



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Montaje, configuración y evaluación experimental de un robot
móvil educativo con herramientas de inteligencia artificial
para percepción y navegación*

Grado en Ingeniería Mecánica

ALUMNO: Jesús Carlos Bastardés López

DIRECTORES: José Antonio González Prieto

CURSO ACADÉMICO: 2025-2026

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Montaje, configuración y evaluación experimental de un robot
móvil educativo con herramientas de inteligencia artificial
para percepción y navegación*

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Cuerpo General

UniversidadeVigo

RESUMEN

Este Trabajo de Fin de Grado presenta el montaje, configuración y evaluación de un robot móvil educativo con herramientas de inteligencia artificial para su uso docente con oficiales alumnos de la Armada. Se selecciona la plataforma ROSMASTER A1 por su coste, contenido y modularidad; se integran en él: una Raspberry Pi 5 (*Robot Operating System*, ROS Humble en contenedores Docker), un LiDAR 2D T-mini, una cámara 2 MP e *Inertial Measurement Unit*, IMU en un chasis Ackermann de aluminio. Se documenta el ensamblaje en cuatro fases y la puesta en marcha de la pila software (Secure Shell (SSH) o Virtual Network Computing (VNC), contenedores, lanzadores ROS 2). Experimentalmente se demuestran funciones de mapeado *Light Detection And Ranging*, LiDAR 2D, navegación autónoma con evitación de obstáculos y navegación sobre mapa, así como visión con seguimiento de color, posturas e identificación de objetos mediante modelos ligeros *You Only Look Once* (YOLO). Se reportan métricas de uso de Central Processing Unit (CPU) y de Random Access Memory (RAM) y se analizan incidencias reproducibles (deriva y errores en SLAM por choques o detenciones; limitaciones de la cámara,...). Los resultados confirman la idoneidad de la plataforma como recurso didáctico para introducir navegación, percepción y fusión sensorial, proponiéndose mejoras en su diseño e incorporando sensores más complejos y líneas futuras. El trabajo aporta guías operativas y checklists reproducibles del funcionamiento de la plataforma para la futura asignatura de Sistemas Autónomos.

PALABRAS CLAVE

Robótica educativa, ROS 2, SLAM, LiDAR 2D, Navegación autónoma

AGRADECIMIENTOS

Con estas palabras, sin extenderme seguramente lo necesario y merecido, pretendo agradecer a todos aquellos que me han acompañado a lo largo de esta singladura en demanda de cumplir lo que es, además de un sueño, una vocación de servicio a España.

En primer lugar, doy las gracias al Centro Universitario de la Defensa por haberme dotado de los medios y la oportunidad de realizar este proyecto y de haberme ayudado a forjar un tipo de carácter de oficial de armada gracias a la capacidad de resiliencia puesta en práctica durante estos cinco años de academia. También por todo lo aprendido tanto práctico para nuestro futuro profesional como de cultura para nuestra propia formación integral como personas. Además, en especial quiero agradecer a mi tutor de TFG por darme libertad plena para llevar a cabo el proyecto y así divertirme a la vez que he aprendido.

En segundo lugar, he de agradecer a mis compañeros y amigos en especial a la “Camareta a Silencio” por los buenos momentos y anécdotas que perdurarán en el recuerdo de nuestro paso por la escuela.

Por último, agradezco a mi familia por todo el apoyo proporcionado en el pasado, presente y en no dentro de mucho, el futuro. Han demostrado estar siempre ahí para lo bueno y lo malo aun siendo difícil hacerlo.

CONTENIDO

Contenido	1
Índice de Figuras	4
Índice de Tablas.....	7
1 Introducción y objetivos	8
1.1 Contexto: robótica educativa e integración de Inteligencia Artificial	8
1.2 Motivación del proyecto	8
1.3 Objetivos generales	9
1.4 Objetivos específicos	9
1.5 Organización de la memoria	10
2 Estado del arte	11
2.1 Siglas y acrónimos	11
2.2 Papel en la enseñanza técnica	11
2.3 Plataformas Comerciales.....	11
2.4 Robots Open-Source	14
2.5 Software Educativo	14
2.6 Plataformas terrestres (UGV).....	16
2.7 Drones educativos (UAV).....	18
2.8 Robots de superficie marítima (USV).....	19
2.9 Plataformas submarinas educativas (ROV/AUV).....	20
2.10 Sensores en robótica móvil	21
2.10.1 Introducción de sensores en robótica móvil	21
2.10.2 Ultrasonidos	21
2.10.3 Cámaras	23
2.10.4 Sensores IR	24
2.10.5 Unidad de medición inercial	26
2.10.6 Sistemas LiDAR	27
2.11 Algoritmos de navegación	28
2.11.1 Control reactivo	28
2.11.2 SLAM y mapeado.....	28
2.12 Uso de IA en robótica móvil	29
2.13 Relevancia de estas plataformas en la formación militar moderna.....	30
2.14 Plataforma elegida para el proyecto.....	31

3 Desarrollo del TFG.....	33
3.1 Dimensiones y características	33
3.2 Hardware.....	34
3.2.1 Componentes estructurales	34
3.2.2 Componentes electrónicos	34
3.2.3 Sensores	35
3.3 Ensamblaje	36
3.3.1 Estado inicial	36
3.3.2 Primera fase	36
3.3.3 Segunda fase	38
3.3.4 Tercera fase.....	39
3.3.5 Cuarta Fase	41
3.4 Software y configuración	43
3.4.1 Raspberry Pi 5.....	43
3.4.2 Programas instalados	44
3.4.3 Movimiento RC	45
3.4.4 Cámara y su funcionamiento	46
3.4.5 LiDAR y su funcionamiento.....	48
3.4.6 Configuración de Altavoz y módulo de IA.....	49
3.5 Funciones Avanzadas.....	50
3.5.1 Mapeado de una superficie	50
3.5.2 Navegación autónoma y evitación de obstáculos	51
3.5.3 Identificación de colores, objetos, gestos y posturas	52
3.5.4 Seguimiento de línea.....	55
3.5.5 Función de enjambre.....	55
4 Resultados	57
4.1 Alcance y criterios de valoración.....	57
4.2 Montaje: síntesis de resultados	57
4.2.1 Resumen del proceso	57
4.2.2 Incidencias y correcciones	58
4.2.3 Tiempo y recursos.....	58
4.2.4 Lecciones aprendidas y mejoras	58
4.3 Configuración y verificación funcional	59
4.3.1 Imagen del sistema y contenedor.....	59
4.3.2 Conectividad y acceso	59
4.3.3 Puesta en marcha de sensores	60

4.3.4 Observaciones de rendimiento.....	60
4.3.5 Checklist reproducibles	62
4.4 Funciones avanzadas y evidencia de funcionamiento.....	63
4.4.1 Mapeado LiDAR 2D.....	63
4.4.2 Navegación	64
4.4.3 Visión e IA.....	64
4.4.4 Seguimiento de línea.....	65
4.4.5 Enjambre.....	65
5 Conclusiones y líneas futuras	67
5.1 Valoración global del proyecto	67
5.2 Conclusiones técnicas	67
5.2.1 Plataforma y hardware	67
5.2.2 Arquitectura software y ecosistema ROS 2	68
5.3 Conclusiones pedagógicas y relevancia para la formación militar	68
5.4 Limitaciones detectadas	69
5.5 Propuestas de mejora	69
5.5.1 Mejoras técnicas	69
5.5.2 Mejoras docentes	70
5.6 Líneas futuras	70
5.7 Conclusión	71
6 Bibliografía.....	73
Anexo I: Siglas Y Acrónimos	76
Anexo II: Implicaciones Sociales, Económicas, y Ambientales	81
Anexo III: Reflexiones Éticas y Sociales	83
Anexo IV: Contenido Audiovisual.....	86

ÍNDICE DE FIGURAS

Figura 2-1 Imagen del mBOT [1].....	12
Figura 2-2 Imagen Thymio II [2]	13
Figura 2-3 Imagen del TurtleBot 4 [3]	13
Figura 2-4 Programa RViz en funcionamiento, imagen propia	14
Figura 2-5 Imagen del Clearpath Jackal (izquierda) y del Clearpath Husky (derecha) [4].....	16
Figura 2-6 Imagen del RB SUMMIT [5]	16
Figura 2-7 Imagen del Husarion ROSbot XL [6].....	17
Figura 2-8 Imagen del DJI Ryze Tello EDU [7] (izquierda) y del DJI RoboMaster TT [8] (derecha)	18
Figura 2-9 Imagen del Crazyflie 2.1 Brushless [9]	19
Figura 2-10 Imagen del Holybro X500 V2 [10].....	19
Figura 2-11 Imagen del BlueBoat [11](izquierda) y del Heron USV [12] (derecha)	20
Figura 2-12 Imagen de un USV de Hardware abierto [13]	20
Figura 2-13 Imagen del OpenROV/Trident [14] (izquierda) y del BlueROV2 [15] (derecha).....	21
Figura 2-14 Esquema de funcionamiento del sensor HC SR04 [16].....	22
Figura 2-15 Imagen de la cámara Luxonis OAK D [17] (izquierda) y de la cámara IMX219 160 [18] (derecha).....	24
Figura 2-16 Imagen de la cámara Himax HM01B0 [19] (izquierda) y del AI deck 1.1 [20] (derecha)	24
Figura 2-17 Imagen de explicación de funcionamiento de sensor IR [21].....	25
Figura 2-18 Imagen del sensor Z Ranger Deck v2 [22] (izquierda) y del sensor Me Line Follower [23] (derecha).....	26
Figura 2-19 Esquema IMU [24]	26
Figura 2-20 Imagen del MPU 9250 [25] (izquierda) y del ICM 42688 P [26] (derecha)	27
Figura 2-21 Explicación gráfica de funcionamiento de un sistema LiDAR [27].....	27
Figura 2-22 Imagen del RPLiDAR A1[28].....	28
Figura 2-23 Imagen del recorrido realizado y del mapa generado mediante una técnica de SLAM [29]	29
Figura 2-24 Imagen de YOLO analizando poses de personas [30].....	30
Figura 2-25 Imagen de las plataformas UXVs en el ámbito militar [31].....	31
Figura 3-1 Imagen de medidas ROSMASTER A1 [32].....	33
Figura 3-2 Imagen de ejemplo de dirección Ackermann [33].....	34
Figura 3-3 Chasis del ROSMASTER A1, imagen propia.....	34
Figura 3-4 Raspberry Pi 5 , imagen propia (izquierda) y Yahboom ROS Robot Control Board V3.0 [32] (derecha).....	35

Figura 3-5 Placa de expansión USB 3.0 (izquierda) y de módulo de IA (derecha), imagen propia	35
Figura 3-6 Sensor T-mini LiDAR (izquierda) y de la cámara 2MP (derecha), imagen propia.....	36
Figura 3-7 Kit básico del ROSMASTER A1 fuera de su embalaje, imagen propia	36
Figura 3-8 Componentes del montaje de la 1º fase, imagen propia	37
Figura 3-9 Interior de la base del ROSMASTER A1 ya ensamblada (a la izquierda) y finalizado (a la derecha), imagen propia	38
Figura 3-10 Componentes para el montaje de la 2º fase, imagen propia	38
Figura 3-11 Montaje de la cámara en el chasis (a la izquierda) y conexión de los cables del servo de la cámara (a la derecha), imagen propia	39
Figura 3-12 Montaje de la 2º fase finalizada, imagen propia.....	39
Figura 3-13 Componentes para el montaje de la 3ª fase, imagen propia	40
Figura 3-14 Instalación de la placa de expansión USB 3.0, imagen propia.....	40
Figura 3-15 Montaje de 3º fase finalizado, imagen propia.....	41
Figura 3-16 Componentes para el montaje de la 4º fase, imagen propia	41
Figura 3-17 Montaje inferior (izquierda) y superior (derecha) de la superestructura, imagen propia	42
Figura 3-18 Montaje del ROSMASTER A1 finalizado, imagen propia	42
Figura 3-19 Tutorial proporcionado por la página oficial de Yahboom [34].....	43
Figura 3-20 Proceso del traspaso de la imagen oficial a la tarjeta microSD empleando el programa de “Raspberry Pi Imager”, imagen propia	44
Figura 3-21 Pantalla LCD del ROSMASTER A1, imagen propia.....	44
Figura 3-22 Captura del escritorio de la Raspberry Pi 5 tras la conexión por medio del programa "RealVNC Connect", imagen propia	45
Figura 3-23 Entrada en el Docker de trabajo, imagen propia.....	45
Figura 3-24 Mando RC, imagen propia.....	46
Figura 3-25 Demostración del funcionamiento y visualización de la cámara del ROSMASTER A1, imagen propia.....	47
Figura 3-26 Programa RViz mostrando la nube de puntos generada por el LiDAR, imagen propia	49
Figura 3-27 Función de mapeado y datos recopilados, a la vez, mostrados a través del programa RViz, imagen propia	50
Figura 3-28 Mapa final obtenido, imagen propia.....	51
Figura 3-29 Función de navegación con mapa iniciada, mapa y ruta cargadas, imagen propia	52
Figura 3-30 Función de seguimiento de color activada, imagen propia	53
Figura 3-31 Función de seguimiento de postura corporal activada, imagen propia.....	53
Figura 3-32 Función de identificación de objetos, imagen propia	54

Figura 3- 33 Función de seguimiento por gestos: avanzar (izquierda) y detenerse (derecha), imagen propia 54

Figura 3- 34 Función de seguimiento de línea, imagen propia 55

Figura 4-1 Mapeado defectuoso, imagen propia 63

ÍNDICE DE TABLAS

Tabla 2-1 Ejemplos de Robots Open-Source mencionados y su ubicación el proyecto	14
Tabla 2-2 Software educativos más empleados	15
Tabla 2-3 Comparativa de robots terrestres educativos (UGV)	18
Tabla 2-4 Sensores de ultrasonidos empleados en diferentes plataformas.....	23
Tabla 3-1 Tabla que relaciona el movimiento del robot con los controles del mando.....	46
Tabla 4-1 Rendimientos asociados a funciones y tareas llevadas a cabo por el robot.....	62
Tabla 4-2 Checklist básico para localizar funciones e iniciarlas.....	62
Tabla A1-1 Lista de siglas y acrónimos del proyecto, con su significado y explicación de uso en el contexto de la robótica móvil educativa	80
Tabla A4-1 Contenido audiovisual del proyecto.....	86

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Contexto: robótica educativa e integración de Inteligencia Artificial

La robótica educativa ha experimentado un crecimiento significativo en los últimos años gracias a la disponibilidad de plataformas accesibles, modulables y de bajo coste que permiten introducir conceptos de ingeniería de forma práctica y motivadora. Estas plataformas facilitan el aprendizaje activo al integrar en un único sistema aspectos de mecánica, electrónica, programación, control automático y procesamiento de señales, convirtiéndose en herramientas eficaces para la formación de alumnos en entornos académicos técnicos.

Paralelamente, el desarrollo de la inteligencia artificial (IA) ha democratizado el acceso a algoritmos de percepción y toma de decisiones que anteriormente requerían una gran capacidad computacional. Modelos ligeros optimizados para dispositivos embebidos permiten que pequeños robots móviles sean capaces de realizar tareas como reconocer objetos, seguir trayectorias, detectar obstáculos y, en general, ser capaces de interpretar e integrar estímulos de diferentes sensores físicos (visuales, Global Positioning System (GPS), unidades inerciales, lidar, sonar, etc.) con la precisión adecuada para poder interpretar su entorno e interactuar de forma dinámica con él, permitiéndole ejecutar tareas de diversa complejidad lo cual permite acercarse al objetivo de crear contenidos adecuados para tareas educativas relacionadas con los sistemas autónomos.

En el ámbito militar, y particularmente en la Armada, la progresiva incorporación de sistemas autónomos, vehículos no tripulados, sensores inteligentes, sistemas de vigilancia, apoyo logístico automatizado y plataformas robotizadas, hace necesario que los futuros oficiales comprendan los principios de funcionamiento de estos sistemas. La robótica educativa con IA accesible se presenta, por tanto, como un recurso formativo idóneo para introducir estas tecnologías desde un entorno controlado y seguro, pero con un elevado valor didáctico.

1.2 Motivación del proyecto

La formación de los oficiales alumnos en la Escuela Naval Militar exige que éstos adquieran competencias tecnológicas acordes con los retos actuales de la Armada, tanto en operaciones como en tareas de mantenimiento, supervisión de sistemas autónomos y toma de decisiones basada en información sensorizada. La utilización de un robot móvil educativo permitiría acercar a los alumnos, de manera práctica y aplicada, conceptos clave relacionados con la navegación autónoma, la percepción mediante sensores y el uso de IA para interpretar el entorno.

Los tres elementos fundamentales que caracterizan a los sistemas inteligentes —percepción, control y planificación— están presentes de forma integrada en este tipo de plataformas educativas. Esta convergencia permite recrear un entorno formativo en el que el alumno puede adentrarse en el estudio y comprensión de los principales componentes que intervienen en el diseño de sistemas autónomos. Gracias a esta estructura, el aprendizaje trasciende la simple programación, ofreciendo una visión global que abarca desde la adquisición y procesamiento de datos sensoriales, hasta la toma de decisiones y la ejecución de acciones en entornos dinámicos y cambiantes

La motivación central de este proyecto radica en analizar y validar las características de una plataforma didáctica que pueda emplearse en sesiones formativas, prácticas de laboratorio o ejercicios complementarios en la asignatura de sistemas autónomos del Centro Universitario de la Defensa, en la Escuela Naval Militar, la cual será impartida durante el cuarto curso del alumnado, tras la implantación de la nueva titulación que entra en vigor a partir del curso 2026-2027. Este robot permitirá que los alumnos experimenten con tecnologías actuales, comprendan sus limitaciones, se familiaricen con su integración y adquieran competencias necesarias para el uso futuro de sistemas autónomos más complejos en su carrera profesional dentro de la Armada.

Además, el proyecto pretende iniciar el estudio e integración de un material educativo económico y escalable, que pueda ser mejorado en futuros cursos académicos o ampliado por otros alumnos en posteriores Trabajos de Fin de Grado, mediante, por ejemplo, la integración de nuevos sensores o actualizaciones y mejoras de los elementos ya existentes o bien probando nuevos sistemas de IA relacionados con el procesamiento de datos de sensores asociados a sistemas autónomos.

1.3 Objetivos generales

El objetivo general de este trabajo es montar, configurar y evaluar las capacidades prácticas de un robot móvil educativo que integre herramientas de inteligencia artificial para tareas de percepción y navegación, con la finalidad de emplearlo como recurso docente en la formación de oficiales alumnos de la Armada.

La tarea relacionada con la selección del robot no se ha contemplado como parte del desarrollo del presente Trabajo Fin de Grado, ya que este proceso tuvo que realizarse con anterioridad al inicio del proyecto. La elección y adquisición del material fueron pasos necesarios para garantizar la disponibilidad del equipo durante la fase de desarrollo e implementación del entorno de simulación. Por tanto, se parte de un sistema robótico previamente definido y operativo, sobre el cual se construyen las herramientas y funcionalidades descritas en este trabajo.

1.4 Objetivos específicos

Para alcanzar el objetivo general se plantean los siguientes objetivos específicos:

1. Justificar la plataforma robótica que se ha considerado en este TFG para el entorno docente militar en el que se pretende emplear esta herramienta educativa.
2. Realizar el montaje mecánico y electrónico del robot, integrando sensores, actuadores y módulos de procesamiento.
3. Configurar y testear la arquitectura software necesaria para controlar el movimiento, gestionar los sensores y permitir la operación autónoma del robot.
4. Integrar herramientas de inteligencia artificial para la percepción del entorno, incluyendo, por ejemplo, técnicas de visión artificial o de procesamiento de datos de la unidad láser (**LiDAR**).
5. Implementar estrategias de navegación autónoma adecuadas a un robot educativo, tales como seguimiento de trayectorias o evitación de obstáculos.

6. Diseñar y ejecutar pruebas experimentales para evaluar el rendimiento del robot en escenarios controlados.
7. Analizar los resultados obtenidos y valorar la idoneidad del robot como herramienta docente en la Armada.
8. Proponer mejoras y líneas futuras para ampliar las capacidades del robot y su utilidad formativa.

1.5 Organización de la memoria

La memoria de este Trabajo de Fin de Grado se estructura de acuerdo con el siguiente esquema general:

- El **Capítulo 1** tiene la finalidad de dar a conocer la introducción y objetivos del presente trabajo de fin de grado, presentando desde un primer momento el contexto de la robótica educativa junto la integración de la IA, la motivación del proyecto, objetivos generales y específicos y la organización de la memoria.
- El **Capítulo 2** presentará el estado del arte relacionado con robots móviles educativos, plataformas comerciales, sensores, algoritmos de navegación y técnicas de inteligencia artificial aplicadas a la percepción.
- El **Capítulo 3** describirá el desarrollo del trabajo que incluye el montaje del robot, la integración de sensores, el funcionamiento de la arquitectura software, la puesta en marcha de los sensores y aplicación de IA, y el proceso de activación de una serie de funciones avanzadas.
- El **Capítulo 4** incluirá la evaluación experimental del montaje, de la configuración del software y de las capacidades avanzadas del robot, así como los resultados obtenidos en las pruebas realizadas.
- El **Capítulo 5** resume las conclusiones del proyecto, desde una perspectiva general, técnica que trata del hardware y de software, mejoras de diseño de la plataforma y propone aplicaciones docentes y posibles líneas futuras.
- Finalmente, en el **Capítulo 6**, se incluye la bibliografía y los anexos sobre las implicaciones sociales, económicas y ambientales, así como las reflexiones éticas y sociales del proyecto.

2 ESTADO DEL ARTE

2.1 Siglas y acrónimos

A lo largo del proyecto nos encontraremos con un gran número de siglas y acrónimos; es por ello por lo que en el Anexo I: Siglas Y Acrónimos, se encuentra la Tabla A1-1 que recoge todos ellos junto a su significado.

Previamente destacamos los siguientes por su importancia, relevancia y frecuencia con la que se menciona a lo largo del proyecto: **SLAM** de Simultaneous Localization and Mapping, Localización y Mapeo Simultáneos, es la tecnología que permite a un dispositivo localizarse y realizar mapa de su entorno; **LiDAR** del inglés Light Detection and Ranging, es una tecnología de teledetección que utiliza pulsos de luz láser para medir distancias y crear representaciones tridimensionales exactas del entorno; **ROS** del inglés Robot Operating System es un middleware, es decir, un conjunto de herramientas y librerías que corre normalmente con Linux; y *Navigation 2* (Nav2), que es el sistema que decide cómo llega al destino esquivando obstáculos el robot.

2.2 Papel en la enseñanza técnica

Los robots móviles educativos actúan como laboratorios portátiles que integran mecánica, electrónica, control, programación, comunicaciones y percepción con IA. Su valor reside en que conectan los conceptos teóricos con evidencia experimental inmediata, pues cada decisión de diseño o de software se traduce en un comportamiento observable como su trayectoria, estabilidad, consumo, precisión y seguridad. Esta trazabilidad fomenta la comprensión profunda y la transferencia de aprendizaje a sistemas más complejos empleados en la práctica profesional.

En el contexto de la Armada, dicha práctica profesional se centra en el uso operacional y táctico de vehículos no tripulados, terrestres, aéreos y marinos, en misiones de vigilancia, exploración y de inteligencia. Por ello se ha de resaltar la importancia de que los miembros que la componen, incluyendo la oficialidad, sean conocedores de que estas plataformas educativas permiten ensayar, de manera segura y controlada, principios y misiones que subyacen a los vehículos no tripulados en cuanto a vigilancia, guiado y evitación de obstáculos se refiere. El alumno adquiere con ellas hábitos de operación, verificación y seguridad extrapolables a entornos reales.

2.3 Plataformas Comerciales

Las plataformas comerciales de robótica educativa ofrecen accesibilidad, documentación sólida y soporte de comunidad, lo que reduce la curva de implantación en el aula y facilita la estandarización de

prácticas. Su selección debe responder a criterios de coste total de propiedad (TCO), robustez, seguridad, facilidad de mantenimiento, ecosistema de software (por ejemplo, compatibilidad con sistemas estándar empleados en robótica como Robot Operating System (ROS)) y potencial para integrar IA de forma didáctica. A continuación, se presenta un resumen de plataformas educativas comerciales orientadas a este tipo de formación técnica.

➤ *mBot / mBot2 (Makeblock)*

A esta plataforma se le asemeja un perfil de iniciación y nivel básico–intermedio, cuya orientación docente lo convierte en ideal para introducir sensorización, control reactivo y hábitos de verificación experimental en laboratorio, sirviendo como primer peldaño de un itinerario que luego escala hacia visión, SLAM y ROS2 con plataformas más avanzadas.

Esta plataforma tal y como se indica en su página oficial [1] se compone de un kit educativo con un coste de 218,50 euros, con chasis compacto y tracción diferencial que integra dos motores, sensor ultrasónico para distancias, sensores infrarrojo, IR para seguimiento de línea y control remoto, y una IMU/giroscopio (según versión). En su configuración de aula, el despliegue es rápido y sencillo, lo que favorece la adopción temprana por parte de estudiantes sin experiencia previa. El modelo corresponde al que se aprecia en la Figura 2-1.

Su Software se nutre del ecosistema mBlock que habilita programación por bloques y transición progresiva a texto. En niveles superiores, puede trabajarse en Arduino C (mCore) o Python (CyberPi), lo que facilita introducir conceptos de lectura de sensores, control de actuadores, temporización, manejo de interrupciones y depuración elemental.

Entre sus ventajas en el ámbito académico, nos proporciona un despliegue rápido en aula con ejemplos listos y buena estabilidad; es idóneo para el control reactivo y sensorización elemental; y posee una seguridad y robustez adecuada para prácticas repetitivas (velocidades moderadas) y componentes reemplazables. Sin embargo, estos modelos de plataformas están más orientados a un ámbito docente de educación primaria y secundaria pero no parecen ser adecuados para un ámbito universitario.

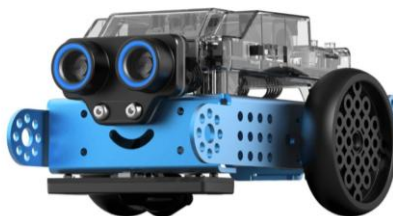


Figura 2-1 Imagen del mBOT [1]

➤ *Thymio II (Mobsya)*

Thymio II es un robot educativo open-source desarrollado por *L'École Polytechnique Fédérale de Lausanne* (EPFL), y producido por la asociación Mobsya. Está diseñado para introducir a estudiantes en programación, percepción y control con un conjunto amplio de sensores/actuadores e interfaces de programación progresivas. La plataforma se distribuye en versión cableada y Wireless, y cuenta con abundante documentación y materiales didácticos.

Entre sus características, según la página oficial de Thymio[2], destacamos sus dos motores de *Direct Current*, DC, independientes, sensores IR de proximidad, acelerómetro, micrófono y Light Emitting Diode (LED) Red Green Blue (RGB) para feedback. Se puede adquirir, por un precio aproximado de 170,0 euros, el modelo básico que se aprecia en la Figura 2-2.

Con respecto a su Software, con Thymio Suite se instala todo lo necesario para empezar, desde probar el robot/simulador hasta programarlo con bloques o texto, haciéndolo ideal para partir desde “cero” e ir ampliando los conocimientos de forma gradual. Además, es compatible con entornos de

programación como VPL (Visual Programming Language), Blockly, Aseba y Python. Estos entornos permiten programarlos en la realidad y/o virtualmente.

Es una plataforma que posee varias ventajas como ser robusto y accesible, ideal para aprendizaje activo y demostraciones en clase; es de Open-hardware/software con documentación pública y con un gran ecosistema educativo (recursos, simulador,...). Sin embargo, su odometría es modesta debido a sus sensores de serie, dando lugar a una percepción básica que no permite un SLAM fiable sin ayuda de elementos externos; y es está enfocado a estudiantes de educación secundaria.



