luaces Cebreiro			Soporte LiDAR-Cámara			5	
Diseño e implementación de un sistema de sensorización y transmisión de datos para un vehículo terrestre no tripulado						6	



			1			
Manuel Luaces Cebreiro		Soporte Raspberry Pi 4		2		
Autor		Designación				
Nº de plano 1		Formato A3				
Formato A3		Escala 1:1		3		
Nº TFG		Diseño e implementación de un sistema de sensorización y transmisión de datos para un vehículo terrestre no tripulado		4		