Figura 2-2 Imagen Thymio II [2]

➤ *TurtleBot*

TurtleBot es la plataforma educativa y de investigación estándar en ROS/ROS 2, concebida para aprender y practicar navegación autónoma, percepción y SLAM con hardware abierto. De serie, según su página web oficial [3] ofrecen un chasis compacto y modular con LiDAR e IMU, ROS 2, LiDAR 2D, y una cámara estéreo OpenCV AI Kit Depth (OAK-D) con IA espacial, además de dock de carga para operaciones continuas. Contando con documentación extensa y ejemplos listos para usar en simulación y en robot real.

Con respecto a su software, incorpora de serie ROS/ROS 2 nativo permitiendo trabajar directamente con la navegación autónoma y SLAM creando mapas y moverse entre ellos. Existe, además, un ecosistema y un gran número de recursos tanto oficiales como de una comunidad docente (tutoriales, foros,...) que permite un mayor desarrollo y evolución en el conocimiento de los alumnos en el ámbito de la robótica.

Esta plataforma, el TurtleBot 4 que se aprecia en la Figura 2-3, destaca por su calidad e integración de los sensores, documentación y simulación integrada, lo cual la hace bastante adecuada para el ámbito docente universitario. Sus únicos inconvenientes son su elevado coste de adquisición, que es de aproximadamente de 3300 euros y su puesta a punto que precisa de un conocimiento avanzado en la materia.



Figura 2-3 Imagen del TurtleBot 4 [3]

2.4 Robots Open-Source

A continuación, trataremos los robots Open-Source, los cuales representan una categoría de robots bastante común en líneas de investigación, por motivos como su transparencia, al poder inspeccionar su stack de software completo, su comunidad y soporte, que facilitan al docente su aprendizaje gracias a comunidades activas (empresas, academias,...) y su capacidad modular que permite que tengan diferentes tipos y niveles de configuraciones en función de la complejidad que se quiera alcanzar.

Esto se logra finalmente gracias a que su software, y en ocasiones su hardware, se encuentra publicado con licencias permisivas facilitando dicho estudio. Con el fin de lograr una curva de aprendizaje escalable para el alumno, encontramos una gran variedad entre las plataformas disponibles pudiendo ser desde terrestres y aéreas hasta incluso submarinas, que pueden modificarse para el interés docente. En la Tabla 2-1 se muestra un listado de este tipo de robots mencionados en el proyecto.

Plataformas	Apartado en el que se desarrollan
TurtleBot 4	Plataformas Comerciales
Husarion ROSbot XL	Plataformas terrestres (UGV)
DonkeyCar	Plataformas terrestres (UGV)
Crazyflie 2.x (Bitcraze)	Drones educativos (UAV)
BlueROV2 (Blue Robotics)	Plataformas submarinas educativas (ROV/AUV)
Holybro X500 V2	Drones educativos (UAV)

Tabla 2-1 Ejemplos de Robots Open-Source mencionados y su ubicación el proyecto

2.5 Software Educativo

A la hora de tratar con software educativo, nos referimos a las aplicaciones informáticas diseñadas para facilitar la enseñanza y aprendizaje a la hora de presentar contenidos, poner en práctica funciones y habilidades, simular e incluso evaluar al alumno, como ROS u Open Source Autopilot (PX4), los cuales permiten programar robots en entornos estandarizados que incluyen la posibilidad de emplear herramientas existentes, como el visualizador 3D Rviz (ROS Visualization) mostrado en la Figura 2-4.

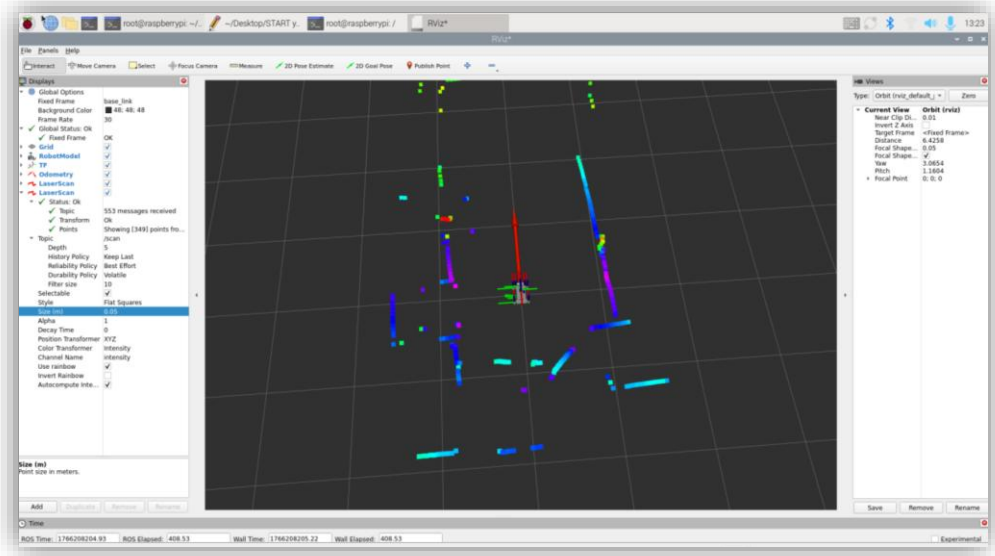


Figura 2-4 Programa RViz en funcionamiento, imagen propia

Entre estos nos encontramos con simuladores y laboratorios virtuales como Gazebo Sim (ROS 2) y Webots que recrean sistemas físicos para probar las funcionalidades de navegación autónoma sin necesidad de disponer del hardware, para lo cual se puede emplear programación en Python, en lenguaje C (para Arduino IDE, por ejemplo) o paquetes de software como Thymio Suite. Su empleo permite practicar técnicas como SLAM, el calibrado de actividades de percepción y control, la programación de la sensorización IR, la odometría, el piloto automático o incluso existen librerías que permiten gestionar la programación de misiones.

Este tipo de software se ha de caracterizar principalmente por su relevancia técnica en el aprendizaje, por su usabilidad y accesibilidad, y por permitir alcanzar objetivos y evaluar competencias. Entre los más comunes se destacan los de la Tabla 2-2.

Herramienta	Tipo	Propósito principal	Plataformas	Notas para docencia
Gazebo Sim	Simulador 3D	Simulación física y sensores; prácticas de SLAM y control	Linux, Windows, macOS	Alta fidelidad; mundos y plugins reutilizables
Webots	Simulador y IDE	Modelado y simulación con biblioteca de robots/sensores	Linux, Windows, macOS	Instalación sencilla; buena documentación y ejemplos
ROS 2	Framework de robótica	Middleware, herramientas y ecosistema de paquetes	Linux, Windows, macOS	Estándar de facto en investigación y docencia
PX4	Autopiloto y simulación	Control de vehículos y pruebas sin hardware	Linux, Windows, macOS	Misiones, telemetría, offboard con MAVSDK
RViz	Visualización 3D	Visualizar sensores, mapas y estado del robot	Linux, Windows, macOS	Esencial para percepción y navegación
Nav2	Navegación	Planificación, control, localización	Linux (principalmente)	Producción; configurable por plugins y BT
Arduino IDE	IDE	Programación de microcontroladores educativos	Linux, Windows, macOS	Aprendizaje de control y sensorización
Thymio Suite	Entorno educativo	Programación con VPL, Scratch, Python	Linux, Windows, macOS; tablet (vía móvil)	Incluye simulador y herramientas de aula

Tabla 2-2 Software educativos más empleados

2.6 Plataformas terrestres (UGV)

Con respecto a las plataformas terrestres *Unmanned Ground Vehicle* (UGV) disponibles para el ámbito docente, nos encontramos con que se pueden clasificar en base a diferentes perspectivas, como por su complejidad de funcionamiento, el tipo de sensores instalados o por capacidades específicas como velocidad o carga máxima. Sin embargo, a continuación, los clasificaremos en si son diseñados para exteriores, interiores o mixtos.

➤ *UGV outdoor y mixto*

Entre los robots con la capacidad de funcionar en el exterior, nos encontramos en primer lugar con dos modelos de “Clearpath”[4] el “Jackal”, mostrado a la izquierda en la Figura 2-5, de 18000 euros y el “Husky” mostrado a la derecha en la Figura 2-5, disponible a partir de los 23000 euros, ambos son UGV compactos estancos y con GPS e IMU e incorporan la integración de ROS/ROS 2. Se diferencian entre ellos en que el segundo modelo es más robusto para trabajos en exterior y es capaz de portar una mayor carga, además de ser una plataforma clásica en universidades y centros de investigación como en ORI Oxford.



Figura 2-5 Imagen del Clearpath Jackal (izquierda) y del Clearpath Husky (derecha) [4]

Otros modelos de UGV son los de la empresa española “Robotnik” [5] como el “RB-SUMMIT”, mostrado en la Figura 2-6, que posee una base 4x4 modular con stack ROS que incluyen paquetes de localización, navegación y simulación ejemplos como el “Robotnik RB-VOGUI” con mayor capacidad de carga y características similares además del kit de seguridad como el soporte del “2D safety lidar”. En ambos modelos existe la opción de incorporar un brazo UR y su coste supera los 30000 euros en ambos modelos.



Figura 2-6 Imagen del RB SUMMIT [5]

➤ *UGV indoor de alta movilidad*

En esta categoría destacamos al “Husarion ROSbot XL”, Figura 2-7, que según su página web oficial[6], es una plataforma modular 4x4 que incluye ROS 2 nativo con un ordenador de placa única a elegir entre Raspberry Pi o Jetson; además incluye sensores LiDAR y cámara de profundidad. Su precio base y más económico, aproximadamente de 4050 euros, aunque puede aumentar en función del ordenador de placa única que incorpore y de los componentes del kit.



Figura 2-7 Imagen del Husarion ROSbot XL [6]

➤ *UGV indoor diseñados para la enseñanza*

Por otra parte, debido al diseño del chasis y la capacidad modular, tenemos una categoría de robot terrestres indoor, de interior, mayormente enfocados a la enseñanza y de menor coste que los mencionados anteriormente. De esta clase destacaremos varios, mostrados en Tabla 2-3.

- El “Husarion ROSbot 3” un modelo 4x4 con RPLiDAR y cámara RGB-D, incluye ROS/ROS2 y es ideal para SLAM y Nav2 en un aula. Su precio, en función de su versión, es a partir de 2749€.
- El “AgileX LIMO” con la posibilidad de cuatro modos de tracción diferentes, LiDAR y soporte y reposicionamiento ROS 2. Su coste es de 2300€.
- El “Donkey Car” que es open-source, modular con Raspberry Pi, cámara, librería en Python, simulador y un coste de 250\$.
- El “Waveshare JetRacer ROS AI Kit” basado en Jetson Nano con cámara y opción de RPLiDAR A1 e IMU, paquete ROS y de un coste de 450€.
- El “ROSMaster A1 AI Large Model ROS2” una plataforma 4x4, con sensores LiDAR y cámara, percepción 3D, hace uso de IA y combina la navegación por SLAM. Además, tiene capacidad modular y existe varios kits de inicialización cuyo coste se encuentra entre los 369€.

Imagen	Plataforma	Sensores típicos	Puntos fuertes	Precio
	Husarion ROSbot 3	LiDAR, cámara de profundidad e IMU	Modular; paquetes de autonomía y manipulación. Construcción robusta	2749€
	AgileX LIMO	LiDAR EAI T-mini Pro, Cámara de profundidad Pantalla táctil y OLED	Ecosistema abierto, preparado para IA (YOLO) y construcción robusta	2300€
	Donkey Car (DIY)	Compatible con cualquier tipo de cámara, LiDAR e IMU	Hardware y Software abierto, infinidad de configuraciones y económico	250€
	JetRacer ROS AI Kit (Waveshare)	Cámara 8MP, LiDAR, IMU y audio Inteligente	Visión + SLAM en Jetson; demos y soporte; y económico	450€

	ROSMaster A1 AI Large Model ROS2	Cámara 2MP, IMU, T-mini LiDAR y módulo IA (YOLO11)	Comunidad viva, numerosos tutoriales disponibles y es modular	369€
---	--	---	--	------

Tabla 2-3 Comparativa de robots terrestres educativos (UGV)

2.7 Drones educativos (UAV)

En este apartado pasaremos de las plataformas terrestres a las aéreas con los drones educativos *Unmanned Aerial Vehicle* (UAV). Al igual que con el apartado anterior nos encontramos modelos con todo tipo de características y posibilidades, es por ello por lo que los clasificaremos en base a kits para una iniciación segura en aula y sencilla, y en kits complejos.

➤ UAVs de iniciación segura en aula

En este grupo se incluyen los kits y modelos más básicos y sencillos para que el alumno se inicie y desarrolle sus conocimientos sobre el mundo de drones UAV; los modelos que se van a mencionar a continuación son una muy buena opción desde el punto de vista económico, puesto que su coste no sobrepasa los 300€ y los recambios son fáciles de encontrar, sustituir y de bajo coste, y desde el punto de vista educativo por la cantidad de ejercicios y pruebas que se pueden llevar a cabo con ellos.

Comenzando por los modelos de la empresa *Da-Jiang Innovations* (DJI), “DJI Ryze Tello EDU” de 198€ [7] y “DJI RoboMaster TT” de 250€ [8] son idóneos para llevar a cabo prácticas de visión simple, realizar recorridos y organizar enjambres, ambos mostrados en Figura 2-8. Entre sus características técnicas, disponen del módulo ESP32, matriz LED 8x8, tecnología *Time of Flight* (ToF), etc. Además, existen recursos oficiales y manuales para el aprendizaje como “Education Hub” y se programan a través Python.



Figura 2-8 Imagen del DJI Ryze Tello EDU [7] (izquierda) y del DJI RoboMaster TT [8] (derecha)

Otro modelo práctico, con multitud de funciones y de bajo coste es el “Crazyflie 2.1 Brushless” de 619,99€ [9] un nano-dron de open-hardware, mostrado en la Figura 2-9. Con él es posible hacer pruebas y ejercicios de control, localización y robótica de enjambres. Trae consigo el software de ROS 2 y de Lighthouse (SteamVR) que permite al dron detectar objetos y reconocer el entorno para poder representarlo virtualmente.



Figura 2-9 Imagen del Crazyflie 2.1 Brushless [9]

➤ *UAVs y kits complejos*

Una vez adquiridos los conocimientos básicos de programación y montaje de UAVs educativos, para ascender un escalón más en este ámbito educativo, se recomienda adquirir plataformas más complejas. La más empleada es el dron “Holybro X500 V2”, mostrado en la Figura 2-10 con el kit “Pixhawk 6C (PX4 Dev Kit)”, ambos se pueden adquirir por 899,95€ [10]. Consiste en una plataforma cuadricóptero lista para funcionar con el PX4 (piloto automático), es de montaje rápido (alrededor de 30-40 minutos), posee cámara de profundidad, telemetría SiK (transmisión inalámbrica de datos) y admite Raspberry o Jetson.



Figura 2-10 Imagen del Holybro X500 V2 [10]

Por otra parte, también se tiene la posibilidad de emplear software para simular las tareas que un dron real puede llevar a cabo, pero en un entorno virtual, permitiendo así solucionar inconvenientes como el coste del hardware, pues es software gratuito, y el de precisar una zona exclusiva y delimitada para el vuelo del UAV con el fin de cumplir con medidas de seguridad a la hora de realizar prácticas con alumnado. El software en cuestión es el “ArduPilot” diseñado y pensado para una simulación académica y profesional.

2.8 Robots de superficie marítima (USV)

Pasamos a continuación al ámbito marítimo con plataformas diseñadas para ello las *Unmanned Surface Vehicule* (USV). Sus objetivos son similares a los de las anteriores plataformas ya mencionadas, el navegar de forma segura en base a los datos recogidos por sus propios sensores, detectar objetos e interpretarlos y poder realizar misiones o tareas que se les encomienden (como seguir una derrota a través de unos waypoints).

Para el ámbito profesional dedicado a la investigación y al ámbito docente nos encontramos con diferentes modelos que los clasificaremos en función de su complejidad y coste.

➤ *USV complejos y de coste elevado*

Se destacan los siguientes modelos, ambos mostrados en la Figura 2-11:

- El “BlueBoat” de Blue Robotics [11], consiste en un catamarán portátil modular con “BlueOS” (open-source) y “ArduRover” (ArduPilot) que permiten la navegación autónoma por waypoint, posee GPS, IMU integrada y RTH (return to home). Es un dispositivo mayormente diseñado para la hidro batimetría y la monitorización ambiental y con un coste de 4135€.
- El “Heron USV” de “Clearpath”[12], un USV portátil tipo catamarán preparado para ROS con GPS, LiDAR e IMU integrados. Es de uso profesional y también empleado en universidades y cursos, la configuración más básica es de 25000€.



Figura 2-11 Imagen del BlueBoat [11](izquierda) y del Heron USV [12] (derecha)

➤ *USV accesible y de menor coste*

Por otra parte, nos encontramos con los USV de hardware y software abierto, que proporciona una mayor capacidad modular y de menor coste. De esta forma se pueden diseñar para un desarrollo académico más específico que permite aprovechar al completo el material y la plataforma, permitiéndose así implementar cursos de inicialización para el ámbito docente de forma accesible y asequible. Lo más común según foros [13], es emplear como casco o plataforma una unidad radio control, RC acuática o imprimir en 3D uno desde cero, como el que se muestra en la Figura 2-12. A éste, se le comenzarían a añadir los sensores más interesantes para un entorno docente como un LiDAR, IMU, GPS y un ordenador (Raspberry Pi o JetsonNano) que coordine todos los datos.



Figura 2-12 Imagen de un USV de Hardware abierto [13]

2.9 Plataformas submarinas educativas (ROV/AUV)

Estas plataformas, *Autonomous Underwater Vehicle* (AUV), al funcionar bajo la superficie marítima precisan de mayores exigencias de simulación y seguridad. Sus principales objetivos pueden atender tanto al ámbito de la investigación, para estudiar la fauna marina o hacer un levantamiento hidrográfico del fondo marino, como al de la formación en universidades aprendiendo a planificar, ejecutar y analizar las misiones que se le encomienden. Entre las plataformas que se encuentran disponibles en el mercado encontramos mayoritariamente las de hardware abierto, esto se debe a que los

sensores que incorporan se han de montar en una plataforma estanca certificada para llegar incluso a los 300 metros de profundidad, en algunos casos.

Se destacan dos plataformas educativas, sin embargo, la primera será mencionada únicamente como antecedente puesto que fue descontinuada en el año 2020, ambas se muestran en la Figura 2-13.

- El “*OpenROV/Trident*” según el foro [14] fue un vehículo operado remotamente (ROV) diseñado para democratizar la exploración subacuática, comercializado como equipo cerrado, listo para operar enfocado a un uso educativo, científico y de inspección ligera. Los sensores que lo componen son un sensor de profundidad, sensor IMU, cámara integrada 1080p. Su coste nuevo de serie era de 1700€, en la actualidad se encuentra descatalogado.
- El “*BlueROV2*” de “*Blue Robotics*” [15] es un ROV modular, su firmware recomendado es el ArduSub para enseñanza, cuenta con IA ligera embarcada y entre sus sensores habituales (dependiendo del kit que se adquiriera) encontramos el sensor inercial IMU 9-DOF, *Degrees Of Freedom*, para el rumbo y la estabilización, barómetro interno y externo, detección de fugas, termómetros y cámara *High Definition*, HD de baja luz. Además, existen kits que incluyen un sónar de multihaz, altímetro a través de ecosonda para medir la altura sobre el fondo y más cámaras alternativas. Su coste de serie es de 3000€ y con kits puede alcanzar los 5000€.

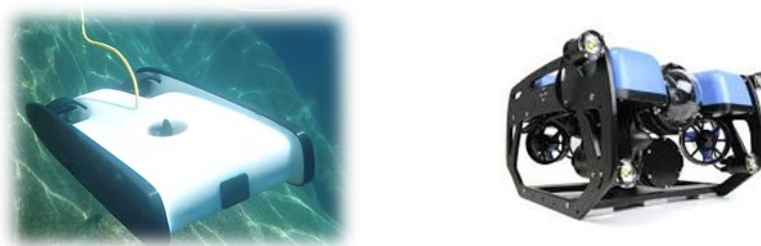


Figura 2-13 Imagen del OpenROV/Trident [14] (izquierda) y del BlueROV2 [15] (derecha)

2.10 Sensores en robótica móvil

2.10.1 Introducción de sensores en robótica móvil

Los sensores son los encargados de dotar a los robots de los sentidos para percibir, recibir información e interactuar con su entorno, equivalentes como, por ejemplo, al del oído y de la vista en los seres vivos. Esto lo logran por medio de transformar magnitudes físicas en señales eléctricas procesables e interpretables. La información que permiten recopilar es empleada por el robot para que éste halle la respuesta a dónde se encuentra, qué hay a su alrededor y cómo llegar a su destino.

2.10.2 Ultrasonidos

Estos sensores consisten en transductores acústicos que emiten un pulso de sonido por encima de los 20kHz, siendo aproximadamente 40kHz en aire y 1000kHz en agua. La información que nos proporciona viene dada por la medición del tiempo de vuelo del eco reflejado por un objeto que encuentre para estimar así la distancia. Esto es así porque la distancia es directamente proporcional al producto de la velocidad de la onda y el tiempo hasta recibir el eco dividido entre dos. En la Figura 2-14 obtenida de [16] apreciamos su funcionamiento de forma esquemática.

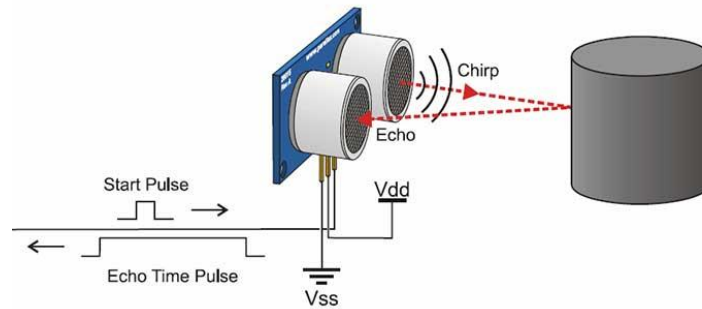


Figura 2-14 Esquema de funcionamiento del sensor HC SR04 [16]

En la robótica móvil este tipo de sensores es empleado para evitar obstáculos, medición de distancia permitiendo a las plataformas navegar con seguridad y proporcionar datos como estimación de altura en UAVs, proximidad a muelles a USVs y altímetros para que AUVs mantengan la altura sobre el fondo. Además, proporcionan una serie de ventajas puesto que son insensibles a la iluminación, funcionan correctamente frente al humo y al polvo, son de bajo consumo y coste; por otra parte, su rango es más corto que sensores como un LiDAR y un radar.

A continuación, se hará una recopilación de este tipo de sensores en las plataformas mencionadas anteriormente, mostrados también en la Tabla 2-4:

- Entre los UGVs tenemos al “Makeblock mBot2” que incorpora el “Me Ultrasonic Sensor” de 42kHz; el “TurtleBot3” no lo trae incorporado de serie, sin embargo, es compatible con el sensor “HC-SR04”, este mismo sensor también es compatible con el “Donkey Car” aunque no lo traiga de serie.
- De los UAVs mencionados el “Holybro X500 V2” aunque no lo tenga incorporado de serie, gracias a su capacidad modular, con el kit “Pixhawk (PX4)” es compatible con el sensor “MaxBotix MB1242” de 40kHz
- En los USVs, el “Blue Robotics” de “BlueBoat” tiene, como opción oficial, la posibilidad de instalar el “Ping2 Sonar” de 115kHz.
- En el ámbito de los AUVs y ROVs, la plataforma de “Blue Robotics” como el “BlueROV2” es compatible con kits que incorporan el sensor “Ping2” de 115kHz empleado como altímetro y el “Blue Robotics Ping360” de 750kHz como sonar de imagen.

Nombre del sensor	Imagen	Tipo de Plataformas	Plataformas que lo Instalan
Blue Robotics Ping360		USV y AUV	BlueROV2 y BlueBoat
MaxBotix MB1242		UAV	Holybro X500 V2
HC SR04		UGV	Makeblock mBot2, Donkey Car y TurtleBot3

Tabla 2-4 Sensores de ultrasonidos empleados en diferentes plataformas

2.10.3 Cámaras

Las cámaras en el mundo de la robótica son el sensor fotosensible más común, pues son capaces de convertir la luz, captada por el objetivo, en señales eléctricas formando una imagen digital. Esto proporciona una gran cantidad de ventajas para las plataformas que las incorporen pues les hacen dotar del sentido de la vista. Centrándonos en sus funciones clave, destacamos la ayuda que proporcionan a la percepción y navegación al ser capaces de detectar obstáculos en tiempo real y mapear entornos con técnicas SLAM. También cumplen la función de guía a la hora de manipular brazos robóticos localizando y colocando objetos. Y, además, si integramos la IA en sus procesadores, los robots son capaces de analizar e interpretar las imágenes obtenidas a través de las cámaras.

Combinando la IA con la cámara se adquiere la capacidad de visión artificial, que es una de las más importantes en el proceso de percepción del entorno. Principalmente se debe a que permiten dotar a las plataformas robóticas no solo la capacidad de ver, sino que también permiten analizar y comprender la información visual obtenida, pudiendo de esta forma ejecutar tareas como localizar obstáculos, e incluso identificar y diferenciar los objetos del entorno con el fin de tomar decisiones en base a dicha información obtenida. Su funcionamiento consiste en combinar elementos de hardware al emplear cámaras para la obtención de las imágenes, y módulos de IA junto con software empleando programas con algoritmos de visión artificial que las interpretan.

Con respecto a los sistemas de percepción visual más empleados, se incluyen diversos tipos de cámaras. Las cámaras 2D y 3D constituyen la base de muchos sistemas de reconstrucción espacial, permitiendo generar modelos tridimensionales del entorno. Entre las 3D se encuentran, las cámaras RGB-D que integran información de color con datos de profundidad, proporcionando una percepción estructurada y detallada del espacio; y las cámaras Time of Flight (ToF) que estiman distancias mediante la medición del tiempo de retorno de un pulso luminoso, lo que facilita una detección precisa de la geometría del entorno. Finalmente, la incorporación de cámaras con capacidades de procesamiento

basado en inteligencia artificial habilita el reconocimiento avanzado de patrones y soporta la toma de decisiones autónomas en robots móviles.

Entre las cámaras de las plataformas mencionadas con anterioridad, nos encontramos:

- En los UGVs tenemos varios modelos que incorporan ya una cámara digital de serie como el “TurtleBot 4” con la “Luxonis OAK-D”, Figura 2-15, de 12MP compatible con IA al igual que la cámara del “Husarion ROSbot 3 / XL”; el “Waveshare JetRacer” trae consigo el módulo de cámara “IMX219-160”, Figura 2-15, de 8MP. Por otra parte, tenemos otras plataformas modulares que no incluyen cámaras de serie en este caso, se recomienda instalarles una “3D RGB-D”.



Figura 2-15 Imagen de la cámara Luxonis OAK D [17] (izquierda) y de la cámara IMX219 160 [18] (derecha)

- En los UAVs, los modelos de “DJI” como el “Ryze Tello EDU” y el “RoboMaster TT” emplean la misma cámara, una de 5MP con vídeo HD 720p/30fs de FPV (First Person View); entre las plataformas modulares destacamos los modelos “Himax HM01B0” monocromática, Figura 2-16, y la “Intel RealSense D435” una cámara de profundidad (3D). Con respecto a la visión artificial, drones de “DJI” incorporan el controlador abierto “ESP32” y el software “SDK”(Software Development Kit), aunque es una visión artificial muy básica. Al “Crazyflie 2.1” se le puede incorpora de forma opcional el “AI deck 1.1”, Figura 2-16, con “GAP8”.



Figura 2-16 Imagen de la cámara Himax HM01B0 [19] (izquierda) y del AI deck 1.1 [20] (derecha)

- De los USVs mencionados de hardware cerrado ninguno trae cámaras de serie, por otra parte, los de hardware abierto se le puede instalar cualquier modelo de cámara de este apartado siempre y cuando se logre la estanqueidad necesaria para su funcionamiento en el entorno marítimo. Con respecto a la visión artificial, el “Clearpath Heron USV” tiene esta capacidad, aunque muy básica y se desconoce qué tipo de hardware le permite tener dicha capacidad.
- Entre los AUVs y ROVs destacamos los modelos de “SONY” “IMX322” e “IMX323” de 1080p gran angular que puede incorporar el “BlueROV2”

2.10.4 Sensores IR

En robótica móvil otro de los sensores más empleados son los sensores infrarrojos, que en lugar de emplear la luz visible (400-790THz) hace uso de la infrarroja (0’3-400THz). Sabiendo ya en qué rango

de frecuencias trabajan estos sensores, sabemos que son capaces de detectar objetos que radien o reflejen ondas a dichas frecuencias (en función de si el sensor es pasivo o activo) como se aprecia en la Figura 2-17.

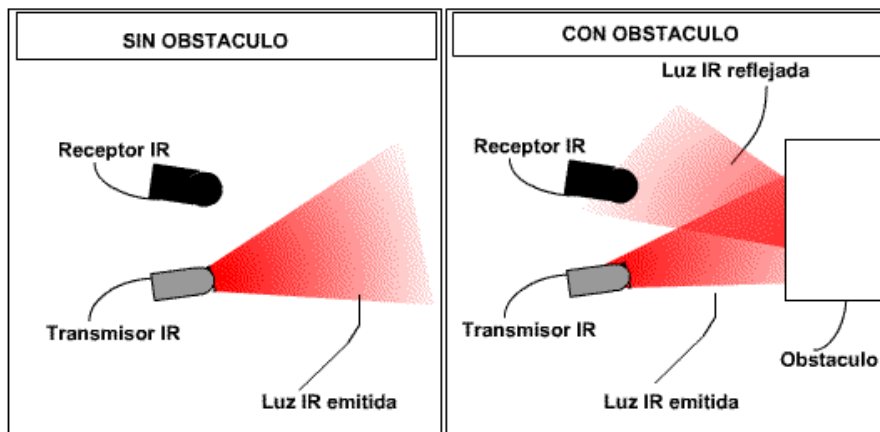


Figura 2-17 Imagen de explicación de funcionamiento de sensor IR [21]

Su función principal es la de medir distancias y detectar objetos u obstáculos para alertar de su proximidad y evitar colisiones, también pueden detectar líneas negras sobre fondo blanco o viceversa, una función muy útil si se quiere que el robot siga una trayectoria en concreto.

Los modelos de sensores IR más comunes son:

- En los UGVs, el “Makeblock mBot” trae de serie el sensor “Me Line Follower”, Figura 2-18, para el seguimiento de línea, el “Thymio II” incorpora de serie nueve sensores (cinco frontales, dos traseros y dos inferiores) que funcionan por emisión y recepción IR. También, son empleados el “IR VL53L0X” y el “IR cliff sensor (IRSS-10)” en los kits para añadirlos a las plataformas.
- En los UAVs, estos sensores suelen pertenecer al sistema de visión inferior para estimar la altura de vuelo, como el módulo ToF. Los modelos de “DJI” integran junto a la cámara un módulo “IR 3D”, en modelos de hardware abierto como el “Crazyflie 2.1” es posible añadirle el “Z-Ranger Deck v2”, Figura 2-18, con ToF e “IR VL53L1x”.
- En los USVs, no es común que de serie los incorporen pues estos sensores tienen un alcance limitado en el agua. Sin embargo, en plataformas modulares si fuese preciso se les podría instalar sensores IR de los ya mencionados.
- En los AUVs/ROVs ocurre la misma situación que con los USVs, no es habitual que los incorporen de serie.

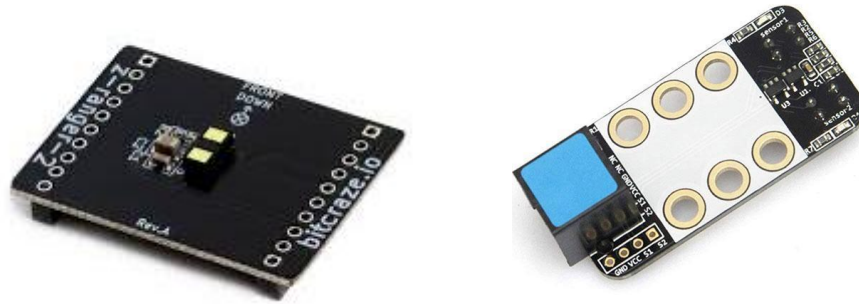


Figura 2-18 Imagen del sensor Z Ranger Deck v2 [22] (izquierda) y del sensor Me Line Follower [23] (derecha)

2.10.5 Unidad de medición inercial

Una unidad de medición inercial, conocida por sus siglas en inglés IMU “*Inertial Measurement Unit*”, consiste en una combinación de acelerómetros o giroscopos que por su finalidad y utilidad se ha convertido en un sensor muy distendido en ámbitos diferentes como el de los teléfonos móviles, aeronáutica, y la robótica móvil.

Su funcionamiento consiste en la recopilación y fusión de datos procedentes de los acelerómetros, giroscopos y magnetómetros que se integran en la IMU. Con esos datos de balanceo, cabeceo y guiñada es posible estimar la orientación y movimiento de la plataforma. Esto permite, en robótica, el determinar su propia posición sin ayudas externas como el GPS, logrando así una navegación autónoma, mantener la estabilidad y control de la trayectoria deseada, proceso que puede verse esquematizado en la Figura 2-19.

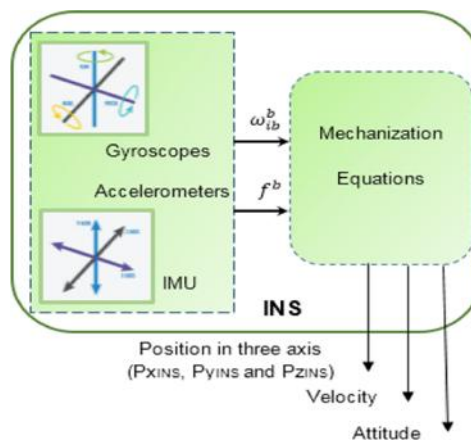


Figura 2-19 Esquema IMU [24]

Entre las plataformas analizadas nos encontramos con las siguientes IMU:

- En los UGVs, el “TurtleBot 4” emplea el “IMU 6-DOF” que mide las aceleraciones en tres dimensiones al igual las variaciones angulares; los modelos de “Clearpath” “Jackal” y “Husky” incorporan el “MicroStrain 3DM-GX5”; el “Husarion ROSbot 3” posee el “BNO055” de 9-DOF que le proporciona una orientación absoluta; el “ROSMaster A1” incorpora el “MPU-9250” de 9-DOF mostrado en la Figura 2-20; y en el “DonkeyCAR” se le puede instalar el “MPU-6050” o el “MPU-9250”.
- En los AUVs, los modelos de “DJI” no especifican el modelo de su sensor. Por otra parte, el “Crazyflie 2.1” incorpora el “Bosch BMI088” de 6-DOF y el “Holybro X500 V2 (PX4)” incorpora el “ICM-42688-P” mostrado en la Figura 2-20 y el “BMI088”.

- En los USVs, el “Blue Robotics BlueBoat” incorpora una IMU de 6-DOF el “ICM-20602”.
- En los AUVs/ROVs, se desconoce qué modelo de IMU incorporan las plataformas anteriormente mencionadas, pero por lo general incorporan una IMU de 6-DOF.



Figura 2-20 Imagen del MPU 9250 [25] (izquierda) y del ICM 42688 P [26] (derecha)

2.10.6 Sistemas LiDAR

Este sistema cuyo nombre procede del inglés “*Light Detection And Ranging*”, permite medir distancias, escanear superficies, detectar objetos y mapear el entorno con gran exactitud mediante el uso de pulsos de luz láser [27].

Su funcionamiento se basa en el empleo de la luz del láser que emite. Éste se refleja en los objetos próximos y midiendo el tiempo que transcurre entre la emisión y el retorno, es capaz de calcular la distancia recorrida, mostrado en la Figura 2-21. Los LiDAR que nos encontramos en robótica pueden ser 2D (captura datos solo en un plano) o 3D generando una especie de “cúpula” a su alrededor pudiendo detectar todo lo que entre en ese volumen.

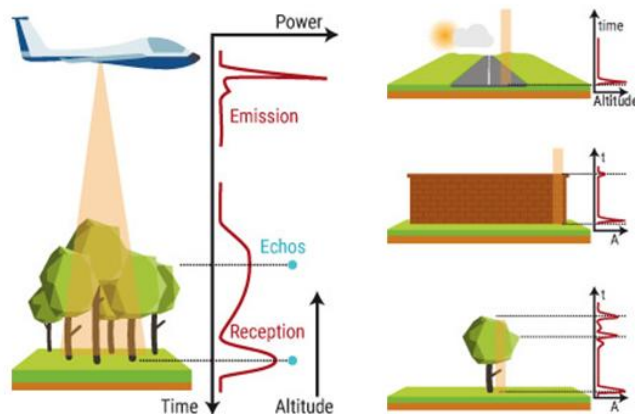


Figura 2-21 Explicación gráfica de funcionamiento de un sistema LiDAR [27]

Los LiDAR más comunes instalados en plataformas de robótica educativa son:

- En los UGVs, el “TurtleBot 4” incorpora el “RPLiDAR A1” apreciado en la Figura 2-22 de 2D; el “Husarion ROSbot” según la versión trae el “RPLiDAR C1/S2” que son 2D y el “ROSMASER A1” dependiendo de los kits, puede incorporar el “D300 DTOF LiDAR LD19P”.
- En los UAVs, no es común que incorporen uno de serie, sin embargo, a los que sean de hardware abierto se le puede instalar un sistema LiDAR de los ya mencionados.

- En los USVs tampoco incluyen sistemas LiDAR de serie, aunque se puede instalar sin ninguna complicación si es de hardware abierto.
- En los AUVs/ROVs, debido a su entorno de funcionamiento, no traen incorporados sistemas LiDAR.



Figura 2-22 Imagen del RPLiDAR A1[28]

2.11 Algoritmos de navegación

2.11.1 Control reactivo

El control reactivo en robótica es un enfoque de navegación y control en el que el robot genera órdenes de movimiento al interpretar al instante los datos captados por sus sensores, sin construir ni consultar previamente un mapa global de la situación. Es decir, que, para lograr su objetivo en lugar de acceder a una base de datos o un mapa global, el robot lleva a cabo sus acciones por el método de prueba y error hasta lograr su objetivo.

Centrándonos en su funcionamiento, tenemos que solamente se nutre de la información cercana y reciente, sin razonar el entorno general. A través de sensores IR, ultrasonidos o una nube de LiDAR 2D el robot obtiene información de su entorno como distancias a obstáculos, rumbo al objetivo, etc. A continuación, evalúa combinando reglas y funciones de coste y actúa seleccionando el rumbo y velocidad que más puntuación ha obtenido.

Entre las ventajas de algoritmo de navegación se destacan su respuesta rápida, simplicidad de implementación y ajuste, y requiere menos procesamiento que el SLAM. Por otra parte, no puede garantizar llegar al objetivo si el entorno exige decisiones a largo plazo. Se trata, por tanto, del modo de control en un ámbito local, en donde el objetivo principal es avanzar hacia el objetivo evitando obstáculos, pero sin una planificación previa por el conocimiento del entorno.

Los algoritmos reactivos más empleados y comunes en plataformas robóticas son el de campos potenciales (Artificial Potential Fields, APF) que consiste en que el objetivo atrae al robot y los obstáculos lo repelen; el *Dynamic Window Approach* (DWA), y el *Dynamic Window Local Planner* (DWB), en ROS2 NAV2 empleado por el “TurtleBot 4” y “TEB” (Timed Elastic Band) empleado por el “Husarion ROSbot 3”.

2.11.2 SLAM y mapeado

La localización y mapeado simultáneo, conocido por sus siglas en inglés SLAM, es la solución algorítmica por la que un robot es capaz de ubicarse (estimando su posición) a la vez que construye un mapa del entorno desconocido empleando sus propios sensores, mostrado en la Figura 2-23.

Es un método más complejo que el control reactivo principalmente por su dependencia mutua de un mapa, es decir, que el robot no se posicionará correctamente hasta que no haya generado él mismo con sus sensores dicho mapa. Además, a lo anterior se le ha de sumar la deriva y aumentos de errores debido al ruido e incluso por los errores que pueden arrastrar los propios sensores. Es por ello, que, aunque el

robot sea capaz de finalizar el mapeado, se ha de tener en cuenta que éste puede estar deformado y que cuando el robot navegue empleándolo no se posicionará correctamente.

En las plataformas que se han analizado, encontramos modelos que también son capaces de realizar mapeados y posicionarse por SLAM como el “TurtleBot 4” a través del “SLAM 2D LiDAR” junto a NAV2 al igual que los modelos de “Clearpath” y muchos más modelos de plataformas.

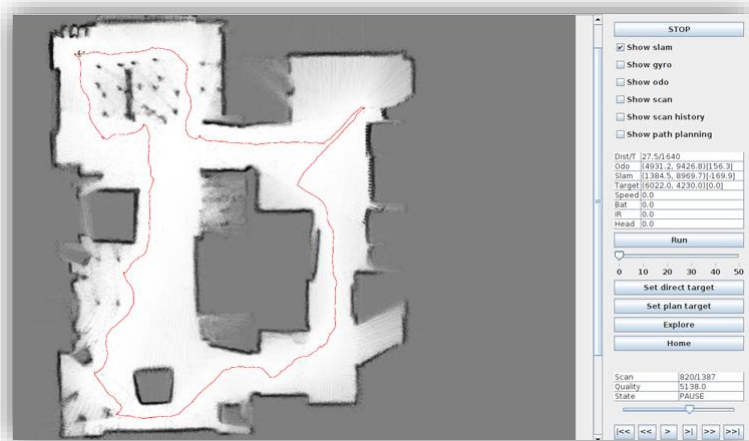


Figura 2-23 Imagen del recorrido realizado y del mapa generado mediante una técnica de SLAM [29]

2.12 Uso de IA en robótica móvil

La integración de Inteligencia Artificial, IA, en la robótica móvil es clave para dotar de percepciones ricas, toma de decisiones y navegación robusta en diferentes entornos. Por medio del empleo de algoritmos, los robots son capaces de adaptarse a condiciones cambiantes y operar en escenarios dinámicos gracias a una toma de decisiones autónoma influenciada por la IA. Esto ha permitido ampliar, cada vez más, la presencia y uso de este tipo de unidades en ámbitos como la industria, la logística, la ciencia, asistencia y educación. Entre las funciones desarrolladas por robots móviles gracias al empleo de IA, en concreto modelos ligeros de IA, se encuentran las de seguimiento, clasificación y localización de objetos, así como la implementación de técnicas de navegación avanzada y SLAM.

Con respecto a los modelos ligeros de IA mencionados anteriormente, éstos son redes neurales pequeñas y rápidas entrenadas con IA capaces, por ejemplo, para el caso de información visual, de interpretar imágenes del entorno captadas por la cámara del robot pudiendo distinguir entre obstáculos, señales de tráfico, personas, etc., tal y como se muestra en la Figura 2-24. La tarea de la IA es la de convertir los píxeles captados en información útil y reaccionar en base a esta interpretación del estado del entorno.

Desde el punto de vista operativo, los modelos ligeros como las variantes de YOLO se caracterizan por su baja latencia, consumo y huella de memoria contenida, suficientes para ejecutar percepción y apoyo a la navegación en tiempo real; comparados con modelos grandes (YOLO de tamaños L y X, por ejemplo) que requieren más *Graphics Processing Unit*, GPU y memoria, aunque en general son más precisos. En esta arquitectura, la cámara junto con el modelo ligero de visión cumpliría el papel del “sentido de la vista”; el control reactivo (como por ejemplo empleando el software Nav2 con el planificador local DWB) actúa como los “reflejos” para evitar obstáculos y estabilizar la trayectoria; y la planificación global de Nav2 aporta el componente “deliberativo”, es decir, la parte de planificación de la trayectoria asociada a una visión global del escenario y la misión a cumplir. La inferencia se ejecuta, normalmente, mediante un “runtime” optimizado para una librería del tipo TensorFlow Lite u *Open Neural Network Exchange*, (ONNX), sobre la computadora embarcada (por ejemplo, Raspberry Pi 5).

En plataformas como el ROSMASTER A1 se emplea un modelo ligero de la familia YOLO para la detección de objetos, priorizando la respuesta en tiempo real con recursos limitados.

Por otra parte, que una plataforma no traiga de serie un modelo ligero no significa que no sea compatible con el uso de IA de visión, sino que el fabricante no la impone. Los usuarios en estos casos suelen elegir instalar módulos ligeros de IA compatibles con “YOLO-tiny”, “YOLOv8-n”, “MobileNet-SSD” y “EfficientNet-Lite” por su baja latencia.

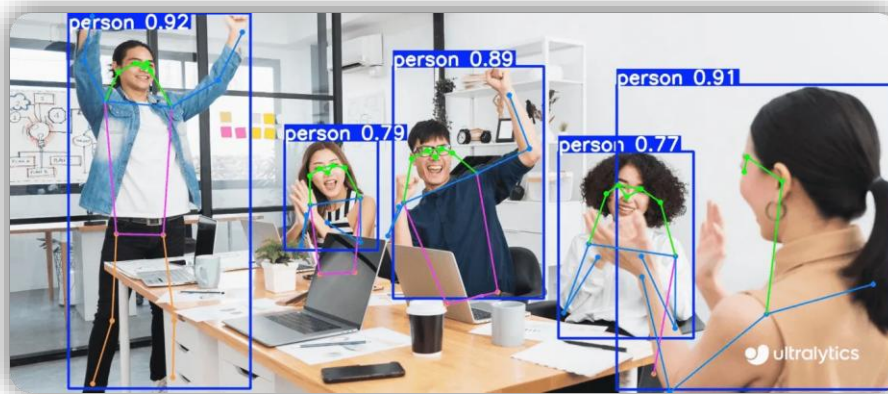


Figura 2-24 Imagen de YOLO analizando poses de personas [30]

2.13 Relevancia de estas plataformas en la formación militar moderna

Actualmente, en plena mitad de la segunda década del siglo XXI, por cada día que transcurre nos encontramos en la cúspide incesante del crecimiento de las nuevas tecnologías, un crecimiento que se podría asemejar al de una curva exponencial. Debido a esta rápida evolución, significa que lo que el alumno aprenda durante sus años lectivos sobre las nuevas tecnologías, muy probablemente haya cambiado o evolucionado una vez finalizado ese periodo.

De la misma manera, teniendo como referencia la historia ya pasada, no podemos pasar por alto la relación directamente proporcional del bienestar, poder hegemónico y económico de una potencia u estado con la inversión, el avance y desarrollo de la tecnología; tomando como ejemplos al antiguo Imperio Español en los siglos XVI y XVII al revolucionar la navegación marítima y transoceánica; o a los Estados Unidos de América siendo la cuna o financiando nuevos proyectos tecnológicos desde comienzos del siglo XX hasta el presente. Es por ello por lo que, tanto por motivos históricos, intereses económicos como prácticos, un alumno perteneciente a un centro de formación militar ha de estar obligado a conocer, comprender, entender y emplear las tecnologías de su actualidad con el fin de asentar las bases necesarias para afrontar a las venideras.

Además, para ratificar la relevancia del uso de estas plataformas en la formación militar, la gran mayoría de inventos y desarrollos tecnológicos en sus orígenes han sido impulsados por y para el ámbito militar, actualmente el invento más mencionado y relevante por sus ventajas y desventajas tácticas y operacionales que implica poseerlo o no, son las plataformas *Unmanned Vehicle* (UXV), mostradas en la Figura 2-25.

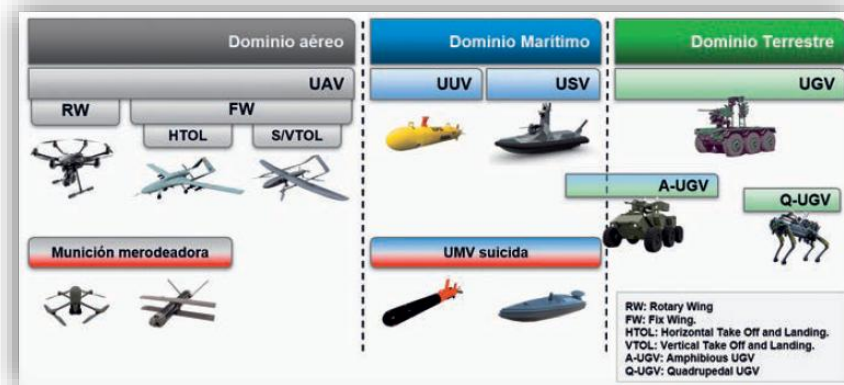


Figura 2-25 Imagen de las plataformas UXVs en el ámbito militar [31]

Por ello es por lo que, para emplearlas o enfrentarse a ellas, un militar ha de conocer sus bases y principios de funcionamiento como la navegación y fusión sensorial de dichas plataformas para luego explotarlas al máximo en diferentes misiones. Entre sus ámbitos de operaciones se destacan las misiones de protección de instalaciones, vigilancia, movilidad en entornos en los que el GPS esté negado, verificación de objetivos, saturación de defensas empleando enjambres, etc.

2.14 Plataforma elegida para el proyecto

Para llevar a cabo nuestro estudio y contestar a preguntas como, ¿qué ha de aprender un futuro oficial de la Armada sobre robótica móvil? o ¿qué conocimientos de robótica móvil se pueden aplicar al ámbito militar?, se analizarán que plataformas robóticas educativas serían las idóneas para ello, teniendo en cuenta una serie de criterios entre los que se incluyen: el coste, la complejidad, tiempo, ayudas al aprendizaje, seguridad y practicidad.

En primer lugar, se atenderá al coste de la plataforma, repuestos y kits, no superando los 400 euros.. Además, es importante tener en cuenta el precio de repuestos, ya que al ser empleados para la enseñanza y utilizados por personal que probablemente carezca de experiencia y conocimientos previos de cómo emplearlos, cabe la posibilidad de que quede inutilizado algún componente. Por otra parte, si se prefiere fomentar el empleo de plataformas modulares de gran compatibilidad se pueden adquirir kits concretos para así no tener material de más que no vaya a emplearse, reduciendo así su coste.

En segundo lugar, se prevalecerá a plataformas que no conlleven un nivel de dificultad muy elevado, puesto que se pretende que alumnos con unos conocimientos básicos sean capaces de, en un corto periodo de tiempo, de comprender el uso y funcionamiento de los sensores, cómo fusionar la información obtenida, cómo el robot la procesa, y finalmente ser capaces de ejecutar tareas y misiones con cierta complejidad en el entorno de trabajo. Además, el emplear una plataforma con un ecosistema desarrollado y completo, con una plataforma oficial en la que se encuentren tutoriales y una comunidad viva, dota al alumno de ayudas y medios que favorecen su aprendizaje.

Por último, la plataforma deberá cumplir con un criterio de seguridad, de forma que no se ha de permitir dejar de lado una serie de precauciones y medidas de seguridad, específicamente teniendo en cuenta que se manipulan baterías de litio, herramientas punzantes como destornilladores. Por otra parte, se hace necesaria una zona delimitada con redes o mamparos de metacrilato en caso de querer operar e integrar el trabajo con UAVs. Existen además otros criterios de practicidad, de forma que en principio cuantos más sensores incluya que sean útiles para la formación del alumno y sea más sencillo de mantener y de operar en entornos menos restrictivos mejor será la plataforma.

En base a los criterios de elección desarrollados, concluimos que la opción más equilibrada entre las plataformas analizadas es el “ROSMaster A1”. El costo del modelo estándar es de 370 euros, incorpora de serie sensores como IMU, LiDAR, y cámara de vídeo; además de incluir una Raspberry Pi 5 de 8GB y operan con IA. Cuenta también con una comunidad viva y creciente y una página oficial con un gran número de videos y tutoriales disponibles. Por otra parte, se trata de una plataforma modular de cuatro ruedas con tracción en el eje trasero y dirección en el delantero (modelo Ackermann estándar), perfecta para el funcionamiento en interiores como un aula de clase o un laboratorio en incluso en exteriores si se dan unas condiciones meteorológicas favorables.

3 DESARROLLO DEL TFG

3.1 Dimensiones y características

Nuestra plataforma para estudiar se trata, específicamente, del ROSMASTER A1 AI Large Model ROS2 Robot con Ackermann chasis y YahboomSKU: 300020088, mostrada en la Figura 3-1. Ésta ha sido escogida principalmente por su capacidad modular, la cual le permite instalar sensores específicos que no son incluidos en el kit de serie; y, por otra parte, por su entorno y ecosistema operativo de yahboom, que se encuentran numerosos tutoriales y foros que facilitan un aprendizaje y conocimiento de la materia de una manera notoria y en corto plazo.

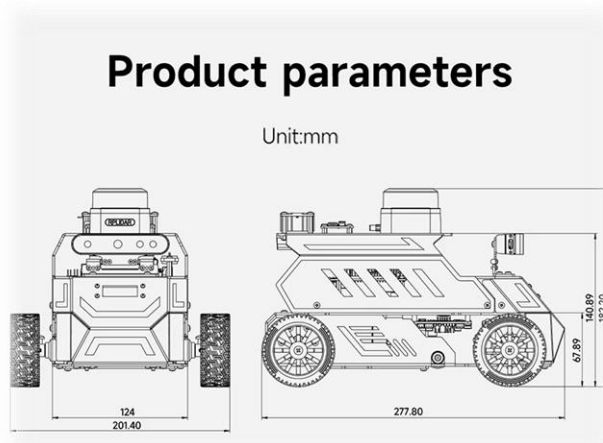


Figura 3-1 Imagen de medidas ROSMASTER A1 [32]

Sus características se han obtenido de la página web oficial de Yahboom [32]. Se trata de un robot cuyas dimensiones son de 27,78cm x182,2cm x 20,14cm. Su chasis está hecho de una aleación de aluminio, que no es estanco. Además, tiene tracción solo en sus dos ruedas traseras a través de un motor JGB 37-520, por cada rueda, de alto par metálico. Por otra parte, sus dos ruedas delanteras si bien no tienen posibilidad de tracción, permiten cambiar el rumbo del robot creando una dirección del tipo “Ackermann”, como se aprecia en la Figura 3-2 obtenida de [33]. Ambas, tracción y dirección, son controladas por el “ROS robot control board”. Su peso es de 2,5 kilogramos y es capaz de alcanzar 1,6 metros por segundo, según fuentes oficiales. Además, posee una batería de 12,6V (2A) que le dota de una autonomía de 5 horas de uso constante.

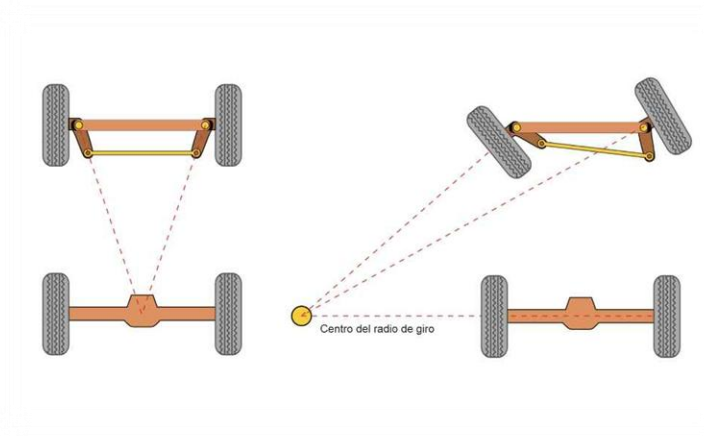


Figura 3-2 Imagen de ejemplo de dirección Ackermann [33]

3.2 Hardware

3.2.1 Componentes estructurales

En cuanto a los elementos estructurales de la plataforma nos encontramos al chasis que lo constituyen cuatro piezas de chapa de aleación de aluminio plegadas y unidas entre sí por tornillos, mostrado en la Figura 3-3. Además, éstas tienen agujeros roscados ya mecanizados para instalar los sensores que se incluyen en el kit y también para los de kits más profesionales. Las piezas son: la base de la plataforma, la tapa de acceso a la batería en la base, la parte delantera del robot y una superestructura que se coloca encima de la base.

Otros componentes son sus cuatro ruedas con llantas y neumáticos de goma blanda con un patrón de banda de rodadura que mejora su tracción. Los tornillos empleados son de métrica M3x6mm, M3x8mm, M2.5x5mm y los separadores y pilares de cobre M2.5 y M3.



Figura 3-3 Chasis del ROSMASTER A1, imagen propia

3.2.2 Componentes electrónicos

Continuando con el análisis, nos enfocaremos en el cableado del robot. Encontramos cables de alimentación, para mandar señales a los dos servos de la cámara, para el paso de información como USB a USB, USB a tipo C y tipo C, además del adaptador del LiDAR y control de los motores y de la dirección.

Entre las diferentes placas que incorpora el robot, tenemos al computador de placa única que es la Raspberry Pi 5 de 8GB, Figura 3-4, que actúa como cerebro de éste y realiza los procesos de comunicación inalámbrica y coordinación de tareas. Se encuentra la “Yahboom ROS Robot Control Board V3.0”, Figura 3-4, que es una placa de expansión, encargada de ejecutar las órdenes físicamente y gestionar telemetría, e incluye un IMU de nueve ejes (giroscopio, acelerómetro y magnetómetro).

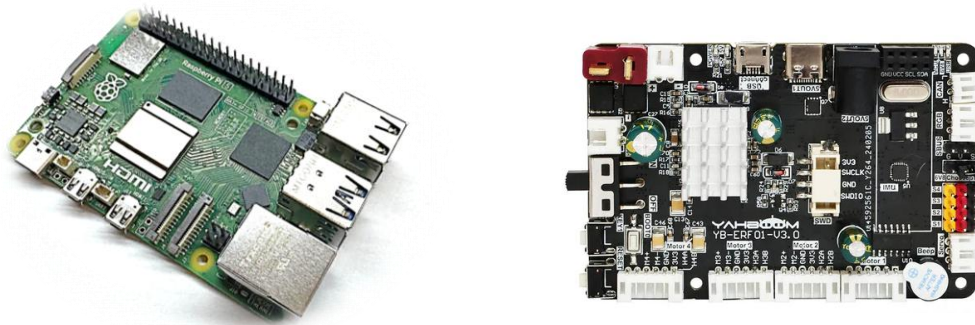


Figura 3-4 Raspberry Pi 5 , imagen propia (izquierda) y Yahboom ROS Robot Control Board V3.0 [32] (derecha)

Otros componentes son la placa de expansión USB 3.0, Figura 3-5, cuya función es la de dotar de puertos USB a la placa principal (en nuestro caso la Raspberry Pi 5) solucionando así la limitación física para conectar el hardware a dicha placa. También incluye un módulo de voz por IA, Figura 3-5, con funciones avanzadas de reconocimiento de voz, complementado por micrófonos y altavoz; se trata de un procesador de señales digitales optimizado para audio que integra algoritmos de IA (Edge AI) capaz de procesar la voz localmente, y se comunica con la Raspberry Pi a través de un nodo de ROS2.



Figura 3-5 Placa de expansión USB 3.0 (izquierda) y de módulo de IA (derecha), imagen propia

3.2.3 Sensores

Entre los sensores que permiten al robot que lleve a cabo la tarea de percepción del entorno para poder ejecutar su navegación autónoma, es decir que, si hacemos un símil, equivaldrían a los propios ojos del robot, tenemos que el robot incorpora un cámara y LiDAR.

En primer lugar, tenemos al “T-mini Plus LiDAR”, Figura 3-6, que se incorpora de serie en los kits más básicos. Consiste en un sensor de rango 2D que mide distancias desde los 5cm a los 12 metros en un ángulo de visión de 360°, según el manual oficial. Es capaz de funcionar en interiores con mucha iluminación porque soporta hasta los 60kLux. Se trata del sensor principal que se emplea para generar mapas en tiempo real y que le dota, al robot, de la capacidad de evitar obstáculos de forma autónoma.

En segundo lugar, se encuentra la cámara de 2MP, Figura 3-6, referida comercialmente como cámara de 200W. Se encuentra integrada en un soporte con dos servos digitales con la capacidad de rotación vertical (máximo 100°) y horizontal (máximo 180°). La lente que posee es de gran angular de 120° y se conecta por USB a la Raspberry Pi. Es imprescindible para las capacidades de visión artificial del robot en tareas de reconocimiento y seguimiento, conducción autónoma siguiendo carriles e interacción dinámica para búsqueda de objetos.

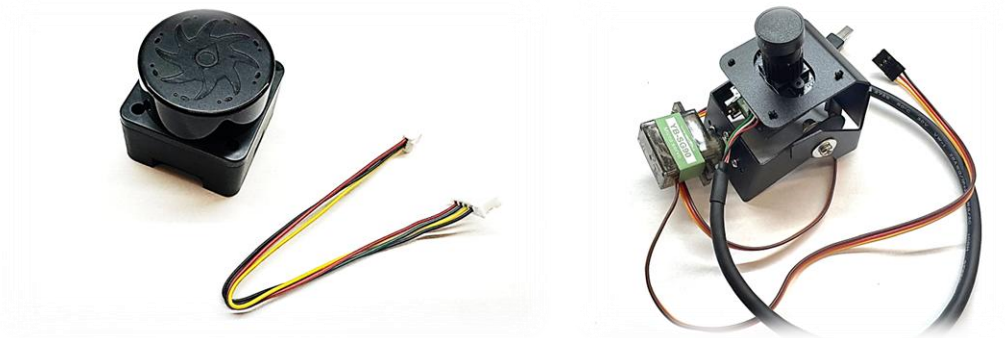


Figura 3-6 Sensor T-mini LiDAR (izquierda) y de la cámara 2MP (derecha), imagen propia

3.3 Ensamblaje

3.3.1 Estado inicial

Una vez que abrimos el embalaje de la plataforma “ROSMASTER A1 AI Large Model ROS2 Robot with Ackermann steering chassis” nos encontramos con ésta desmontada. Por una parte, está el chasis ensamblado junto a las cuatro ruedas, los dos motores, la pantalla *Liquid Crystal Display*, LCD y la placa “Yahboom ROS Robot Control Board V3.0”; y por otra, se encuentran el resto de los componentes electrónicos y sensores en sus envoltorios propios junto a dos destornilladores, paquetes pequeños enumerados para el montaje de los módulos, un mando RC, cableado, la batería, adaptadores para puertos (USB y Tipo C) y una memoria microSD de 128GB, como se muestra en la Figura 3-7.



Figura 3-7 Kit básico del ROSMASTER A1 fuera de su embalaje, imagen propia

3.3.2 Primera fase

A continuación, se procederá a comenzar el montaje de la plataforma, éste será dividido en cuatro fases. En la primera, el objetivo es el de instalar la batería en la base de la plataforma y el cable que conectará la placa de control (ya montada) con la Raspberry Pi. Los elementos y componentes necesarios son los que se muestran en la siguiente imagen, Figura 3-8.

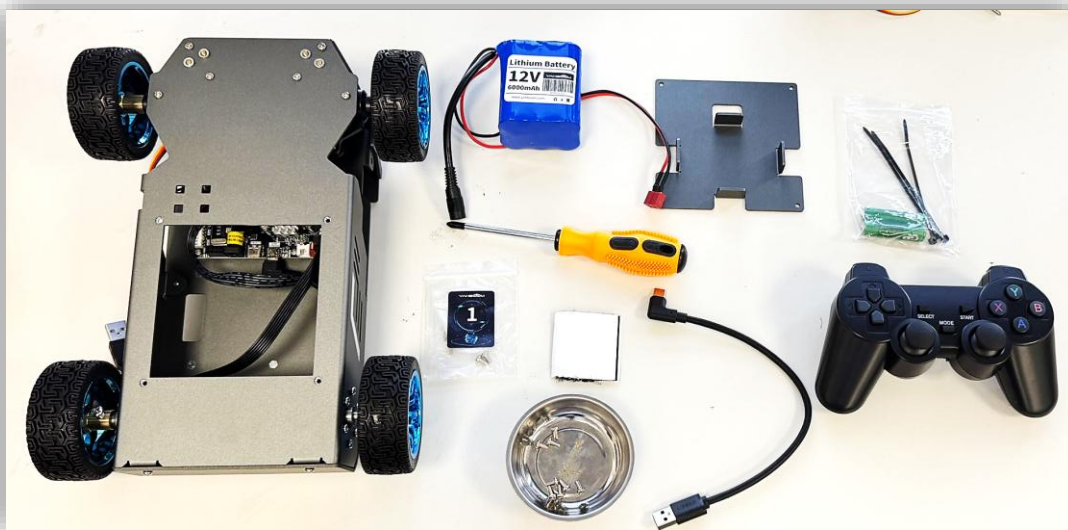


Figura 3-8 Componentes del montaje de la 1º fase, imagen propia

Material que emplear:

- Base de la plataforma
- Batería de litio de 12V y 6000mAh
- Destornillador
- Velcro
- Dos bridas
- Dos baterías AAA
- Mando RC y su adaptador Bluetooth
- Paquete de tornillos N°1
- Cable tipo C

El proceso comienza desmontando la superestructura y la parte delantera de chasis de la base de la plataforma desatornillando un total de ocho tornillos. Luego se desmonta la tapa inferior para introducir la batería a la que se ha de pegar el velcro en el lateral que haga contacto con la tapa. Se procede a conectar la batería con la placa de control a través de corriente de ésta y el cable hembra del cargador se hace fijo por medio de las dos bridas al chasis. Se conecta el cable tipo C a la placa de control y se coloca pasando al nivel superior mediante un orificio ya mecanizado en la chapa de la base. Por último, se cierra el acceso al compartimento de la batería, mostrado en la Figura 3-9.

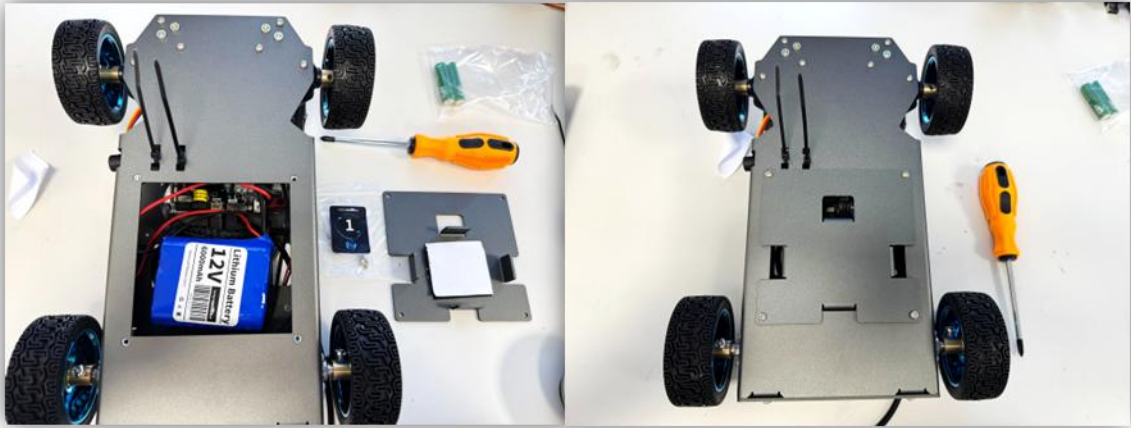


Figura 3-9 Interior de la base del ROSMASTER A1 ya ensamblada (a la izquierda) y finalizado (a la derecha), imagen propia

3.3.3 Segunda fase

En esta fase se instala la cámara de 2MP. Partimos de tal y como se aprecia en la imagen Figura 3-10.

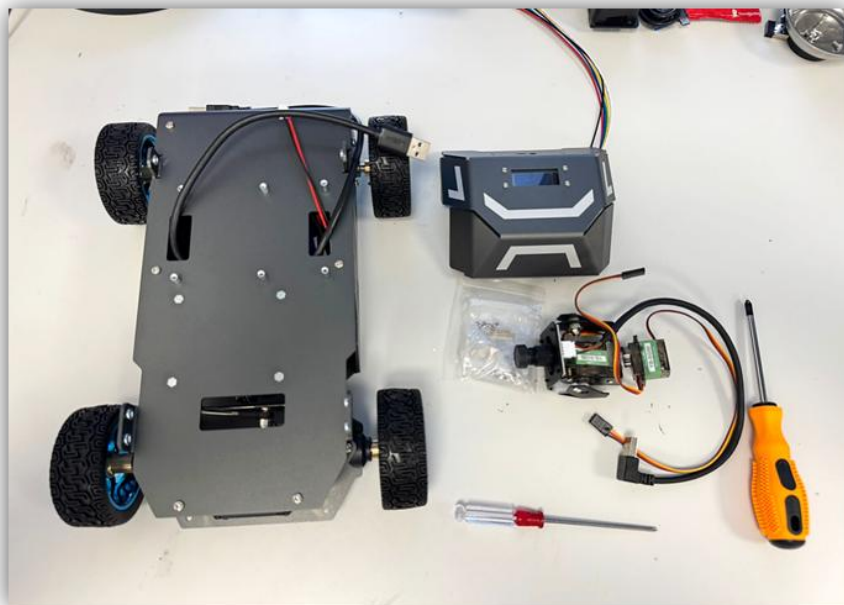


Figura 3-10 Componentes para el montaje de la 2º fase, imagen propia

Material que emplear:

- Base de la 1º fase
- Cámara 2MP con sus dos servos ya montados
- Destornilladores
- Paquete de tornillos y repuestos del módulo de la cámara

En este proceso se comienza por fijar la cámara a la chapa delantera del chasis (se desmontó en la 1º fase). Para ello se ha de desmontar el servo inferior de la cámara. Posteriormente se inserta el módulo de la cámara por un agujero en dicha chapa y se une mediante tornillos al servo inferior quedando ambos conectados y fijos, Figura 3-11.

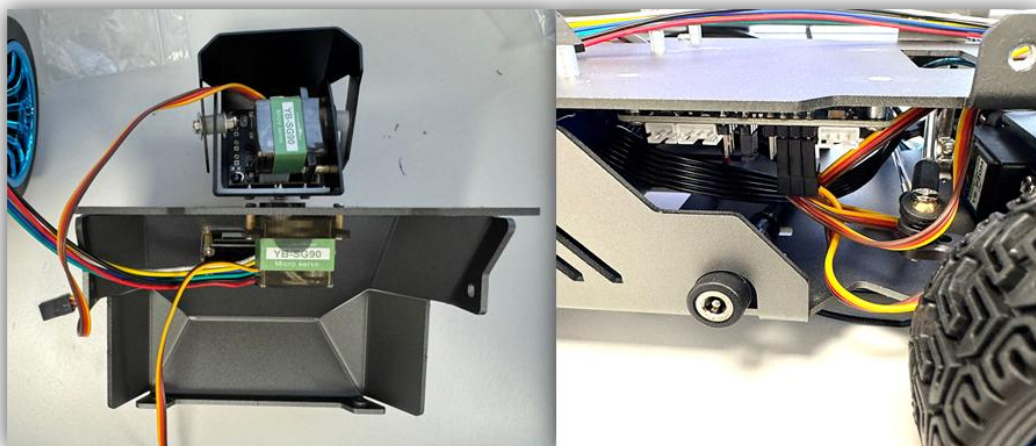


Figura 3-11 Montaje de la cámara en el chasis (a la izquierda) y conexión de los cables del servo de la cámara (a la derecha), imagen propia

Por último, se une mediante tornillos la chapa delantera a la base y se conectan los cables (marrón, naranja y amarillo) de cada servo a la placa de control de la base, Figura 3-11. Este último paso es complicado de realizar debido al poco espacio disponible para operar. Finalizando de esta manera la fase dos, Figura 3-12.



Figura 3-12 Montaje de la 2ª fase finalizada, imagen propia

3.3.4 Tercera fase

En esta tercera fase se instala el que sería el cerebro del robot, la Raspberry Pi 5 y la placa de expansión de USB 3.0 que proporciona más puertos a la placa de computación central. A continuación, en la Figura 3-13 se pueden ver los componentes necesarios para esta fase.

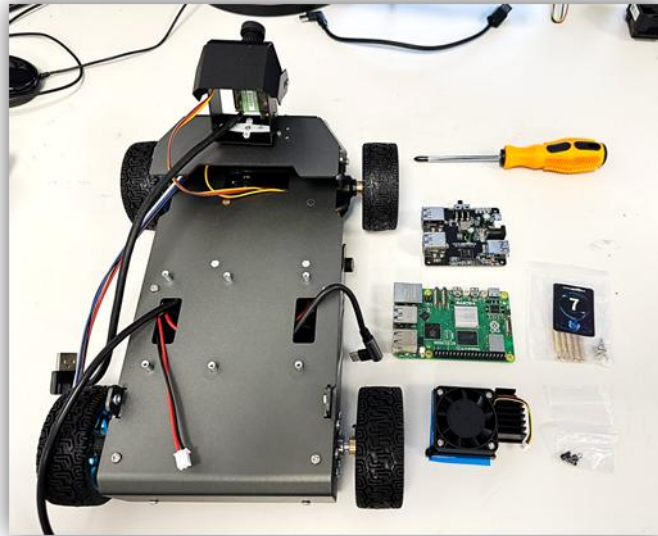


Figura 3-13 Componentes para el montaje de la 3ª fase, imagen propia

El material que emplear es:

- La 2ª fase como base
- Destornillador
- Raspberry Pi 5
- Ventilador Pi 50
- Placa de expansión USB 3.0
- Paquete de tornillos N°7
- Paquete de tornillos del ventilador

En primer lugar, se atornilla la placa de expansión USB 3.0 al nivel superior de la base del chasis, Figura 3-14. Y sobre ésta se atornilla a través de unos pilares y tornillos, la Raspberry Pi 5 que a su vez se conecta por cable y se le adhiere con un adhesivo y dos tornillos el ventilador Pi 50, Figura 3-15.

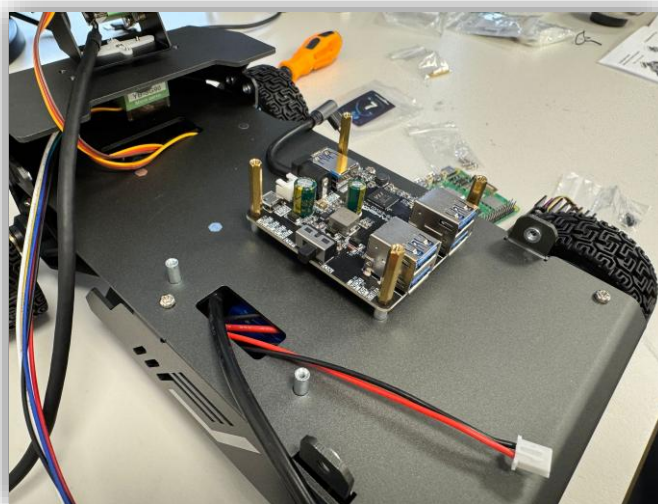


Figura 3-14 Instalación de la placa de expansión USB 3.0, imagen propia

A partir de este momento ya se pueden conectar las terminaciones de los cables de la siguiente manera:

- 1) La cámara se conecta por USB a la Raspberry Pi.
- 2) La Raspberry Pi 5 y la placa de expansión USB 3.0 se conectan por USB.
- 3) La placa base de control se conecta a la Raspberry Pi 5 por un cable tipo C.
- 4) El conector Bluetooth del mando RC se inserta en un puerto USB de la Raspberry Pi 5.
- 5) Los cables de la pantalla LCD se conectan a la Raspberry Pi 5.
- 6) Se inserta la tarjeta microSD en la Raspberry Pi 5.

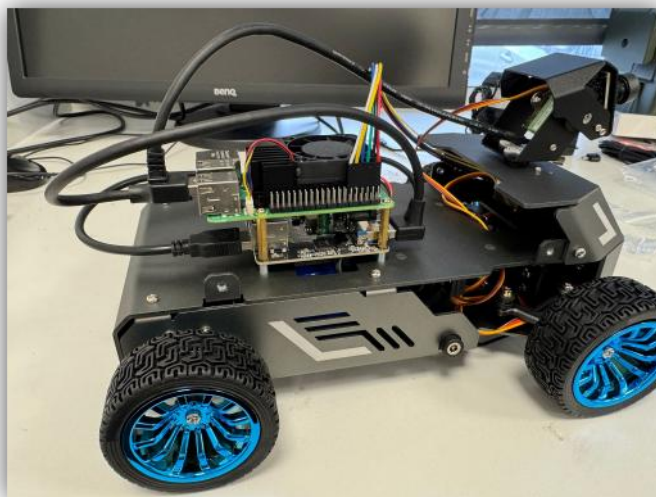


Figura 3-15 Montaje de 3º fase finalizado, imagen propia

3.3.5 Cuarta Fase

En esta última fase se procede a montar el LiDAR, el módulo de IA y el altavoz en la superestructura del chasis que posteriormente se hará firme al resto de la plataforma, a partir de los componentes mostrados en la Figura 3-16.



Figura 3-16 Componentes para el montaje de la 4º fase, imagen propia

Material que emplear:

- La plataforma de la 3º fase

- La chapa de la superestructura del chasis
- El T-mini LiDAR
- Un altavoz
- El módulo de IA y su cable
- La placa adaptadora del LiDAR
- Un destornillador
- Los paquetes de tornillos N°3 y N°5
- Cable que conecta el adaptador del LiDAR con la Raspberry Pi 5

Se comienza atornillando la placa adaptadora del LiDAR en la parte inferior de la chapa de la superestructura. En la parte superior de esta se atornillan, en la parte delantera, el LiDAR y en la trasera el módulo de IA junto al altavoz, resultado mostrado en la Figura 3-17.



Figura 3-17 Montaje inferior (izquierda) y superior (derecha) de la superestructura, imagen propia

Con respecto al cableado, se conecta el altavoz al módulo de IA mediante un cable por conexión Tipo-C y éste último se conecta a la Raspberry Pi 5, mediante el extremo USB; por otra parte, el LiDAR se conecta a su adaptador y éste a la Raspberry Pi. Por último, la superestructura se atornilla al resto del robot quedando ya montado tal y como se aprecia en la imagen de la Figura 3-18.



Figura 3-18 Montaje del ROSMASTER A1 finalizado, imagen propia

3.4 Software y configuración

3.4.1 Raspberry Pi 5

A continuación, comenzaremos la configuración del ordenador mono placa instalada. Lo primero de todo es acceder a la página web del ROSMASTER A1 de Yahboom[34] para descargar la imagen oficial, un archivo que posee la configuración de serie por parte de yahboom de todos los sensores y componentes electrónicos.

Una vez que abrimos el sitio web, nos debemos de desplazar hacia abajo en el índice que se encuentra en la izquierda de la página web, mostrado en la Figura 3-19. Donde vemos toda la información como tutoriales, ejemplos, etc, que nos ofrece Yahboom sobre su entorno y ecosistema. Encontramos el enlace “Robot_Image_System_File” que nos envía a una carpeta de Google Drive compartida en la que debemos elegir y descargar la imagen para Raspberry Pi que se encuentra comprimido en un archivo .zip.

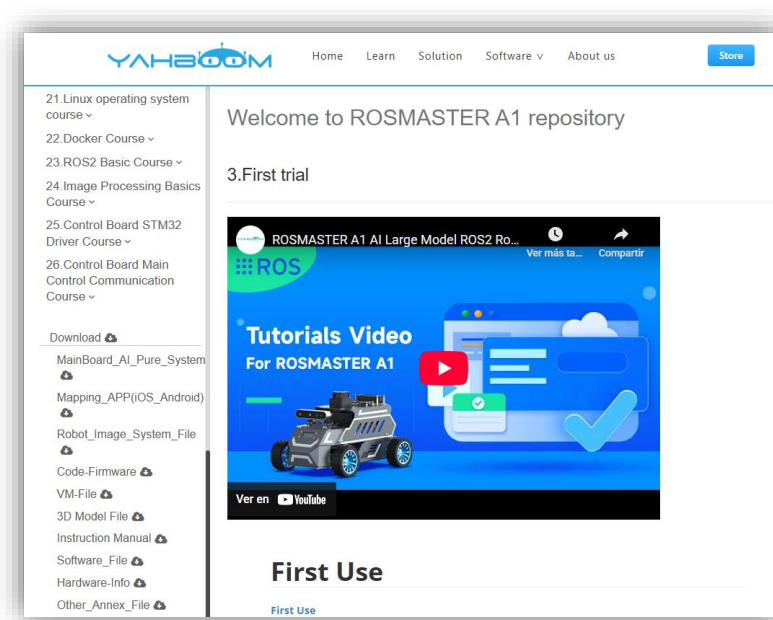


Figura 3-19 Tutorial proporcionado por la página oficial de Yahboom [34]

Con ese archivo, debemos abrir la aplicación de Raspberry Pi Imager donde seleccionamos nuestro dispositivo (Raspberry Pi 5) y elegimos el sistema operativo personalizado para instalar la imagen oficial (ya descomprimida), mostrado en la Figura 3-20. El siguiente paso sería elegir la tarjeta microSD y escribir la imagen en ella. Este es un proceso lento que puede durar entre una hora y cuarto y una hora 30 minutos, para agilizarlo se puede emplear una tarjeta microSD de mayor calidad comparada con la que se incluye en kit, en este caso se ha emplea una tarjeta SanDisk Extreme Pro de 128GB, logrando realizarlo en aproximadamente 50 minutos.

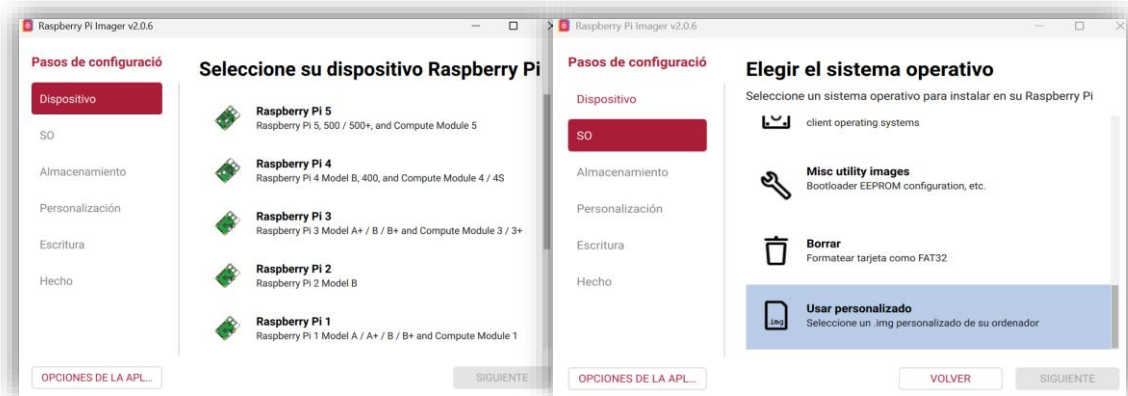


Figura 3-20 Proceso del traspaso de la imagen oficial a la tarjeta microSD empleando el programa de “Raspberry Pi Imager”, imagen propia

Finalizado el traspase de datos, instalamos la tarjeta en la Raspberry Pi 5 del robot y procedemos a encenderlo. Entonces, se mostrará en la pantalla LCD del robot los datos de su Internet Protocol (IP), porcentaje de uso de CPU, porcentaje de RAM y espacio empleado de la memoria (de la tarjeta microSD), mostrado en la Figura 3-21.



Figura 3-21 Pantalla LCD del ROSMASTER A1, imagen propia

Por último, empleando una aplicación para conexión remota por SSH (Secure Shell) y conectándonos a la red “ROSMASER-A1” que proporciona el robot cuya contraseña de serie es “12345678”, se nos preguntará el nombre de usuario y la contraseña que son “pi” y “yahboom” para acceder a la Raspberry Pi.

3.4.2 Programas instalados

Para poner en funcionamiento el robot y comenzar con el aprendizaje, se deberá hacer la instalación previa de una serie de aplicaciones para conectarse al ROSMASTER A1 de forma remota con la opción de una vista de ello más visual y dinámica que otras. Para su descarga, no encontraríamos ningún problema dependiendo del sistema operativo, ya que existen versiones para *Windows*, *IOS* y *Linux*.

En primer lugar, se necesita la aplicación de Raspberry Pi Imager para escribir la imagen oficial en la tarjeta microSD, tal y como se ha explicado anteriormente en 3.4.1, se trata de un programa gratuito, de descarga rápida y que ocupa un espacio de 4MB en la memoria. Se obtiene a partir del siguiente enlace [35].

En segundo lugar, están las aplicaciones para realizar la conexión con el robot. La página oficial de Yahboom nos recomienda el empleo de “Putty” que consiste en un programa que supone poco espacio de memoria para el ordenador (2MB), es gratuito y sencillo de emplear; nos abre una ventana en la que

escribimos la dirección IP del robot y a continuación nos abre una terminal de la Raspberry pudiendo empezar a trabajar con ella desde ese momento, se descarga a partir del enlace[36]. Por otra parte, se recomienda también el uso del programa “RealVNC Connect” que nos proporciona una imagen visual del escritorio de la Raspberry Pi 5, mostrado en la Figura 3-22 Captura del escritorio de la Raspberry Pi 5 tras la conexión por medio del programa "RealVNC Connect", imagen propia, su descarga es de 30MB, se encuentra disponible para diferentes sistemas operativos, sin embargo, su empleo gratuito solo es posible en caso de conexión alámbrica, para la inalámbrica es necesario crear una cuenta e iniciar sesión y se nos proporciona una prueba gratuita de 15 días, para un uso más prolongado es necesario pagar una suscripción, su descarga se realiza a través del enlace [37].

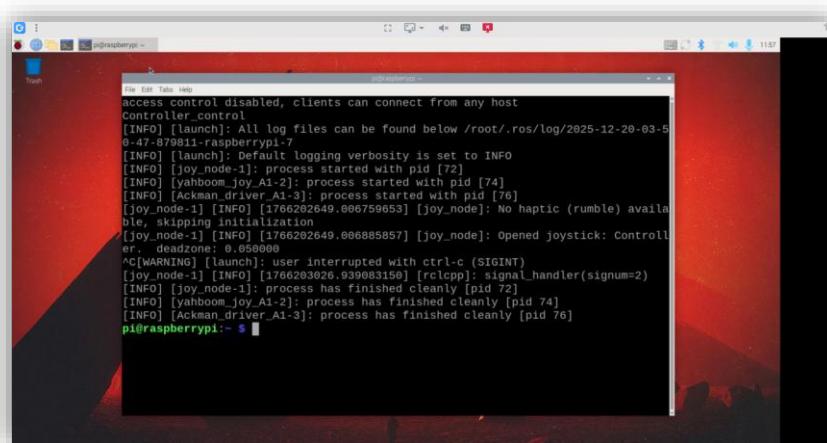


Figura 3-22 Captura del escritorio de la Raspberry Pi 5 tras la conexión por medio del programa "RealVNC Connect", imagen propia

Una vez dentro de la Raspberry Pi, hay que saber en qué *Docker* (contenedor), se encuentran los archivos para hacer funcionar y trabajar nuestra unidad. Para ello, con el comando “docker ps -a” seremos capaces de saber qué *Dockers* existen y han existido, de forma que elegimos el “yahboom_ros2_gui”. Ejecutamos el comando “sudo docker exec -u root -it yahboom_ros2_gui /bin/bash”, y ya estaremos dentro con acceso a todos los archivos de ROS2 que incorpora la imagen oficial proporcionada por Yahboom. Además, nos muestra el tipo de robot, y los modelos de LiDAR y cámaras instalados, Figura 3-23.

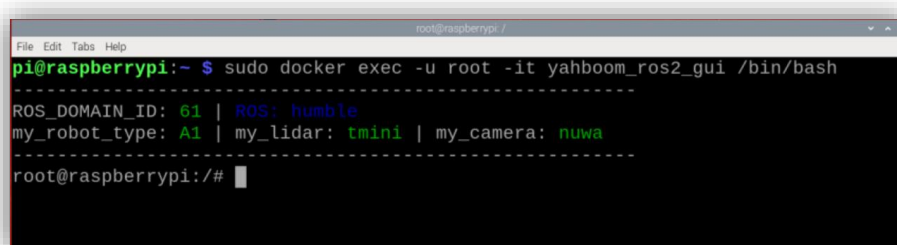


Figura 3-23 Entrada en el Docker de trabajo, imagen propia

3.4.3 Movimiento RC

El método más sencillo de mandar señales a nuestra plataforma para que ésta se mueva es a través del mando radiocontrol Figura 3-24. De esta manera, el alumno sería el piloto de la plataforma y la dirigiría para hacer tareas, por ejemplo, de mapeado de una habitación, localización de un objeto o ver las poses en la que se encuentra un grupo de personas, es decir, el alumno puede tener el control de la

unidad a la hora de hacer tareas de reconocimiento. Por medio del módulo de IA, el robot podría llevarlas a cabo sin necesidad de un piloto, sin embargo, lo consideramos cómo una función avanzada.

Seremos capaces de controlar de forma remota el robot de dos maneras, ambas con el mando teledirigido que incluye el kit básico; siempre que se quiera emplear se ha de pulsar una vez el botón “R2” del mando, las funciones de cada botón se muestran en la Tabla 3-1. La conexión de dicho mando con la Raspberry Pi se hace por bluetooth una vez que se conecta el receptor de éste a la placa base vía USB. El primer método para controlarlo remotamente siempre se nos ofrece nada más encender el robot cuando se abre una terminal. El segundo método consistiría en mandar nosotros mismos el comando desde una nueva terminal (si se ha cancelado o cerrado la terminal inicial), sin necesidad de entrar en un Docker, es decir, se ejecuta desde la terminal de la Raspberry Pi con el comando siguiente: `bash /home/pi/Rosmaster/rosmaster/start_ros2_humble.sh`

Se puede ver el manejo RC en el vídeo del siguiente enlace:

https://drive.google.com/file/d/1Skpl5MfMDpr_P4edYsLEg9j-geMOYOtx/view?usp=sharing



Figura 3-24 Mando RC, imagen propia

Movimiento	Botón
Avanzar hacia delante/atrás	Joystick izquierdo
Cambio de rumbo izquierda/derecha	Joystick derecho
Movimiento de la cámara en el Eje Y	Flecha izquierda/derecha
Movimiento de la cámara en el Eje X	Flecha arriba/abajo
Acelerar	Pulsar Joystick derecho
Decelerar	Pulsar Joystick izquierdo
Conexión Bluetooth	Mantener pulsado START

Tabla 3-1 Tabla que relaciona el movimiento del robot con los controles del mando

3.4.4 Cámara y su funcionamiento

En este apartado, se abordarán características, puesta en funcionamiento y funciones que posee y puede llevar a cabo nuestra plataforma a través de su cámara, el sensor que le dota del sentido de la vista.

Además, para las pruebas que se van a llevar a cabo, seremos capaces de ver lo mismo que ve nuestra unidad como en la Figura 3-25, gracias al software instalado por medio de la imagen oficial de Yahboom.

En cuanto a sus características, la cámara conectada por USB a la Raspberry Pi posee una resolución de 2MP HD capaz de detectar colores, no la profundidad, para debido a lo cual en caso necesario el robot se apoya en la información del LiDAR. Su velocidad de fotogramas varía entre los 15 y 30 *Frames Per Second* (FPS) su campo de visión es fijo, de 75°, y para tener un rango de visión más panorámico puede emplear los dos servos motores que permiten mover y reorientar la estructura de la cámara. Si la comparamos con el modelo incorporado en los kits superiores, un punto negativo del modelo adquirido es que no es capaz de detectar a qué profundidad se encuentran los objetos que ve, sin embargo, para un curso de iniciación, nuestro modelo es suficiente para ejecutar tareas básicas como la localización e identificación de objetos, por ejemplo.

Entre las funciones que puede realizar, y que se pretenderán llevar a cabo más adelante, se encuentran la posibilidad de detectar e identificar objetos empleando modelos de IA, de forma que puede realizar tareas como seguimientos faciales o de colores. Para visualizar el vídeo que la cámara percibe es necesario poner en funcionamiento el mecanismo de envío y recepción del flujo de vídeo mediante estos pasos:

- 1) Entramos en nuestro Docker de yahboomros2 y en la terminal ejecutamos el siguiente comando para activar la cámara: `ros2 launch usb_cam camera.launch.py`
- 2) Y para la ver lo que muestra la cámara ejecutamos: `ros2 run rqt_image_view rqt_image_view`

Se puede ver el funcionamiento y movimiento de la cámara en el vídeo del siguiente enlace:

<https://drive.google.com/file/d/1M2KjMdmZMQ6Rc-VByv6GevojmhmMH3Kn/view?usp=sharing>



Figura 3-25 Demostración del funcionamiento y visualización de la cámara del ROSMASTER A1, imagen propia

3.4.5 LiDAR y su funcionamiento

A continuación, nos centraremos en las características, información y puesta en funcionamiento del LiDAR T-mini Plus que incorpora nuestra unidad con el kit básico de Yahboom. Consiste en un sensor TOF (Time-of-Flight) que proporciona un escaneo todo horizonte, 360°, en dos dimensiones con la finalidad de llevar a cabo con él mapeos de zonas y navegación SLAM precisa. Según fuentes oficiales [32] posee un alcance máximo de 12 metros y una velocidad de escaneo de 12Hz.

En primer lugar, para activar el LiDAR se ha de abrir una terminal en la Raspberry Pi, entrar y buscar en el “Docker”, contenedor, de Yahboom dónde se encuentran los archivos ejecutables para este sensor. Una vez hecho, introducimos el siguiente comando en la terminal dentro del Docker:

```
ros2 launch yahboomcar_bringup laser_bringup_launch.py
```

Entonces, comenzaremos a oír como el LiDAR comienza a funcionar y dar vueltas. Para ver los datos que nos proporciona en tiempo real tenemos dos opciones. La primera, es ver los datos a través de la terminal, una forma efectiva, pero poco visual que nos permite leer la información que recoge el sensor sin emplear demasiada memoria RAM (13%) y CPU (16%). Por otra parte, tenemos el segundo método que es mediante el empleo del programa RViz, una herramienta de visualización 3D de ROS2 y mapeo de robots, que muestra en tiempo real y de manera visual lo mismo que el robot detecta de su entorno Figura 3-26. Sin embargo, su uso implica que es necesario consumir más CPU (36%) y RAM (18%). Los comandos para la visualización de los datos se han de ejecutar en el mismo Docker de Yahboom, pero en una terminal distinta. Los comandos serían los siguientes:

- 1) Ver datos en la terminal: `ros2 topic echo /scan`
- 2) Ver los datos en Rviz:
- 3)
 - I. Escribir el siguiente comando en la terminal: `ros2 run rviz2 rviz2`
 - II. Cambiar de “Reliable” a “Best Effort” en la interfaz de la izquierda entrando en LaserScan>Topic> Reliability Policy

El funcionamiento del LiDAR y su visión en el programa RViz se puede ver en el vídeo del siguiente enlace:

https://drive.google.com/file/d/1NkmOp_FdPD8Jck0vTBxZ7L6dUMCwlmvN/view?usp=sharing

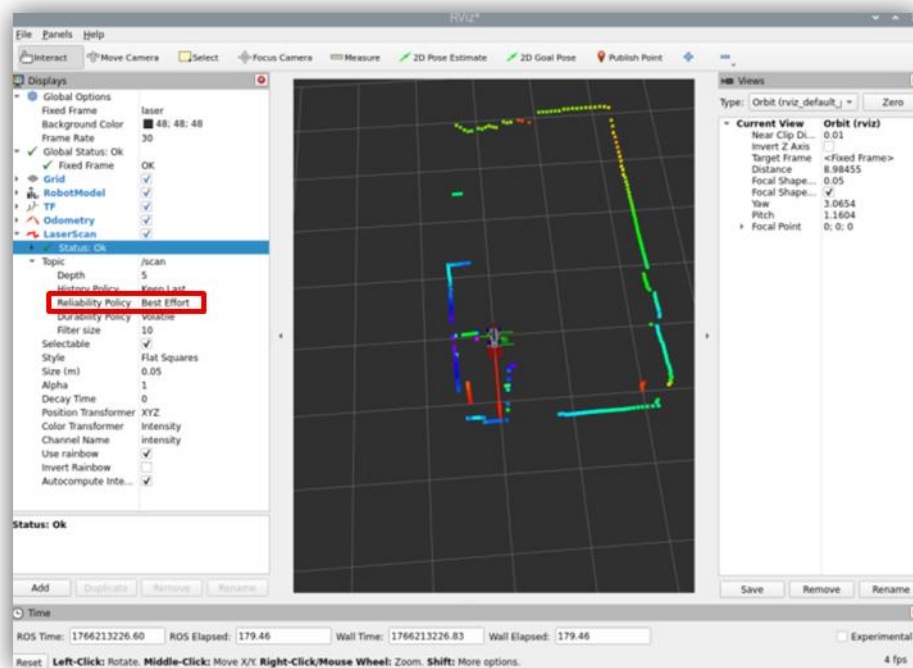


Figura 3-26 Programa RViz mostrando la nube de puntos generada por el LiDAR, imagen propia

En este programa, RViz, se nos permite ajustar la forma de visualización del LiDAR a gusto del usuario, desde cómo visualizar la nube de puntos, por tamaño, rango de colores, etc., activar o desactivar la odometría, al igual que el modelo del robot y los vectores, etc. Sin embargo, la visualización del programa demanda un porcentaje de CPU y RAM tal, que los FPS de la presentación de la Raspberry Pi se reducen considerablemente, y en ocasiones se llega a congelar la imagen.

3.4.6 Configuración de Altavoz y módulo de IA

En el ROSMASTER A1, el módulo de IA y el altavoz, buzzer, son uno para el otro, pues están conectados entre ellos físicamente por cable. Es al módulo de IA al que se le envía las señales y comandos para que haga sonar al altavoz, ya sea con unas palabras concretas o cuando el robot toma la iniciativa propia a la hora de cumplir tareas, además de contestar e interpretar órdenes recibidas por el micrófono que posee el altavoz.

Con respecto a su configuración, en primer lugar, empleando un comando sencillo en la terminal de la Raspberry Pi comprobamos si éste funciona al escuchar que emite un sonido. Sin embargo, para que reproduzca un sonido específico y que diga palabras en concreto deberemos instalar previamente en el Docker que emplearemos, un motor de voz. Los comandos serían los siguientes:

- 1) Comando para probar funcionamiento: `speaker-test -D plughw:2,0 -t Pink -c 1`
- 2) Instalamos el motor de voz en el Docker: `sudo apt install espeak -y`
- 3) Comando generar palabras durante un intervalo de tiempo: `while true; do espeak -v es -s 160 -p 40 "hola mundo" --stdout | aplay -D plughw:2,0; sleep 0.5; done`

En el siguiente enlace se puede ver un vídeo en el que el robot, por medio del altavoz, reproduce un texto escrito. Con el fin de probar el funcionamiento de éste:

<https://drive.google.com/file/d/12AL-201BQtGcn2yggasgapm-47yqkMVx/view?usp=sharing>

3.5 Funciones Avanzadas

3.5.1 Mapeado de una superficie

El ROSMASTER A1 cuenta entre sus funciones la capacidad de generar un mapa de una superficie por medio de su sensor LiDAR gracias a la nube de puntos que produce éste. Gracias a la imagen oficial de Yahboom descargada, encontraremos dentro del Docker que empleamos, los ejecutables que pondrán en marcha esta función del robot.

El proceso para la activación de esta función, mostrada en la Figura 3-27 es la siguiente, en primer lugar, se deberá entrar al Docker, siempre que sea necesario abrir una terminal nueva. A continuación, se ejecutarán en terminales distintas el comando que active el LiDAR, el que inicie la función del mapeado y otro para que por medio del programa RViz, se vea en tiempo real el área que está analizando la unidad. Para que el robot se desplace en la zona, será necesario que se inicie el control remoto empleando el mando RC o el teclado de nuestro ordenador, puesto que supone un menor esfuerzo de procesamiento para la Raspberry Pi comparado con un control autónomo por parte del robot.

Se puede ver como realiza el mapeado RC de una sala en el vídeo del siguiente enlace:

<https://drive.google.com/file/d/10aITUyPOdSvTC9P4t6VhKHxJk6RQmUYS/view?usp=sharing>

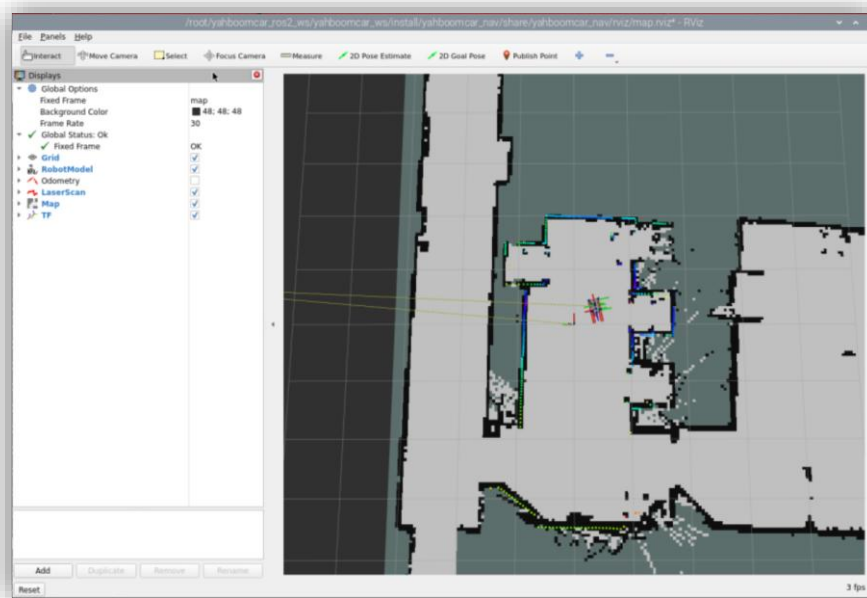


Figura 3-27 Función de mapeado y datos recopilados, a la vez, mostrados a través del programa RViz, imagen propia

Una vez finalizado el análisis y la obtención de datos, ejecutando un nuevo comando en otra terminal, somos capaces de obtener y guardar una imagen de dicha superficie, Figura 3-28. Esta característica dota al usuario de la capacidad de conocer y comprender cómo se puede obtener un plano de un entorno que es, inicialmente, desconocido. Además, a partir del mapeado inicial realizado, es posible llevar a cabo nuevas funciones como son el SLAM y la navegación autónoma.

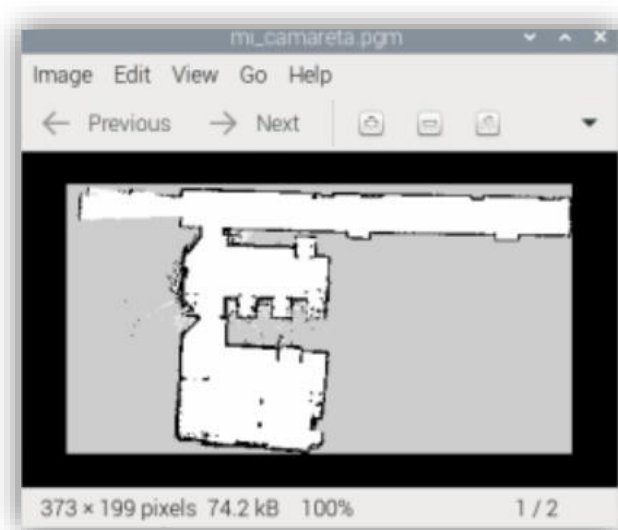


Figura 3-28 Mapa final obtenido, imagen propia

3.5.2 Navegación autónoma y evitación de obstáculos

Una función más que posé el ROSMASTER A1 haciendo uso del LiDAR es la de ser capaz de navegar de forma autónoma por una superficie. Cabe destacar que el robot lo puede llevar a cabo de dos formas. La primera forma es por medio de la ejecución de un código procedente de la imagen oficial de Yahboom, que consiste en analizar los datos obtenidos por el LiDAR y en función de ellos el robot decide si continuar hacia delante en línea recta o cambiar de rumbo en caso de que detecte un objeto u obstáculo demasiado cerca, sin necesidad de tener un mapa previo guardado. Para activar la función se necesitan dos comandos que se han de ejecutar en dos terminales diferentes dentro del mismo Docker, el primero será para activar el movimiento del chasis y el segundo para iniciar el código que ejecuta dicha función. Además, este método se puede combinar con la función de mapeado de un área para que sea el propio robot el que recorra la zona sin necesidad de un piloto que lo controle de forma remota y así obtener el mapa.

La función de evitación de obstáculos se puede ver en el vídeo del siguiente enlace:

https://drive.google.com/file/d/12CKqV_JXcBObOU7qaEtgTKcnS5g_UvWB/view?usp=sharing

La segunda forma de navegación autónoma es a partir de un mapa ya guardado previamente que el robot empleará para desplazarse y cumplir tareas en un área ya conocida, mostrado en Figura 3-29. Para llevarlo a cabo se ha de ejecutar en terminales diferentes los comandos para realizar el mapeado con una función diferente a la mencionada en 3.5.1 para que devuelva un archivo “.pbstream” que es necesario para que RViz lo pueda interpretar; activar el movimiento autónomo del chasis, el LiDAR, y ejecutar el archivo para la navegación por medio de un mapa ya guardado y entonces se abrirá el programa de RViz en el que se mostrará el mapa y la opción de situar de forma estimada la ubicación del robot en él, que el mismo corregirá con respecto a las medidas que toma con su LiDAR, y la de mandarle a una zona en concreta a la que desplazarse.

La navegación con mapa, SLAM, se puede ver en el vídeo del siguiente enlace:

<https://drive.google.com/file/d/1mIdufACGnsGNILNBZfsZ7p673RqKRJHB/view?usp=sharing>

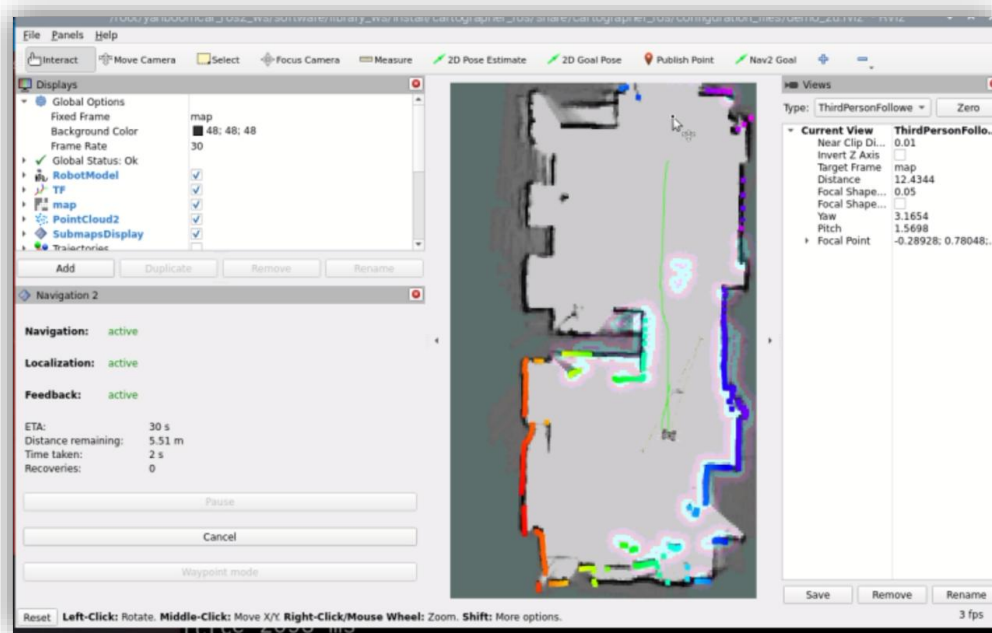


Figura 3-29 Función de navegación con mapa iniciada, mapa y ruta cargadas, imagen propia

3.5.3 Identificación de colores, objetos, gestos y posturas

En cuanto a las funciones que puede realizar gracias al empleo de la cámara y del módulo de IA, se encuentra la posibilidad de identificar desde posturas y posiciones corporales y de manos, así leer e interpretar códigos QR, pues se le puede mandar tareas e instrucciones a través de estos. También posee la capacidad de identificar y hacer un seguimiento de un objeto seleccionado por medio del color de éste. E incluso, gracias a la imagen oficial de Yahboom, posee un archivo con los diferentes tipos de objetos que al detectarlos mediante la cámara y junto al módulo de IA es capaz de identificar y clasificar lo que el robot ve, desde humanos y animales hasta objetos como material de escritorio y vehículos, mostrado en la Figura 3-32.

Con respecto a cómo se inician estas funciones, en primer lugar, buscaremos dentro del Docker el paquete que contiene los ejecutables de las funciones de “Yahboom astra”. Una vez que conocemos los nombres exactos de cada función se tendrán que ejecutar tan solo dos comandos en dos terminales diferentes, uno para activar el movimiento del chasis y otro para activar la función concreta. A continuación, se abrirá una ventana que mostrará la imagen en directo de la cámara del robot junto a la función elegida. Entonces se podrá seleccionar con el ratón un área de un color para poder realizar el seguimiento (si es la función de “colorTracker”) Figura 3-30, si se interpone una persona en su visión ésta será analizada y veremos una silueta con su postura (si es la función de “postureTracker”) Figura 3-31. Por otra parte, si se pulsa a continuación la barra espaciadora de nuestro teclado, el robot tratará de mantener centrada la imagen del objetivo seleccionado.

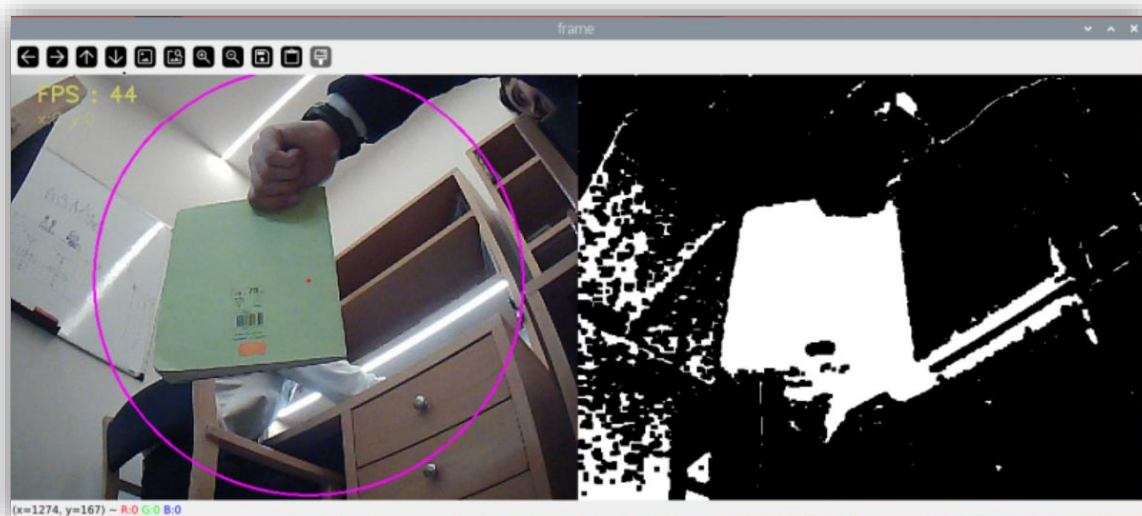


Figura 3-30 Función de seguimiento de color activada, imagen propia

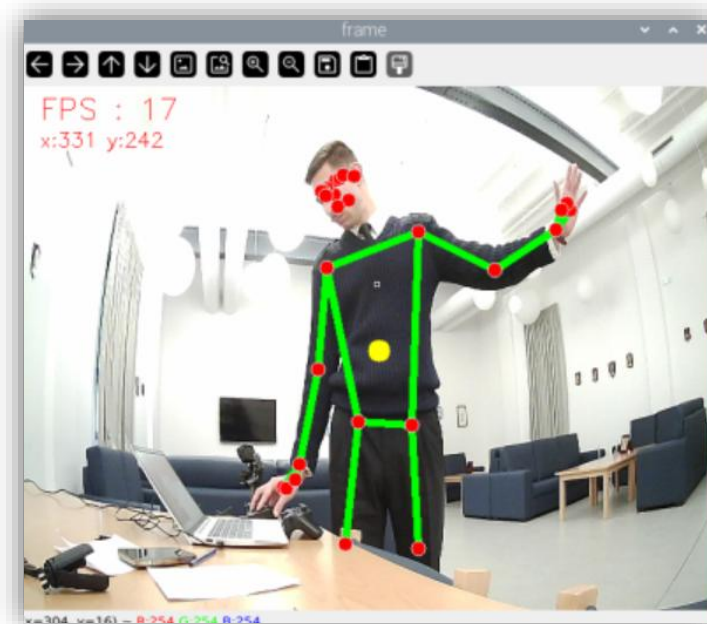


Figura 3-31 Función de seguimiento de postura corporal activada, imagen propia

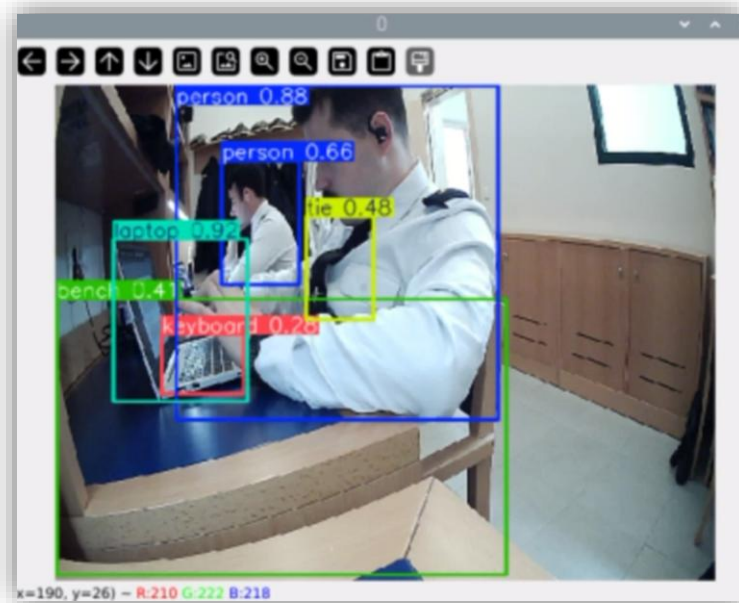


Figura 3-32 Función de identificación de objetos, imagen propia

Además, la imagen oficial de Yahboom incluye la función de seguimiento por gestos. Consiste en que el robot fija su visión en una mano en una mano del sujeto y en función del gesto que se haga con ella, que puede ser de parar, “STOP”, cerrando el puño, o la de avanzar, “RUN”, mostrándole la palma de la mano. Se inicia ejecutando su función con el comando: `yolo predict model=/root/ultralytics/ultralytics/yolo11n.pt source=0 show=True`, en una terminal y en otra se ha de activar el movimiento del chasis. A continuación, se mostrará una venta que muestra lo que ve la cámara, la mano que fija para interpretar las órdenes y la orden recibida en la parte superior.

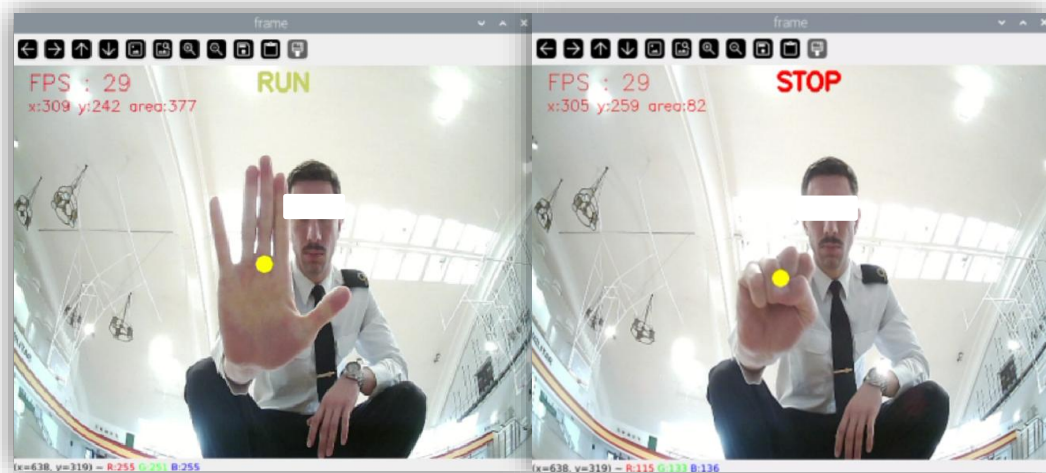


Figura 3- 33 Función de seguimiento por gestos: avanzar (izquierda) y detenerse (derecha), imagen propia

En los siguientes enlaces se pueden ver las funciones de la cámara desarrolladas:

- 1) Seguimiento por colores:

<https://drive.google.com/file/d/1vdl5AmAJvA78CloxInN3O2nSVMhE02Ur/view?usp=sharing>

- 2) Seguimiento de personas:

<https://drive.google.com/file/d/1tVtKYPZsELSHHzZnN9b4t3ElosfA43mq/view?usp=sharing>

3) Identificación de objetos:

<https://drive.google.com/file/d/1JSOEXKft1EWCCv68cjlFJkBK-pCVbnny/view?usp=sharing>

4) Seguimiento por gestos:

<https://drive.google.com/file/d/1-eeIIVaoRNIJTFiHrjpkUNepF19pfne0/view?usp=sharing>

3.5.4 Seguimiento de línea

Otra función avanzada que posee el ROSMASTER A1 es la de desplazarse de manera autónoma siguiendo una ruta delimitada por una línea en el suelo, Figura 3- 34. Esta capacidad la logra mediante el empleo de la cámara de 2MP y servos *Pan Tilt Zoom* (PTZ) que incorpora la plataforma. Su activación se lleva a cabo ejecutando dentro del mismo Docker y en terminales diferentes, el comando que inicia el movimiento del chasis autónomo y el ejecutable de “Yahboom astra” de seguimiento de línea, que iniciará la cámara y abrirá una nueva ventana en la que se ha de seleccionar un área de la línea que muestra la cámara, se pulsa la tecla espaciadora del teclado y el robot comenzará el seguimiento de ella.

Esta función se puede ver en el vídeo del siguiente enlace:

<https://drive.google.com/file/d/1TOn6-ApYqI8TGANln-25QZv4nbuVl091/view?usp=sharing>

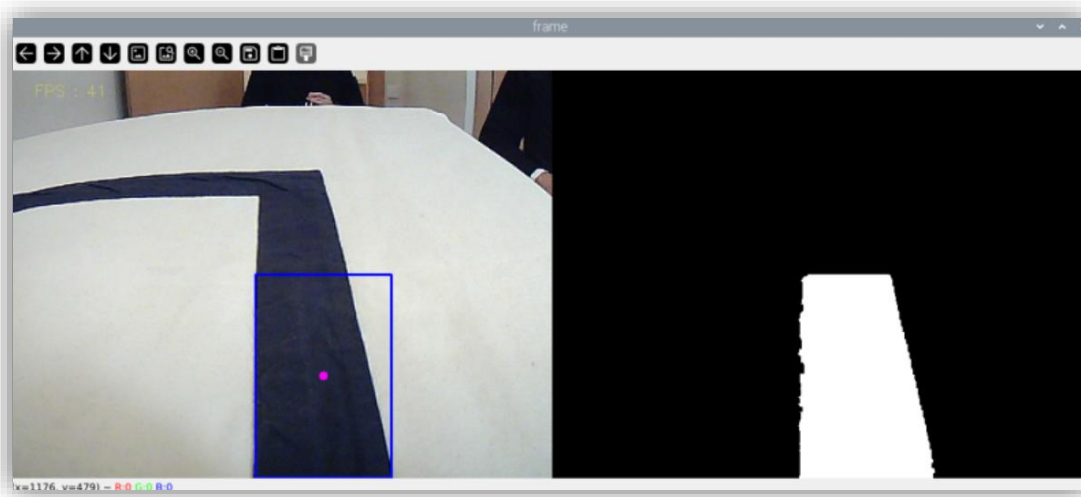


Figura 3- 34 Función de seguimiento de línea, imagen propia

3.5.5 Función de enjambre

Con esta función se consigue que dos o más unidades de ROSMASTER A1 se enlacen entre sí con el fin de controlarlas simultáneamente como si tan solo fuera una misma unidad. Para llevarla a cabo, se ha de realizar en primer lugar un paso vital, que es el de enlazar las unidades conectándolas a la misma red. El desarrollo de este proceso se han empleado dos unidades una facilitada por el Centro Universitario de la Defensa (CUD) que se le denominará “robot 2”, y otra que es personal, denominada “robot 1”.

En primer lugar, para realizar el enlace se empleará la red del robot 1 como nodo, por lo que cuando entremos con el programa RealVNC Connect en ambas Raspberry Pi; con el robot 2 abriremos las opciones de conexión Wifi y nos conectamos a la del robot 1, cuyo nombre de serie es “ROSMAS-TER-A1” (si se desea se puede cambiar el nombre de la red). Llegados a este punto se encontrarán conectados a la misma red, y el robot 1 tendrá su propia IP y el robot 2 también.

En segundo lugar, se ha de abrir una terminal nueva, iniciar el Docker de trabajo y entrar en él, en cada robot. Para a continuación, ejecutar un comando que denomine de manera “oficial”, para la

ejecución de la práctica que se pretende llevar a cabo, cuál es el “robot 1” y el “robot 2”. La primera comprobación que se hace es la del control remoto sincronizado, en el que las dos unidades responden de la misma manera al recibir la orden procedente únicamente de un mando RC que está conectado por bluetooth al robot 1. Para ello, se ha de ejecutar el comando de dicha función en una de las terminales y entonces con el mando RC, conectado al robot sobre el que se le ha ejecutado en la terminal, ese robot transmitirá las órdenes al otro y comenzarán a desplazarse según la orden dada.

En el siguiente enlace se puede ver un video que muestra dos unidades siendo controladas por un único mando RC:

<https://drive.google.com/file/d/1jP1zjpRX9n2GxUd1dsmcdSljbRO8JS8U/view?usp=sharing>

Por otra parte, se encuentran funciones avanzadas relacionadas con el SLAM que también se pueden realizar de forma conjunta. A partir de un mapa ya creado y denominando cual es el “robot 1” y “robot 2”, es posible que, siguiendo una secuencia de comandos, ambos robots sean capaces de navegar en una misma zona con un mismo mapa, desde independientemente hasta en formación. Es decir, con el mapa cargado, la función de SLAM activada (desde una terminal de uno de los robots) y el programa RViz, abierto, veremos en dicho programa las figuras 3D de cada robot sobre el mapa, y se nos da la posibilidad de estimar la posición de ambos y de mandarles la tarea de desplazarse a una zona concreta de manera independientemente, de la misma manera que se ha explicado en el apartado 3.5.2. De la misma manera, es posible abrir una ventana de configuración de funciones de procesos iniciados y seleccionar la opción de que vayan los dos en formación, ya sea en columna como en línea de avance, y determinar la distancia de separación entre unidades. Entonces, a la hora de mandar la tarea de desplazarse a una zona concreta éstos actúan como si fueran uno y evitan los obstáculos teniendo en cuenta la envergadura de la formación.

4 RESULTADOS

4.1 Alcance y criterios de valoración

En este capítulo se presentarán los resultados obtenidos sobre el montaje, la configuración y la ejecución de funciones avanzadas del ROSMASTER A1, con el objetivo de mostrar su operatividad y recoger lecciones útiles sobre su uso docente. Quedarán fuera de alcance los análisis estadísticos exhaustivos o comparativas de rendimiento entre algoritmos; priorizando la evidencia de funcionamiento en el entorno de un aula o laboratorio y la reproducibilidad básica de los procedimientos. La valoración se realiza mediante criterios de aceptación simples por área de montaje, configuración y funciones avanzadas, sustentados en evidencias mínimas (como fotos, capturas de pantalla y registros de consola). Se empleará la siguiente escala Cumple/Cumple con observaciones/No cumple para cada criterio y se anotarán incidencias y mejoras detectadas para su futura utilización en prácticas.

4.2 Montaje: síntesis de resultados

4.2.1 Resumen del proceso

El proceso de montaje de la plataforma ROSMASTER A1, desarrollado en 3.3, que se ha dividido en cuatro fases por la relevancia de la instalación de los componentes que se instalan en la plataforma. Es decir, desarrollando el montaje de la fuente de energía, de la cámara, de la Raspberry Pi 5, y del LiDAR y el módulo de IA. Para ello se han seguido las directrices tanto del manual físico que incluye el kit como del tutorial que proporciona la página oficial [34]. En ambos, se muestra paso por paso y detalladamente la colocación exacta de cada componente y su respectiva conexión alámbrica, además de la métrica empleada en los diferentes tornillos y en los mecanizados hechos en la chapa de la plataforma.

En la primera fase, se llevó a cabo el montaje de la fuente de alimentación, una batería de litio de 12V y 6000mAh. No se detectaron fallos, sin embargo, a la hora de insertar la batería en su cavidad se ha de hacer de manera precisa para no ejercer ni efectuar presión sobre las conexiones de los motores de propulsión trasera. Por otra parte, en la segunda fase en la que se procede a la instalación de la cámara con sus dos respectivos servos PTZ, se encuentran dos deficiencias que pueden afectar a su uso prolongado empleado en docencia debido a la calidad de sus componentes. La primera deficiencia se detectó al desmontar el servo PTZ inferior para realizar el montaje de la cámara en su respectivo emplazamiento en la plataforma del robot, ya que se apreció que los engranajes de éste están hechos de un material plástico por lo que esfuerzos que produzcan demasiada fricción y temperatura es posible que a largo plazo sea necesario cambiar los engranajes o el servo entero, teniendo en cuenta que ambos son

reemplazables y se pueden encontrar disponibles por medio de la compraventa online. La siguiente deficiencia se encuentra a la hora de realizar la conexión por cable de los servos a la placa de control mostrada en la Figura 3-11, la cual, debido al diseño del robot, presenta un montaje complicado por la falta de espacio para operar pudiendo llegar a quebrar dichas conexiones.

Durante la tercera fase, se instala la Raspberry Pi 5 y la placa de expansión de puertos USB, y no se destaca ninguna imperfección en el montaje ni a la hora de realizar las conexiones. Por último, en la cuarta fase en la que se monta la superestructura del robot con el LiDAR y el módulo de IA junto al altavoz, se lleva a cabo sin ninguna novedad. Se puede afirmar que no es necesario tener una habilidad previa específica a la hora de ensamblar los componentes y que junto a los diferentes tutoriales no se convierte una tarea difícil.

4.2.2 Incidencias y correcciones

Con respecto a las incidencias y las correcciones llevadas a cabo, solo se destacará un caso, principalmente porque con él sí se ha alterado una conexión del diseño del robot, mientras que el resto de las incidencias mencionadas en 4.2.1 se subsanaron sin cambiar dicho diseño. Retomando la incidencia, se trata de la conexión de la cámara por cable USB a la Raspberry Pi 5, ya que tal y como viene explicado en el manual, se designa un puerto de la Raspberry específico para realizar la conexión. Sin embargo, al conectar la cámara se puede apreciar cómo el cable está sometido a una flexión excesiva que puede ocasionar tanto la rotura de éste como que la información recopilada por la cámara se envíe mal o con retardo. Por ello la solución a la que se llegó, ha sido cambiar el puerto USB de conexión preestablecido y emplear otro de la Raspberry Pi. Una vez que el robot esté en funcionamiento y la conexión remota por ordenador también, entonces mediante un comando en la terminal se puede comprobar si la Raspberry reconoce la conexión de la cámara, y si es afirmativo quiere decir que el cambio de puerto no afecta al funcionamiento y desempeño del robot.

4.2.3 Tiempo y recursos

Para el montaje de la unidad se le ha dedicado un tiempo y unos recursos concretos. Cabe destacar que el montaje se ha realizado 2 veces, pues se han empleado dos unidades de la misma plataforma, por lo que en base a la experiencia obtenida del primer montaje se ha logrado reducir el tiempo del segundo siendo el del primero de una hora y 50 minutos aproximadamente y el segundo de 55 minutos medidos con cronómetro. Esta varianza en el tiempo es debido a las incidencias y deficiencias tratadas en los apartados 4.2.1 y 4.2.2, que para el segundo montaje ya se conocía su solución. En cuanto a los recursos empleados, se encuentran los que incluye el kit mencionado en 3.3.1, como los destornilladores y las bolsas con el material necesario correspondiente a cada paso del montaje.

4.2.4 Lecciones aprendidas y mejoras

Una vez finalizado el montaje de las dos unidades del ROSMASTER A1 y haber analizado su estructura, ventajas y desventajas en este apartado se pretende mencionar una serie de lecciones aprendidas y mejoras en caso de que sea necesario.

Se comenzará por las lecciones aprendidas ventajosas. Entre estas cabe destacar el material del que está hecha la plataforma, de aluminio, lo que la hace resistir a caídas y a golpes leves. Además, la base metálica se encuentra mecanizada para facilitar la instalación de los sensores incluidos tanto en kit adquirido como en kits más avanzados, dotándole así una gran capacidad modular y la posibilidad de instalar sensores actualizados y renovados en un futuro que puedan ser atractivos dentro del ámbito docente. También se destaca la existencia de tutoriales, que reducen el tiempo y facilitan el proceso de montaje.

En cuanto a mejoras, se recomienda el empleo de un imán a la hora de realizar el ensamblaje para que se unan a él los tornillos de las diferentes fases del montaje con el fin de que éstos no se extravíen con tanta facilidad. Por otra parte, tenemos el diseño, en el que se encuentran un alto número de cables y conexiones expuestos que por consecuente hacen que la plataforma no sea idónea para su uso en

exteriores siendo muy vulnerable a la humedad y a la lluvia, y también en cuanto a su empleo en prácticas de laboratorio y sea manipulado por alumnos se corre el riesgo de que uno se enganche a un cable y que arranque la conexión de la manera. Por ello en ese caso se propondría diseñar una superestructura ligera que ayude a mantener toda la instalación de cableado cubierta y segura.

4.3 Configuración y verificación funcional

4.3.1 Imagen del sistema y contenedor

Una de las características principales que hace destacar al ROSMASTER A1 para su empleo en el ámbito docente es su software. Se trata de un software que ha sido desarrollado y puesto en práctica en centenares de unidades, por lo que se espera trabajar y operar en un entorno con una determinada calidad y garantía a la hora de emplearlo. Para sacar un buen partido de esta herramienta y exprimir el uso de la plataforma al máximo es necesario conocer el procedimiento para poner en funcionamiento correctamente dos componentes del software: su imagen oficial y su Docker (contenedor) de trabajo.

En cuanto a su imagen oficial, ésta se trata de un archivo de 42GB, lo que lo convierte en un archivo muy pesado teniendo por consecuente que dedicarle un tiempo de espera relativamente largo de 30 a 40 minutos, en función de la red que se emplee. El proceso para su obtención se encuentra desarrollado en 3.4.1 y también del proceso que se ha de llevar a cabo para instalarla en la tarjeta microSD. Este último proceso, en cuanto al tiempo empleado se refiere, se le considera mejorable y poco eficiente por el motivo expuesto en 3.4.1, pues supera la hora y cuarto de espera. Sin embargo, si se sustituye la tarjeta microSD que incluye el kit (cuyas características y empresa de manufacturación se desconoce) por una de calidad superior se reduce el tiempo del traspaso de datos hasta los 40 minutos. En nuestro caso, para llegar a esa conclusión se ha hecho la comparación empleando una tarjeta microSD SanDisk Extreme pro de 128GB cuya velocidad de lectura es de hasta 200MB/s (sus características técnicas se pueden obtener en [38]).

Con la imagen ya instalada y la Raspberry Pi en funcionamiento, para la puesta en marcha de cualquier función y sensor es necesario tener unos conocimientos básicos de programación y de empleo de terminales Linux, que es el sistema operativo que se emplea en la susodicha plataforma. Entre esos conocimientos es necesario saber localizar los contenedores disponibles, encenderlos, apagarlos y entrar en ellos, de lo contrario no se sería capaz de activar el robot para que realice las tareas que se le encomiende, puesto que todos los archivos ejecutables y códigos se encuentran en los contenedores (*Docker*).

4.3.2 Conectividad y acceso

El proceso siguiente, que es vital para la puesta en marcha del robot, es la conexión y acceso a éste de forma remota. Siguiendo las recomendaciones del tutorial de Yahboom se obtienen unos resultados favorables, esto se debe a que existen diferentes métodos posibles y válidos empleando el software de los programas Putty y RealVNC Connect. Si bien con ambos se obtiene el mismo resultado, es cierto que mediante el uso del segundo programa se logra una experiencia más interactiva.

Analizando de manera crítica el programa Putty se obtiene que su uso es menos intuitivo para un alumno con conocimientos demasiado básicos en programación e informática, por la necesidad expresa de trabajar sobre una terminal desde el primer momento, además del hecho de que a la hora de ejecutar archivos que necesiten varias terminales con el mismo *docker*, se ve limitado ante la imposibilidad de abrir más terminales. Por otra parte, tenemos el programa RealVNC Connect cuya principal ventaja es que su software permite mostrar en la pantalla del ordenador personal el escrito de la Raspberry Pi, logrando esa faceta interactiva. Sin embargo, está presente el inconveniente de que es un programa de pago, tal y como se ha comentado en 3.4.2.

4.3.3 Puesta en marcha de sensores

Una vez dentro de la Raspberry Pi y haber adquirido soltura a la hora de emplear la terminal, el siguiente paso es el de comprobar el funcionamiento de los sensores. Si bien es cierto que tanto en los tutoriales audiovisuales como en el tutorial paso por paso y función por función de Yahboom [34] los comandos que emplear para activar los sensores dependiendo del modelo de kit adquirido, las rutas pueden variar en algunos casos, lo que obliga a una búsqueda de los archivos previamente en la terminal dentro del contenedor *docker*. Antes de comenzar con los resultados de la primera puesta en marcha de los sensores, hay que destacar que sus archivos ejecutables y software se encuentran dentro del *docker* por lo que se deberán abrir tantas terminales de *docker* como funciones que se quieran activar.

La primero que se ha probado ha sido su función de control remoto por medio del mando RC, aunque no sea un sensor como tal, sí es una habilidad que posee el robot, dando uno resultados óptimos por su sencilla activación explicada en 3.4.3 y por su alcance superando los 20m de distancia de manejo del robot. Continuando con la cámara, después de localizar su comando para activarla y abrir una ventana que muestra lo que ve el robot en tiempo real, se puede dar por concluida su activación con éxito. Sin embargo, la cámara no alcanza a tener una buena calidad de imagen y tampoco de retransmisión, siendo ésta de menos de 24FPS, y en ocasiones genera cierto retardo y al alejarse más de 10 metros con paredes de por medio la retransmisión desde la Raspberry Pi hacia el programa de RealVNC Connect comienza a empeorar hasta llegar a congelarse la imagen.

El siguiente sensor que se puso en funcionamiento fue el T-mini LiDAR, que es el modelo que incorpora la unidad evaluada en este TFG. Frente al modelo de C1 del LiDAR de 3D que incorporan kits superiores, el LiDAR evaluado posee una desventaja notable al ser 2D. Sin embargo, esa carencia le dota de un menor consumo de energía y además al recoger menos datos facilita el procesamiento de estos en la Raspberry Pi, así como la retransmisión de información visual que es representada mediante el programa RViz en el ordenador desde el que se realiza la conexión. Con respecto a cómo el diseño del robot contempla la colocación del LiDAR, se puede afirmar que es óptima pues se encuentra en la parte más elevada de la superestructura permitiendo que tenga una visión de 360° sin obstáculos, es decir, que a la hora de recopilar datos para el SLAM lo hace en todas direcciones sin ningún tipo de obstrucción. Aunque haya otros componentes de hardware instalados en la parte superior del robot como el altavoz y el módulo de IA, no afectan al LiDAR, sin embargo, si se ve afectado cuando la posición de la estructura de la cámara se encuentra en su posición máxima elevación, pues interfiere al posicionarse entre el sensor y el objeto a detectar.

Por último, se llevó a cabo la puesta en marcha del altavoz y del módulo de IA de manera independiente y se ha comprobado que ambos funcionan correctamente y que junto al software incorporado mediante la imagen oficial de *Yahboom*, se pueden realizar funciones sin necesidad de una configuración y/o programación previa del módulo de IA. Por otra parte, es posible hacer funcionar al altavoz independientemente controlándolo desde una terminal dentro del *docker* de trabajo, y descargando un paquete de idiomas, el robot es capaz de reproducir textos escritos en forma de audios empleando la lengua especificada. También, cabe la posibilidad de generar un código para que cuando detecte un objeto al que se éste aproximando o cuando haya detectado un cara u objeto concretos, el altavoz reproduzca desde un sonido cualquiera a una frase con un volumen, tono y repetición especificadas por el operario, en nuestro caso el alumno. En cuanto al módulo de IA con YOLO 11, tenemos que éste se activa automáticamente al iniciar las funciones de la cámara y funciona de manera correcta a la hora de identificar los nodos de las articulaciones en personas, entender órdenes mediante gestos de “run” y “stop”, reconocimiento facial y la identificación de objetos.

4.3.4 Observaciones de rendimiento

En este apartado se pretenden abordar resultados del rendimiento del robot más enfocados a la duración de la batería, al calentamiento de los componentes durante su uso, y al porcentaje de exigencia de trabajo para la Raspberry Pi. Para ello nos basaremos en la experiencia obtenida de la práctica y de

los datos que nos proporciona el robot con su pantalla LCD, mostrada en el apartado 3.4.1 en la Figura 3-21.

Con respecto a su batería, cuyas características se mencionan en el apartado 3.1, según la página oficial, tiene una duración de cinco horas con un uso moderado del robot. En nuestro caso se ha apreciado, que para usos prolongados en los que solamente se trabaja en entender y aprender a utilizar el entorno de la Raspberry Pi durante dos periodos de tres horas al día, la batería ha aguantado hasta 3 días. Para configuración, puesta en funcionamiento y pruebas del movimiento RC, LiDAR y cámara en dos periodos de 3 horas al día la batería, ha suministrado energía suficiente durante un día y medio. Y en usos intensos, es decir, llevando a cabo funciones de SLAM, evitación de obstáculos y en identificación de objetos y seguimiento de objetivos al cabo de dos horas seguidas se aprecia como comienza a desplazarse más lento el robot, y además en la retransmisión por RealVNC Connect se aprecia que la Raspberry Pi comienza a aumentar los tiempos de carga al llevar a cabo funciones. Por último, cuando la batería se encuentra trabajando bajo mínimos, el robot comienza a emitir un pitido hasta que se detiene. Con respecto a la carga de ésta, se realiza mediante un cable que se conecta a la batería sin necesidad de desmontarla gracias al diseño de la plataforma; y, por otra parte, el transformador del cargador tiene un enchufe con clavijas estadounidense, pero como el kit incluye un adaptador internacional no supone un problema.

Luego tenemos el recalentamiento que se produce en los componentes. Entre los afectados se encuentran la Raspberry Pi, LiDAR y la placa de control del movimiento de chasis. No se trata de un aumento de temperatura extremo, sin embargo, es posible que el rendimiento se vea afectado a la hora de cargar funciones y tareas pues se aprecia una mayor lentitud al cabo de más de tres horas de funcionamiento. Para mitigar el aumento de la temperatura, a la Raspberry Pi se le conecta un ventilador modelo Pi 50, cuyo montaje se aprecia en el apartado 3.3.4 en la Figura 3-15, además el diseño del chasis permite la entrada de un caudal de aire que ayuda a refrigerar el resto de los componentes, mitigando así ese aumento de calor.

Por último, mostraremos los resultados del rendimiento de la Raspberry Pi, con los datos de los porcentajes de CPU y RAM. Para mostrarlos, se van a mostrar en Tabla 4-1 las funciones que realizan ya sean individualmente o combinadas y los valores mostrados por la pantalla LCD.

Funciones	CPU	RAM	Evaluación
Cámara	50-60%	17%	Cumple con observaciones
LiDAR	9-15%	17%	Cumple
Cámara + LiDAR	57-72%	19%	Cumple con observaciones
Mapeado	49-54%	19%	Cumple
Mapeado + cámara	86-90%	24%	Cumple con observaciones
Evitación de obstáculos	9-13%	14%	Cumple con observaciones
Navegación con mapa	72-77%	22%	Cumple

Evitación de obstáculos + mapeado	49-54%	20%	No cumple
Diferentes tipos de seguimiento por cámara	43-45%	18%	Cumple con observaciones
Identificación de objetos	53-59%	17%	Cumple

Tabla 4-1 Rendimientos asociados a funciones y tareas llevadas a cabo por el robot

4.3.5 Checklist reproducibles

Con el fin de estandarizar la forma de preparar y operar al ROSMASTER A1 para que realice las tareas que se le encomiende, en este apartado se pretende mostrar un checklist básico en la Tabla 4-2 que incluye los conceptos y la forma de actuar a seguir con el fin de que el alumno o el interesado puede hacerlo funcionar sin necesidad de acudir al manual de *Yahboom* [34].

Proceso que seguir	Cómo llevarlo a cabo	Realizado (Sí/No)
Encendido del robot	Con el interruptor de la placa de control	Sí/No
Conexión de Wifi	Conectando nuestro ordenador con la red “ROSMaster-A1”	Sí/No
Inicio de sesión para entrar en Raspberry Pi	Introducir el nombre de usuario y contraseña	Sí/No
Manejo RC	No cerrar terminal que se abre por defecto al encender el robot y pulsar R2 en el mando	Sí/No
Búsqueda del Docker de trabajo	Abrir una terminal e introducir: docker ps	Sí/No
Iniciar Docker	docker start (nombre Docker)	Sí/No
Entrar en el Docker	sudo docker exec -u root -it (nombre Docker)/bin/bash	Sí/No
Ver funciones de la cámara USB	ros2 pkg executables yahboomcar_astra	Sí/No
Iniciar funciones de la cámara USB	ros2 run (nombre función)	Sí/No
Ver las funciones del LiDAR	ls \$(ros2 pkg prefix yahboomcar_nav)/share/yahboomcar_nav/launch/	Sí/No
Para iniciar las funciones de LiDAR	ros2 launch (nombre función)	Sí/No
Para abrir el programa RViz	ros2 launch yahboomcar_nav display_map_launch.py	Sí/No
Iniciar identificación de objetos	yolo predict model=/root/ultralytics/ultralytics/yolo11n.pt source=0 show=True	Sí/No

Tabla 4-2 Checklist básico para localizar funciones e iniciarlas

4.4 Funciones avanzadas y evidencia de funcionamiento

4.4.1 Mapeado LiDAR 2D

La función de mapeado depende directamente de dos componentes de hardware: el sensor LiDAR, a la hora de recopilar los datos de distancia con respecto a los objetos de su entorno; y el que es vital para plasmar esos datos en un mapa es la IMU de 9 ejes (3 acelerómetros, 3 giroscopios y 3 magnetómetros integrados) que genera la información necesaria para crear el mapa en base a la posición estimada del robot durante el proceso.

Para poner en práctica la función tan solo hay que llevar a cabo dos procesos: la activación del LiDAR y la herramienta de SLAM, y a mayores, para visualizar el robot y lo que detecta con el LiDAR se abre el programa Rviz. Es decir, que no es necesario programar código para realizar estas funciones, pues lo único e indispensable es localizar el archivo ejecutable de esos procesos en el contenedor *docker*. Una vez con todo activado e iniciado, el siguiente hito es el de configurar el programa RViz para lograr tener una visualización óptima; llegados a este punto se recomienda no activar las funciones que puedan prescindibles para la interfaz del usuario con el fin de reducir la carga de trabajo y que el flujo de video tenga continuidad y no se congele la imagen.

Cabe destacar una serie de inconvenientes detectados para tener en cuenta. El primero es que a la hora de controlarlo remotamente con el mando RC, se ha de evitar que el robot se atasque o que se frene durante un instante al chocar con un objeto, ya que si sus motores siguen propulsando en la lectura de datos, se entiende que sigue avanzando, aunque no lo haga y se sobrescriben los datos del LiDAR, generando entonces una imagen errónea del mapa de la zona, mostrado en Figura 4-1, también ocurre lo mismo cuando el robot cambia de rumbo de manera brusca. Por ello se recomienda siempre reducir la velocidad pulsando el botón del mando correspondiente, siguiendo las instrucciones mostradas en Tabla 3-1 en el apartado 3.4.3, para reducir esos errores en la lectura de datos. En segundo lugar, hay que tener en cuenta la cantidad de memoria RAM y CPU que se emplea al ejecutar cada función avanzada, por lo que se recomienda que una vez activado todo, recorrer la zona con el robot y luego abrir el programa Rviz, para mejorar el rendimiento de la Raspberry Pi evitando que tenga que transmitir el vídeo mientras genera el mapa. Además, también es necesario dedicar un determinado tiempo para habituarse al software de RViz y sacar todo su partido, en caso no tener experiencia con él.

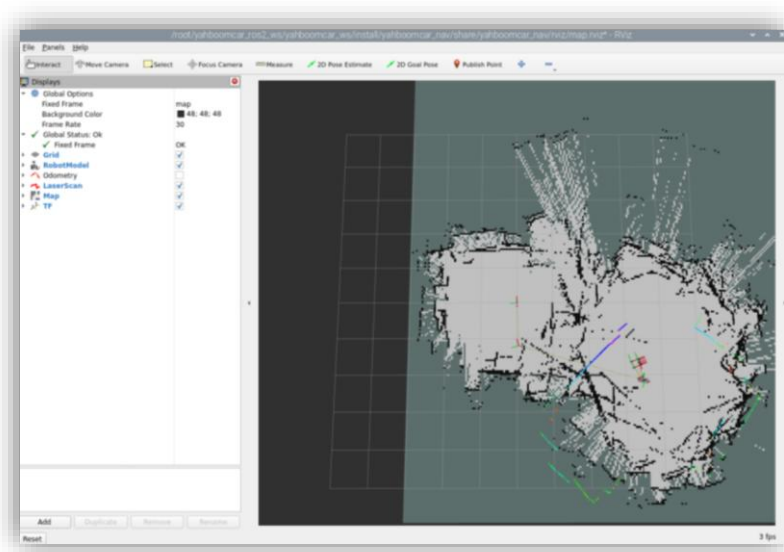


Figura 4-1 Mapeado defectuoso, imagen propia

4.4.2 Navegación

Entendiendo por navegación a la función del robot de desplazarse de forma autónoma hacia una posición determinada, se han analizado dos funciones que lo llevan a cabo: la navegación mediante evitación de obstáculos y mediante un mapa previamente generado. En ambos casos el robot necesita que el LiDAR se encuentre funcionando y que esté ejecutada la función del movimiento autónomo del chasis.

Primero se puso en práctica la evitación de obstáculos, de forma que su proceso para iniciarlo es sencillo, pues solo consta de tres pasos: la activación del LiDAR, del movimiento autónomo del chasis y de la función de navegación, explicado en el apartado 3.5.2. Una vez hecho eso, entonces el robot comenzará a avanzar y cambiará de rumbo hacia su izquierda o derecha en función del dato captado, analizado e interpretado a partir del sensor LiDAR. Si bien funciona puesto que evita obstáculos, es cierto que puede que, debido al código, se ha configurado para evitarlos a una determinada distancia demasiada próxima al robot y además existe unos segundos de retardo entre la detección y la reacción, por lo que si tiene que hacer un cambio brusco de rumbo es posible que no lo realice a tiempo y acabe colisionando contra el objeto, deteniéndose en el sitio, pues los únicos movimientos que realiza el robot en esta función son: avanzar hacia delante y girar hacia la derecha y a la izquierda, es decir, que no puede cambiar el sentido de la marcha para volver a una posición favorable después de un impacto con un obstáculo. Debido a ese resultado, no se recomienda hacer un mapeado de una zona con el robot controlado por esta función, puesto existe una alta probabilidad de que se quede parado y que produzca errores en el mapa resultante.

En cuanto al siguiente método de navegación por medio del uso de un mapa ya generado, el proceso de activación se vio retrasado debido a que la función de generar un mapa del apartado 3.5.1 da como resultado un mapa que no es compatible con este segundo método de navegación, por lo que, empleando un método de prueba y error, se dedujo el motivo de por qué siguiendo el manual de *Yahboom* no se era capaz de activarse. El problema reside en el tipo de documento con el que el mapa se guarda, es decir el mapeado resultaba un mapa cuyo formato es “.yaml” que no es compatible para la navegación autónoma pues necesita que sea de formato “.pbstream”. Una vez solucionado esto generando el mapa con la función adecuada sí que se puede emplear el programa RViz con el mapa cargado y listo para mandarle desplazarse a zonas concretas, siguiendo el proceso desarrollado en el apartado 3.5.2. Se ha obtenido un resultado muy satisfactorio pues el robot una vez que se estima su posición en RViz, éste la actualiza y la corrige en base a los datos captados en el momento mediante el LiDAR; además en caso de colisionar es capaz de invertir rumbo para volver a continuar con su misión.

4.4.3 Visión e IA

Finalmente, tras localizar la ubicación de los archivos de todas las funciones y aplicaciones que se pueden llevar a cabo con la cámara y de haberlas puestos en práctica se han obtenido una serie de resultados tanto favorables como negativos.

Comenzando con los puntos positivos, gracias a la imagen oficial instalada cabe la posibilidad de poner en práctica las funciones mencionadas en 3.5.3 y 3.5.4, existiendo comandos específicos que ejecutan esa función estando el robot en estático, es decir que solamente sigue el objeto con los servos de la cámara, quedando el chasis detenido; y otros para ejecutar las mismas funciones de tal manera que incluso el chasis se mueve para facilitar el seguimiento del objetivo. La activación de estas funciones se puede decir que es sencilla una vez localizados los archivos y los comandos, que además son fáciles de memorizar pues la diferencia entre ellos son las palabras en inglés empleadas para denominar dicha función. En cuanto a la identificación se refiere, es una función que trabaja positivamente, pues genera un recuadro cuyo área incluye al objeto a identificar y lo hace en base a la base de datos instalada mediante la imagen oficial; además, es capaz de diferenciar objetos dentro de recuadros de otros de mayor tamaño.

Continuando con los aspectos negativos, se presentan los relacionados con la calidad de la cámara, con la retransmisión en directo y con la configuración. Con respecto a la calidad de la cámara, cabe destacar que tiene un mal balance de blancos y una exposición automática lenta, es decir, se puede decir que tiene un rango dinámico limitado al ser incapaz de captar detalles en zonas tanto de mucha sombra como luminosas perdiendo a objetivos que se encuentren en esas zonas; además, tarda en ajustar el brillo y el color con los cambios de iluminación y en esos instantes de espera es posible que pierda a su objetivo evitando que el robot finalice su tarea. Por otra parte, tenemos la incidencia relacionada con la retransmisión en directo, una vez que se activa la visión de la cámara para visualizar lo que ve nuestro robot (cuando se trata de las funciones avanzadas de la cámara, la ventana se activa automáticamente), ya que disminuye los FPS de la visión por RealVNC Connect ligeramente, y, además, la imagen se ve con retardo y sin superar los 22 FPS. A esto hay añadir el hecho de que, si el robot se aleja del ordenador del “operario” la calidad y los FPS de la retransmisión se reducen bastante, posiblemente debido a que la red empleada para la retransmisión (la “ROSOMASTER A1” mencionada en el apartado 3.4.1) no sea capaz de dar el soporte de velocidad de transmisión necesaria si la distancia aumenta.

Por último, tenemos el aspecto relacionado con la posible configuración y/o programación, ya que, por ejemplo, sucede que una vez que se fija el objetivo a seguir, la cámara intenta mantenerlo en el centro de su campo de visión. Sin embargo, en un intervalo aproximado de tiempo de 10 a 15 segundos la cámara pierde al objetivo, de forma que, si está activada la navegación, el robot comienza a moverse erráticamente tras ese instante y pierde a su objetivo. A esto se le ha sumar las incidencias por la calidad de la cámara, pues con los movimientos bruscos el robot no es capaz de procesar los cambios de luz y acaba perdiendo el objetivo o continúa desplazándose en búsqueda de éste.

4.4.4 Seguimiento de línea

El seguimiento de línea se encuentra dentro de las funciones y tareas disponibles que nuestra unidad puede llevar a cabo mediante el uso de su cámara. Cabe destacar que se han obtenido unos resultados favorables pues la realiza correctamente y su proceso de activación es sencillo y comprensible para una persona no habituada a ello como un alumno. Además de seguir a líneas sobre el suelo, puede seguir líneas paralelas al suelo que se encuentran en paredes, en ese caso el robot avanza hacia delante paralelo a la pared con la línea enfocada.

Sin embargo, se han detectado una serie de incidencias. La primera es posible que sea debida a la cámara, por el motivo de su calidad que no diferencia con claridad los colores y es muy sensible a los cambios de luz y brillo, por lo que en ocasiones si sobre la línea hay un reflejo de luz el robot interpreta ese cambio de color de forma errónea y comienza a avanzar erráticamente buscando de nuevo a la línea de forma que si esto no llega a ocurrir se detiene. Otra incidencia detectada es que no puede hacer el seguimiento de líneas con curvas muy cerradas o con ángulos rectos, ya que, aunque reconozca la líneas o por la manera en la que se haya programado, sino que el robot al seguirla y colocarse sobre ella en esos casos del diseño de líneas, la propia estructura del robot no permite que la cámara realice el seguimiento simplemente porque la ruedas o parte del chasis se interpone entre la visión de la cámara y la línea en cuestión.

4.4.5 Enjambre

En este apartado se pretende mostrar los resultados obtenidos tras haber conectado ambos robots entre sí. En primer lugar, para la conexión de ambas unidades en la misma red, se acudió a los inicios del tutorial de *Yahboom* [34], pues en el tutorial para aplicar las funciones relacionados con el multi-control de unidades no se especifica como llevarlo a cabo. De esta forma una vez conectadas ambas unidades a la misma red y haber cambiado las direcciones IP (porque de serie ambas tienen la misma IP:192.168.1.11), se puede continuar el tutorial ejecutando los comandos para ello. En cuanto al

ordenador que se emplea para visualizar las Raspberry Pi de cada robot, es posible que se aprecie una ralentización de carga, aunque es prácticamente inapreciable.

En primer lugar, se puso en práctica el control remoto de ambas unidades de manera simultánea mediante un solo mando RC. Tras, denominar en cada terminal dentro del *docker* de trabajo de cada unidad, cual es “robot 1” y “robot 2”, y ejecutar el comando que activa dicha función en uno de los robots, ésta se inicia de manera inmediata. La única incidencia detectada es que en una de las unidades se aprecia que hay un leve retardo al recibir una señal como la de avanzar o cambiar de rumbo en una dirección; de forma que el causante de esto es posible que sea el mando que manda la señal con las órdenes a un robot, al que se encuentra conectado por bluetooth, y a continuación éste se la manda al router, nodo de conexión entre ambas unidades, y lo comparte con el segundo robot conectado a la red, y en ese proceso es posible que se produzca dicho retardo.

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Valoración global del proyecto

El trabajo desarrollado ha permitido abordar con éxito el análisis integral de una plataforma de robótica móvil educativa con capacidades de inteligencia artificial, empleando como caso práctico el kit básico del robot ROSMASTER A1 Large Model IA. La ejecución de este proyecto ha requerido la concurrencia de conocimientos multidisciplinares en los que se incluyen mecánica, electrónica, programación, computación embebida y percepción artificial, que reflejan fielmente el carácter transversal de la robótica moderna. El proceso completo, desde la selección de la plataforma hasta la validación experimental de sus capacidades, ha demostrado que incluso un robot de coste reducido puede ofrecer un entorno de aprendizaje suficientemente rico para desarrollar competencias alineadas con las necesidades de la formación militar contemporánea.

Los objetivos generales y específicos planteados en la introducción se han cumplido satisfactoriamente. La plataforma se ha montado con éxito, se ha configurado la arquitectura software basada en ROS 2 y contenedores *docker*, se han activado y probado todos los sensores incluidos en el kit, y se han evaluado experimentalmente funciones avanzadas de navegación, visión artificial e interacción mediante IA. Asimismo, se han obtenido evidencias empíricas que permiten valorar, con rigor, las capacidades reales del robot y su capacidad para su empleo como recurso didáctico en la futura asignatura Sistemas Autónomos.

El conjunto de tareas realizadas ha permitido comprender de forma profunda la brecha existente, inevitable y formativa, entre el comportamiento teórico de los algoritmos de percepción y navegación y su rendimiento real sobre hardware embebido. Este contraste aporta un valor pedagógico considerable, puesto que muestra a los alumnos las limitaciones reales de la robótica, sus desafíos técnicos y la importancia del análisis crítico al operar sistemas autónomos.

5.2 Conclusiones técnicas

5.2.1 Plataforma y hardware

El ensamblaje del robot ha puesto de manifiesto que el ROSMASTER A1 está diseñado con una clara orientación educativa: los componentes vienen organizados, las piezas mecánicas están mecanizadas con precisión y la estructura de aluminio proporciona rigidez, modularidad y durabilidad. Este diseño facilita que alumnos sin experiencia previa sean capaces de familiarizarse rápidamente con los elementos estructurales de un robot móvil.

No obstante, se han identificado varios puntos de mejora. Los servos de la cámara utilizan engranajes plásticos que pueden desgastarse o dañarse con relativa facilidad, especialmente en contextos docentes donde el robot será manipulado por múltiples usuarios a lo largo del tiempo. El cableado, por su parte, se encuentra expuesto y resulta delicado, lo que puede causar tensiones no deseadas en los conectores, especialmente durante la conexión a la Raspberry Pi o a la placa de expansión. Este aspecto se considera especialmente relevante dado que, en un entorno de prácticas, los alumnos pueden manipular el robot con menos precisión que un usuario experto.

En cuanto a los sensores, el LiDAR 2D ha mostrado un funcionamiento satisfactorio dentro de los márgenes esperables para un sensor de bajo coste, proporcionando información suficiente para mapeado y navegación básicos. Su ubicación elevada en la superestructura es adecuada, pero su rendimiento se ve afectado si la cámara está orientada en un ángulo que obstruye parcialmente la trayectoria del láser. La IMU, integrada en la placa de control, es funcional para estimaciones de orientación y movimiento, pero presenta las limitaciones habituales de estos sensores: deriva progresiva y dependencia de algoritmos de fusión de sensores que no siempre pueden compensar movimientos bruscos.

La cámara, pese a ser idónea para iniciarse en visión artificial, constituye uno de los puntos más débiles de la plataforma debido a su baja resolución, bajo rango dinámico y sensibilidad a cambios de iluminación. Estas deficiencias afectan directamente a gran parte de las funciones de visión avanzada, incluyendo el seguimiento de color, la identificación de objetos y la detección de posturas mediante IA.

5.2.2 Arquitectura software y ecosistema ROS 2

La estructura software basada en una imagen oficial preconfigurada con ROS 2 Humble y contenedores *docker* ha demostrado ser útil y práctica. Esta arquitectura ofrece varias ventajas pedagógicas: por un lado, reduce la probabilidad de errores críticos en la configuración del entorno; por otro, permite comprender conceptos modernos de ejecución de software como la creación de contenedores, que también tienen aplicación en sistemas embarcados reales.

La Raspberry Pi 5 ha mostrado capacidad suficiente para ejecutar los distintos procesos de la plataforma, si bien ha quedado patente que su potencia es limitada cuando se activan simultáneamente el mapeado, la navegación, la visión artificial y la interfaz gráfica remota. Pues en estos casos, la carga de CPU puede alcanzar valores elevados, reduciendo significativamente la fluidez del robot.

El acceso remoto mediante SSH (Putty) o escritorio virtual (RealVNC Connect) ha permitido operar la Raspberry Pi sin necesidad de emplear periféricos físicos, algo esencial en prácticas docentes con robots autónomos. Aunque Putty es funcional y ligero, RealVNC ofrece una experiencia más cercana a la operación “real”, lo que facilita la interacción del alumnado con el entorno gráfico, especialmente con RViz.

Desde un punto de vista didáctico, el uso de ROS 2, estándar de facto en la robótica moderna, constituye una ventaja considerable, ya que introduce al alumnado en conceptos como nodos, tópicos, mensajes, parámetros y herramientas de visualización que luego encontrarán de forma directa en sistemas robóticos profesionales.

5.3 Conclusiones pedagógicas y relevancia para la formación militar

La robótica y la inteligencia artificial tienen un papel central en la evolución operativa de las Fuerzas Armadas. Los vehículos no tripulados entre los que se encuentran aéreos, terrestres y marítimos, emplean tecnologías muy similares a las utilizadas en este proyecto como lo son el LiDAR, la navegación autónoma, los sensores inerciales, las cámaras, la visión artificial y algoritmos de IA ejecutados en tiempo real. Por ello, entender su funcionamiento constituye un componente esencial de la formación de un oficial de la era contemporánea.

El ROSMASTER A1 ofrece un entorno controlado, modular, seguro y económicamente asumible para introducir a los alumnos en esas tecnologías mediante prácticas reales. Gracias a él, los alumnos pueden:

- Comprender cómo se fusionan datos de sensores para interpretar el entorno.
- Observar las limitaciones reales de los algoritmos frente a su formulación teórica.
- Adquirir hábitos de verificación, diagnóstico y resolución de incidencias.
- Relacionar directamente estas prácticas con los sistemas autónomos empleados en la Armada.
- Desarrollar una base técnica que facilite su comprensión futura de plataformas más complejas.

Además, el proyecto se encuentra alineado con el nuevo plan de estudios y con la futura asignatura de Sistemas Autónomos, que buscará dotar al alumno de conocimientos sólidos en navegación, percepción y control de robots en entornos dinámicos.

5.4 Limitaciones detectadas

El análisis experimental y el uso continuado del robot han revelado varias limitaciones que condicionan su uso docente:

- La cámara integrada presenta un rendimiento inferior al esperado, especialmente en condiciones de iluminación variables.
- La carga computacional combinada de SLAM, navegación y visión puede saturar la Raspberry Pi, ralentizando funciones críticas.
- El cableado expuesto y la fragilidad de algunas conexiones hacen aconsejable una manipulación cuidadosa.
- El sistema de mapeado produce errores si el robot experimenta frenadas, choques o cambios bruscos de velocidad.
- La conectividad inalámbrica puede deteriorarse notablemente con la distancia o la presencia de obstáculos físicos.
- La duración de la batería, aunque adecuada, se reduce significativamente en uso intensivo de IA o navegación compleja.

Estas limitaciones no invalidan la utilidad pedagógica del robot, pero sí deben ser conocidas de antemano para gestionar las expectativas y organizar las prácticas de manera adecuada.

5.5 Propuestas de mejora

5.5.1 Mejoras técnicas

Con el fin de aprovechar y continuar con el uso de la unidad ROSMASTER A1, se proponen una serie de mejoras en cuanto a su diseño del chasis, del hardware y de la configuración de software. Entre los que se encuentran:

- Sustitución de la cámara por un modelo de mayor resolución o incluso una cámara RGB-D, adquiriendo así la capacidad de detectar la distancia a la que se encuentra un objeto que ve el robot.
- Diseño de una cubierta protectora para el cableado de la superestructura, para evitar accidentes que puedan afectar al estado de sus conexiones.
- Optimización del uso de la Raspberry mediante ajustes en la frecuencia de publicación de sensores y en el consumo de CPU.
- Empleo de una microSD de alta velocidad para reducir tiempos de carga.

- Añadido de sensores complementarios (ultrasonidos, infrarrojos, GPS para exteriores), que le doten de la capacidad de detectar baches/ alteraciones en el terreno, que le permita posicionarse, seguir rutas con un método más preciso, etc.
- Evaluación de redes neuronales más ligeras que YOLO11 para reducir carga computacional.

5.5.2 Mejoras docentes

Para la aplicación e incorporación docente de este modelo de plataforma de robot móvil educativo o similares, se presentan una serie de posibles mejoras para la comprensión, entendimiento y ejecución de las bases teóricas que se desarrollarán e impartirán en la asignatura de sistemas autónomos.

En esas mejoras se incluyen la creación de una guía detallada de prácticas con ejercicios progresivos, en los que a medida que se vaya avanzado y aumentando el nivel de complejidad de éstos, se experimente inicialmente en el uso de cada sensor de manera individual, para luego aprender a realizar su posterior integración conjunta. Por otra parte, implementar también sesiones específicas donde se analicen fallos reproducibles del robot para conocer los puntos débiles de estos sensores que se encuentran también en la industria de guerra.

Con el fin de que se muestren en las sesiones de prácticas el fruto de las bases teóricas aprendidas previamente, se recomienda llevar a cabo sesiones que se asemejen lo máximo posible, de forma académica, a las aplicaciones reales de estos dispositivos y sensores en el ámbito militar. Como ejemplos de estas aplicaciones de un vehículo terrestre no tripulado operado por la Infantería de Marina Española en una zona sobre la que se ha proyectado la fuerza naval, que puede llevar a cabo el ROSMASTER A1 encontramos:

- Un UGV de carga que realiza el seguimiento de un soldado por medio de su silueta.
- Cumplimiento de órdenes procedentes de un soldado transmitidas por gestos.
- Realizar un mapeado de una zona desconocida para llevar a cabo un primer reconocimiento y planificación de una misión. Tanto de manera autónoma como por control remoto.
- Introducir el UGV en un área cuyo plano se conoce con el fin de recopilar información de los objetos detectados en una zona concreta del área, sin necesidad de enviar a un soldado por alto nivel de amenaza que supone.
- Identificación de caras de sujetos para su posterior seguimiento.
- Enlazar y enviar varias unidades en modo de enjambre para lograr un mayor éxito en el cumplimiento de la misión y saturar al enemigo.

5.6 Líneas futuras

El proyecto abre múltiples líneas de desarrollo futuro que permitirán expandir el conocimiento básico y esencial para su aprendizaje, explotación de recursos y utilidad docente:

- Integración de sensores avanzados como cámaras 3D, LiDAR 3D o módulos térmicos capaces de detectar e identificar siluetas térmicas.
- Desarrollo de ejercicios de enjambre con dos o más robots trabajando de forma coordinada, entre solamente UGVs, UAVs y/o combinando diferentes tipos de plataformas.
- Migración a plataformas de mayor capacidad para niveles avanzados del grado como el empleo de la plataforma Jetson Nano por su mayor y mejor capacidad de rendimiento frente a la Raspberry Pi 5.
- Implementación de un sistema mixto de simulación, en donde se integre el robot real para prácticas híbridas y poder realizar simulaciones previas a las misiones reales con los robots.
- Diseño de una plataforma desde su base para misiones complejas donde los alumnos deban combinar mapeado, detección e identificación de objetivos y navegación autónoma, enfocados al ámbito militar.

- Optimización del uso de IA embarcada mediante modelos más recientes y eficientes, para identificación de objetos, gestos y situaciones en las que se analice el entorno de la plataforma con el fin de que ejecute acciones de manera autónoma.

5.7 Conclusión

El ROSMASTER A1 se ha revelado como una plataforma educativa idónea para introducir al alumnado en la robótica móvil y los sistemas autónomos. A pesar de sus limitaciones, propias de un robot de coste reducido, proporciona un entorno de aprendizaje realista, seguro y técnicamente relevante. El proyecto ha permitido comprender en profundidad los procesos de montaje, configuración y validación de un robot educativo con IA, aportando una base sólida para la futura implementación de prácticas docentes relacionadas con percepción artificial, navegación autónoma y fusión sensorial.

La experiencia adquirida confirma que esta plataforma puede servir como punto de partida para el desarrollo de nuevos proyectos, prácticas y líneas de investigación orientadas a mejorar la formación técnica de los futuros oficiales de la Armada, quienes deberán enfrentarse a sistemas autónomos cada vez más complejos y omnipresentes en el entorno operativo moderno.

6 BIBLIOGRAFÍA

- [1] «mBot 2 - El nuevo robot educativo de Makeblock», ROBOTIX. Accedido: 28 de febrero de 2026. [En línea]. Disponible en: <https://www.robotix.es/es/mbot-2>
- [2] «Home», Thymio. Accedido: 28 de febrero de 2026. [En línea]. Disponible en: <https://www.thymio.org/>
- [3] «Features · User Manual». Accedido: 28 de febrero de 2026. [En línea]. Disponible en: <https://turtlebot.github.io/turtlebot4-user-manual/overview/features.html>
- [4] «Clearpath Robotics: Mobile Robots for Research & Development», Clearpath Robotics. Accedido: 28 de febrero de 2026. [En línea]. Disponible en: <https://clearpathrobotics.com/>
- [5] «Fabricante de Robots - Robótica Móvil | Robotnik®». Accedido: 28 de febrero de 2026. [En línea]. Disponible en: <https://robotnik.eu/es/>
- [6] «ROSbot XL - Indoor AMR | Husarion». Accedido: 28 de febrero de 2026. [En línea]. Disponible en: <https://husarion.com/manuals/rosbot-xl/overview/>
- [7] «Página oficial de Tello - Shenzhen Ryze Technology Co., Ltd». Accedido: 2 de marzo de 2026. [En línea]. Disponible en: <https://www.ryzerobotics.com/es/tello-edu>
- [8] «★ Comprar Robomaster TT Tello Talent», DJI MADRID Las Rozas. Accedido: 2 de marzo de 2026. [En línea]. Disponible en: <https://stockrc.com/es/robomaster-s1/2870-robomaster-tt-tello-talent.html>
- [9] «Crazyflie 2.1 Brushless», Drone Prix S.L. Accedido: 2 de marzo de 2026. [En línea]. Disponible en: <https://droneprix.es/bitcraze/1058-crazyflie-22.html>
- [10] «PX4 Development Kit - X500 v2», Holybro Store. Accedido: 2 de marzo de 2026. [En línea]. Disponible en: <https://holybro.com/products/px4-development-kit-x500-v2>
- [11] «BlueBoat Uncrewed Surface Vessel», Blue Robotics. Accedido: 2 de marzo de 2026. [En línea]. Disponible en: <https://bluerobotics.com/store/boat/blueboat/blueboat/>
- [12] «Heron USV Tutorials — Heron Tutorials 0.1.1 documentation». Accedido: 2 de marzo de 2026. [En línea]. Disponible en: <https://www.clearpathrobotics.com/assets/guides/melodic/heron/index.html>

- [13] «New 3D printed Hardware - Build / DIY USVs», Blue Robotics Community Forums. Accedido: 2 de marzo de 2026. [En línea]. Disponible en: <https://discuss.bluerobotics.com/t/new-3d-printed-hardware/10712>
- [14] «OpenROV Trident - An Underwater Drone for Everyone», Kickstarter. Accedido: 2 de marzo de 2026. [En línea]. Disponible en: <https://www.kickstarter.com/projects/openrov/openrov-trident-an-underwater-drone-for-everyone>
- [15] «BlueROV2», Blue Robotics. Accedido: 2 de marzo de 2026. [En línea]. Disponible en: <https://bluerobotics.com/store/rov/bluerov2/>
- [16] «iRoboBumper Project | Mouser». Accedido: 5 de marzo de 2026. [En línea]. Disponible en: <https://www.mouser.es/applications/invisible-robotics-bumper-project-page-1/>
- [17] The4, «OAK-D», Luxonis. Accedido: 7 de marzo de 2026. [En línea]. Disponible en: <https://shop.luxonis.com/products/oak-d>
- [18] «IMX219-160 Camera 8MP 160 Degree FOV for Jetson Nano and AI Projects», ThinkRobotics.com. Accedido: 7 de marzo de 2026. [En línea]. Disponible en: <https://thinkrobotics.com/products/imx219-160-camera-with-160-fov-for-jetson-nano-online>
- [19] «HM01B0-MNA-00FT870», DigiKey Electronics. Accedido: 7 de marzo de 2026. [En línea]. Disponible en: <https://www.digikey.es/es/products/detail/himax/HM01B0-MNA-00FT870/14109821>
- [20] «Bitcraze AI Deck 1.1 for Crazyflie 2.X». Accedido: 7 de marzo de 2026. [En línea]. Disponible en: <https://www.tinytronics.nl/en/platforms-and-systems/bitcraze-crazyflie/bitcraze-ai-deck-1.1-for-crazyflie-2.x>
- [21] «Infrarrojos», Protexam. Accedido: 7 de marzo de 2026. [En línea]. Disponible en: <https://protexam.wordpress.com/sensores-infrarrojos/>
- [22] «Crazyflie Z-ranger Deck v2 - Distance measurement downwards (up to 4m)», EXP Tech. Accedido: 7 de marzo de 2026. [En línea]. Disponible en: <https://exp-tech.de/en/products/crazyflie-z-ranger-deck-v2>
- [23] «Buscador de Líneas IR MakeBlock Me», RobotShop. Accedido: 7 de marzo de 2026. [En línea]. Disponible en: <http://www.robotshop.com/es/products/buscador-de-lineas-ir-makeblock-me>
- [24] «The INS structure with three-axis sensors mounted inside IMU and...», ResearchGate. Accedido: 7 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/The-INS-structure-with-three-axis-sensors-mounted-inside-IMU-and-outputs-of-INS_fig13_343735738
- [25] «MPU-9250 TDK InvenSense | Mouser», Mouser Electronics. Accedido: 7 de marzo de 2026. [En línea]. Disponible en: <https://www.mouser.es/ProductDetail/TKD-InvenSense/MPU-9250?qs=u4fy%2FsgLU9OhGjFkQSZssA%3D%3D>
- [26] «ICM-42688-P TDK InvenSense | Mouser», Mouser Electronics. Accedido: 7 de marzo de 2026. [En línea]. Disponible en: <https://www.mouser.es/ProductDetail/TKD-InvenSense/ICM-42688-P?qs=OIC7AqGiEDk8QrNAV728zw%3D%3D>
- [27] «¿Cómo funciona el LiDAR? Tecnología láser de YellowScan», YellowScan. Accedido: 7 de marzo de 2026. [En línea]. Disponible en: <https://www.yellowscan.com/es/knowledge/how-does-lidar-work/>
- [28] «RPLidar A1M8 - Kit de Desarrollo de Escáner Láser de 360 Grados», RobotShop. Accedido: 7 de marzo de 2026. [En línea]. Disponible en: <http://eu.robotshop.com/es/products/rplidar-a1m8-kit-de-desarrollo-de-escaner-laser-de-360-grados>

- [29] «Lidar Robot ROS + SLAM - Proje Hocam». Accedido: 7 de marzo de 2026. [En línea]. Disponible en: <https://www.projehocam.com/portfoy/lidar-robot-ros-slam/>
- [30] «YOLO11: Visión por ordenador en robótica | Ultralytics». Accedido: 7 de marzo de 2026. [En línea]. Disponible en: <https://www.ultralytics.com/es/blog/integrating-computer-vision-in-robotics-with-ultalytics-yolo11>
- [31] «rgmmar2024_Parte04» Documento interno.
- [32] « **【Unboxing and Reviewing】** ---ROSMASER A1 AI Large Model ROS2 Robot with Ackerman steering chassis», Yahboom. Accedido: 2 de marzo de 2026. [En línea]. Disponible en: <https://category.yahboom.net/es/blogs/news/unboxing-and-reviewing-rosmaster-a1-ai-large-model-ros2-robot-with-ackerman-steering-chassis>
- [33] R. Pérez, «¿Qué es el ángulo Ackermann y cómo mejora la conducción?», El Motor. Accedido: 2 de marzo de 2026. [En línea]. Disponible en: <https://motor.elpais.com/conducir/que-es-el-angulo-ackermann-y-como-mejora-la-conduccion/>
- [34] «Yahboom». Accedido: 2 de marzo de 2026. [En línea]. Disponible en: <https://www.yahboom.net/study/ROSMASER-A1>
- [35] R. P. Ltd, «Raspberry Pi software», Raspberry Pi. Accedido: 2 de marzo de 2026. [En línea]. Disponible en: <https://www.raspberrypi.com/software/>
- [36] «Download PuTTY - a free SSH and telnet client for Windows». Accedido: 2 de marzo de 2026. [En línea]. Disponible en: <https://putty.org/index.html>
- [37] «Download RealVNC® Connect», RealVNC®. Accedido: 2 de marzo de 2026. [En línea]. Disponible en: <https://www.realvnc.com/en/connect/download/>
- [38] «128GB Tarjeta SanDisk Extreme PRO microSDXC™ UHS-I», Sandisk. Accedido: 9 de marzo de 2026. [En línea]. Disponible en: <https://www.sandisk.com/es-es/products/memory-cards/microsd-cards/sandisk-extreme-pro-uhs-i-microsd.sku=SDSQXCG-032G-GN6MA>

ANEXO I: SIGLAS Y ACRÓNIMOS

En este anexo se pretende reunir las siglas y acrónimos empleados en el documento, con su significado y una explicación breve en el contexto de la robótica móvil educativa. Su finalidad es la de facilitar la lectura de los capítulos del TFG en los que se abordan conceptos como la navegación autónoma, percepción y mapeado. Tal y como se aprecia en la Tabla A1-1.

Sigla	Significado	Explicación / Uso
APF	Artificial Potential Fields (Campos de Potenciales Artificiales)	Método reactivo: el objetivo atrae y los obstáculos repelen; simple, pero con mínimos locales.
AUV	Autonomous Underwater Vehicle (Vehículo Submarino Autónomo)	Submarino autónomo para misiones de exploración y mapeo.
BlueOS	Blue Robotics Operating System	Sistema basado en Linux para gestión de vehículos marinos y sensores de Blue Robotics.
CPU	Central Processing Unit (Unidad Central de Proceso)	Procesador principal del sistema embebido (p. ej., Raspberry Pi).
DC	Direct Current (Corriente Continua)	Motores y alimentación CC comunes en plataformas educativas.
DJI	Da-Jiang Innovations	Fabricante de drones/robótica educativa (Tello EDU, RoboMaster TT) y SDKs educativos.
DOF	Degrees of Freedom (Grados de Libertad)	Número de ejes independientes de movimiento/medición de un sistema o sensor.
DWA	Dynamic Window Approach	Planificador local reactivo que evalúa velocidades factibles y selecciona la trayectoria segura/eficiente.
DWB	DWB (Dynamic Window) Local Planner	Implementación en Nav2 del enfoque DWA; calcula comandos de velocidad en tiempo real.
EPFL	École Polytechnique Fédérale de Lausanne	Universidad suiza vinculada al desarrollo de Thymio/MOBOTS.

FPS	Frames Per Second (Fotogramas por Segundo)	Frecuencia de imágenes de la cámara o de renderizado de visualización.
GPS	Global Positioning System	Sistema de posicionamiento satelital usado para navegación en exterior.
GPU	Graphics Processing Unit (Unidad de Procesamiento Gráfico)	Acelera operaciones paralelas, útil para visión e IA.
HD	High Definition	Calidad de vídeo/imagen 720p/1080p mencionada en cámaras del TFG.
IA	Inteligencia Artificial	Área que dota a sistemas de capacidades de percepción, decisión y aprendizaje.
IMU	Inertial Measurement Unit (Unidad de Medición Inercial)	Sensor que fusiona acelerómetros, giróscopos y, a veces, magnetómetro para estimar orientación y movimiento.
IP	Internet Protocol (Dirección IP)	Identificador de red del robot para conexión SSH/VNC y comunicación ROS.
IR	Infrared (Infrarrojo)	Tecnología/ sensores que operan en el espectro infrarrojo para proximidad, seguimiento de línea o estimación de altura.
LCD	Liquid Crystal Display (Pantalla de Cristal Líquido)	Pantalla integrada en el robot para mostrar estado (IP, CPU, RAM, etc.).
LED	Light-Emitting Diode (Diodo Emisor de Luz)	Indicadores luminosos o matrices LED para feedback y docencia.
LiDAR	Light Detection And Ranging	Sensor que mide distancias con pulsos láser para generar mapas 2D/3D y evitar obstáculos.
Nav2	Navigation2	Conjunto de paquetes de ROS 2 para navegación: planificación global/local, mapas, localización y control.

OAK-D	OpenCV AI Kit – Depth	Módulo de cámara con VPU integrada (Myriad X) para visión e IA con profundidad.
ONNX	Open Neural Network Exchange	Formato abierto para intercambiar modelos entre frameworks.
PTZ	Pan-Tilt-Zoom	Montura de cámara con servo-panorámica e inclinación; en el robot se usa pan/tilt.
PX4	Open-source Autopilot	Stack de control de vuelo de código abierto para drones y robots móviles.
RAM	Random Access Memory (Memoria de Acceso Aleatorio)	Memoria volátil empleada por el sistema y los procesos durante la ejecución.
RC	Radio Control / Remote Control	Mando inalámbrico para teleoperación y pruebas básicas de movimiento.
RGB	Red-Green-Blue	Modelo de color; en LEDs/cámaras indica canales de color.
RGB-D	Red-Green-Blue + Depth	Cámaras que proporcionan imagen en color y mapa de profundidad sincronizado.
ROS	Robot Operating System	Ecosistema de middleware y herramientas para robótica. Proporciona comunicación entre nodos, paquetes y utilidades.
ROS 2	Robot Operating System 2	Evolución de ROS con arquitectura basada en DDS, mayor seguridad y soporte en tiempo real. Base del stack Nav2.
ROV	Remotely Operated Vehicle (Vehículo Operado Remotamente)	Vehículo submarino controlado por cable o radio; típico en inspección y docencia de control.
RTH	Return To Home	Función de seguridad que ordena regresar al punto de partida automáticamente.
RViz	ROS Visualization	Herramienta 3D de visualización y depuración de

		datos de sensores, mapas y estados de ROS/ROS 2.
SDK	Software Development Kit	Conjunto de librerías, ejemplos y herramientas para desarrollar sobre una plataforma (p. ej., DJI, Luxonis).
SLAM	Simultaneous Localization and Mapping	Algoritmos que estiman la pose del robot mientras construyen un mapa del entorno de forma simultánea.
SSH	Secure Shell	Protocolo para acceso remoto seguro a la Raspberry Pi mediante terminal.
TEB	Timed Elastic Band	Planificador local que optimiza una banda de poses en el tiempo para generar trayectorias suaves y respetar restricciones.
TensorFlow Lite	TensorFlow Lite (TFLite)	Runtime y herramientas para ejecutar modelos de TensorFlow en dispositivos edge (Raspberry Pi, móviles).
ToF	Time of Flight (Tiempo de Vuelo)	Técnica de medición de distancia por el tiempo que tarda la señal en ir y volver (ultrasonidos/láser).
UAV	Unmanned Aerial Vehicle (Vehículo Aéreo No Tripulado)	Dron o aeronave sin piloto a bordo. Usado para docencia y pruebas de navegación/autopiloto.
UGV	Unmanned Ground Vehicle (Vehículo Terrestre No Tripulado)	Plataforma terrestre autónoma o teleoperada. Base típica para ejercicios de SLAM y Nav2.
USB	Universal Serial Bus	Bus de conexión de periféricos (cámaras, LiDAR, adaptadores) al SBC.
USV	Unmanned Surface Vehicle (Vehículo de Superficie No Tripulado)	Embarcación no tripulada que opera sobre la superficie del agua; útil para batimetría y vigilancia.

UXV	Uncrewed/Unmanned Vehicle	Término paraguas para UAV/UGV/USV/AUV (plataformas no tripuladas).
YOLO	You Only Look Once	Familia de redes de detección de objetos en tiempo real; existen variantes ligeras (tiny/nano).

Tabla A1-1 Lista de siglas y acrónimos del proyecto, con su significado y explicación de uso en el contexto de la robótica móvil educativa

ANEXO II: IMPLICACIONES SOCIALES, ECONÓMICAS, Y AMBIENTALES

Este anexo recoge, de forma estructurada, la conciencia y las medidas adoptadas en el proyecto para atender a los criterios sociales, de salud y seguridad, ambientales y económicos/industriales, de acuerdo con el resultado del aprendizaje. El trabajo consiste en el montaje, configuración y evaluación experimental de un robot móvil educativo con herramientas de IA para percepción y navegación, orientado a su uso docente con oficiales alumnos de la Armada.

La implantación de un robot móvil educativo con herramientas de inteligencia artificial en el aula se justifica, ante todo, por su finalidad formativa y por el interés público asociado a una capacitación rigurosa y segura del alumnado en tecnologías autónomas de uso creciente en ámbitos civiles y militares. Al trabajar en un entorno académico controlado, con objetivos de aprendizaje explícitos, secuencias didácticas graduales y supervisión docente constante, la práctica reduce la distancia entre la teoría y la experiencia con sensores y actuadores reales, al tiempo que disminuye riesgos y favorece hábitos profesionales de verificación, registro y mejora continua; la teleoperación con mando, la delimitación del área de ensayos, la verificación previa del cableado y las paradas seguras conforman una arquitectura pedagógica que eleva la calidad de la formación práctica y que responde a la necesidad institucional de egresar oficiales y técnicos con criterio, prudencia y competencias operativas contrastadas. En este marco, el robot no es un fin en sí mismo, sino una herramienta que facilita la observación, el análisis y la toma de decisiones en condiciones reproducibles y trazables, de modo que el aprendizaje técnico quede inseparablemente unido a la reflexión ética sobre el impacto de estas tecnologías y al compromiso con su uso responsable en contextos reales. Esta orientación es coherente con la memoria técnica y sus anexos, donde se ordenan los procedimientos y evidencias que refuerzan la coherencia entre objetivos, recursos y resultados, y donde se privilegia una enseñanza con enfoque de seguridad operacional, trazabilidad y evaluación transparente.

El respeto a la privacidad y la protección de datos constituye una condición de diseño, especialmente cuando la plataforma integra cámara y micrófono para funciones de percepción y retroalimentación didácticas. En las prácticas no se graba ni se almacena material audiovisual identificable: la señal se utiliza exclusivamente en tiempo real y sin persistencia, evitando cualquier cesión o reutilización que exceda el propósito docente inmediato; solo si fuese imprescindible obtener evidencia para evaluación u auditoría pedagógica se activaría una captura limitada y proporcional, aplicando minimización, difuminado de rostros, información y aviso previo a los participantes y borrado seguro al término de la finalidad. Este principio se complementa con controles técnico-organizativos razonables en el contexto de laboratorio: segmentación lógica de la red del robot, desactivación de servicios no requeridos, gestión de identidades y permisos por rol, cifrado en tránsito cuando proceda y registro de auditoría del ciclo de vida de cualquier dato sensible; además, se excluyen prácticas de reconocimiento biométrico o identificación personal, manteniendo siempre el humano en el bucle para decisiones relevantes. El resultado es un entorno donde la confianza no descansa en la buena voluntad, sino en garantías verificables de privacidad por defecto y desde el diseño, alineadas con el carácter académico del proyecto y con la documentación de referencia elaborada en la propia memoria y sus anexos.

El enfoque social del proyecto se completa con un compromiso firme con la inclusión y la accesibilidad, de manera que todo el alumnado, con independencia de su punto de partida o de necesidades específicas, pueda participar y progresar. Las interfaces gráficas remotas facilitan la interacción desde distintos dispositivos y escenarios de aula; la teleoperación con control RC ofrece una vía de entrada intuitiva mientras se adquiere soltura con la pila de software; los manuales paso a paso, los checklists de arranque y cierre, las guías de resolución de problemas y las rúbricas de evaluación transparentes permiten clarificar expectativas, asegurar una participación informada y reducir barreras

técnicas en los primeros hitos. En paralelo, la docencia integra discusiones sobre sesgos algorítmicos, equidad y fiabilidad de modelos ligeros de IA, reforzando la lectura crítica de resultados, la validación en escenarios diversos y la comunicación honesta de limitaciones, lo que dota a la práctica de un espesor ético indispensable para el uso responsable de estas tecnologías.

La tutela de la salud y la seguridad se despliega con un conjunto de precauciones proporcionadas a la naturaleza del sistema y de la actividad: la batería de litio se carga con el cargador específico, bajo supervisión y sobre superficie ignífuga; se inspecciona visualmente su estado para detectar hinchazones, olores anómalos o incrementos de temperatura; se evita operar si hay signos de daño compresión o cortocircuitos. Aunque el sistema trabaje a baja tensión, las intervenciones en la electrónica se realizan sin energizar; la gestión ordenada del cableado disminuye fallos intermitentes y riesgos de contacto. En el plano mecánico, se extreman las precauciones ante bordes y elementos móviles de la cámara, manteniendo dedos, cabello y ropa fuera del recorrido. El sensor de distancia, aun siendo de clase baja, no debe mirarse de cerca ni orientarse a la altura de los ojos; fijarlo con solidez y en orientación adecuada reduce molestias o deslumbramientos. La operación en laboratorio acota el área de pruebas con señalización visible, retira obstáculos y establece procedimientos de parada segura con telemetría disponible; el orden y la limpieza, recogida de cables, marcaje de perímetros, son tan importantes como el control del robot, porque minimizan tropiezos y golpes. En conjunto, estas medidas forman una malla de seguridad razonable que permite experimentar con confianza, registrar incidencias y extraer lecciones para perfeccionar tanto la docencia como el diseño del prototipo, tal como se documenta en la memoria y en los resultados.

La perspectiva ambiental y económica refuerza el carácter responsable del proyecto. Al término de su vida útil, placas, sensores, cables, baterías y soportes se entregan en puntos RAEE, Residuos de Aparatos Eléctricos y Electrónicos, autorizados; durante el curso se evitan residuos innecesarios, priorizando mantenimiento preventivo, reutilización y reparación, limpieza óptica del sensor, reapriete de tornillos, sustitución de conectores antes que reemplazar conjuntos completos y si se fabrican carcasas o protecciones, se prefieren polímeros reciclables y diseños que faciliten el desmontaje y la segregación de materiales. El consumo energético de una sesión tipo se mantiene contenido, con un ordenador mono placa de referencia que oscila en cargas de trabajo habituales, un sensor de distancia de consumo moderado y electrónica auxiliar; la planificación por turnos, el cierre de servicios no necesarios y la racionalización de visualizaciones exigentes ayudan a evitar picos sin penalizar la experiencia didáctica. En términos de coste total de propiedad, el desembolso inicial por unidad es acotado y permite reservar una pequeña partida para microSD de alta velocidad y repuestos de servos y cableado; el coste operativo real se vuelve competitivo cuando se consolida una documentación viva de contenedores, procedimientos, checklists y lecciones aprendidas que acortan drásticamente los tiempos de puesta en marcha en cursos sucesivos. La interoperabilidad de la pila, ROS 2, contenedores y hardware estándar, limita la dependencia de proveedores, facilita sustituciones razonadas de sensores equivalentes y enseña herramientas con demanda real en la industria, lo que racionaliza el coste total y fortalece el valor formativo del proyecto, tal y como se refleja en las secciones de desarrollo y software de la memoria.

ANEXO III: REFLEXIONES ÉTICAS Y SOCIALES

Este anexo presenta una reflexión ética y social alineada con el resultado de aprendizaje que pretende algo más que una declaración de principios: busca dotar al estudiantado de un marco de comprensión para evaluar implicaciones, anticipar riesgos y decidir con prudencia cuando la tecnología, y en particular la robótica con inteligencia artificial se traslada del laboratorio al mundo real. En este proyecto, la plataforma se utiliza con fines docentes e investigadores, pero su potencial de doble uso obliga a examinar con rigor qué funciones se enseñan, en qué condiciones se prueban y bajo qué límites se validan. Sostener esa mirada no es accesorio: es el cimiento para que el dominio técnico coexista con una conciencia viva de contexto, consecuencias y responsabilidades. El aprendizaje se ancla, así, en prácticas guiadas y secuenciadas, en un entorno controlado donde cada decisión queda trazada y cada resultado es interpretable a la luz de propósitos formativos, evitando que la destreza instrumental se desconecte de la reflexión sobre el bien público. Esta orientación doctrinal no nace en el vacío: se integra en la arquitectura pedagógica y documental del proyecto, con cuadernos de pruebas, evidencias de reproducibilidad y guías operativas, que el estudiantado consulta y actualiza como parte de su propia práctica profesional incipiente. De ese modo, lo ético y lo social no son un apéndice teórico, sino una forma de trabajar que se aprende haciéndola.

Reconocer el carácter de doble uso significa admitir que una misma base tecnológica puede servir a fines civiles valiosos como la formación, investigación aplicada, mejora de procesos, y a fines militares legítimos o, en el extremo, a propósitos reprobables si se descontextualiza. Por eso el proyecto adopta una noción acotada de propósito, compatible con la distinción, la proporcionalidad y la necesidad pues se proponen únicamente escenarios seguros, simulados y no letales; y se lleva un control docente de cambios para verificar que ninguna evolución del prototipo habilite, directa o indirectamente, mecanismos de daño. La finalidad formativa no solo acota lo que se hace, sino cómo se hace: se opera con velocidades reducidas, áreas delimitadas, paradas seguras y teleoperación disponible, y se documenta por qué ciertos experimentos quedan fuera del alcance académico. De esta forma, el alumnado interioriza que la evaluación de un sistema autónomo no se limita a si “funciona”, sino a si su despliegue es compatible con un estándar exigente de conducta responsable. Frente a la tentación de confundir lo técnicamente factible con lo moralmente deseable, el proyecto normaliza la pregunta por el coste humano, ambiental y social de cada capacidad, cultivando un criterio que no se deje deslumbrar por la novedad técnica.

La dignidad y la privacidad se protegen desde el diseño, no como remiendo ex post. La presencia de cámara y microfonía exige una regla clara: no se graba ni se almacena material identificable por defecto, y las señales se emplean exclusivamente en tiempo real para fines de percepción y control; cuando, por razones estrictamente pedagógicas, fuese imprescindible registrar evidencia, se aplica una combinación proporcional de minimización (capturar solo lo indispensable, durante el menor tiempo viable y en la resolución necesaria), anonimización o seudonimización (difuminado de rostros, supresión de metadatos y de rasgos distintivos) y limitación temporal con borrado seguro y verificado. Ese tratamiento se realiza en espacios de práctica controlados, con información previa comprensible para todos los participantes y con un registro de quién autorizó, con qué propósito y por cuánto tiempo se conservan los datos. La visión por computador se limita a detección genérica de formas, colores u obstáculos, y las decisiones con potencial impacto sobre personas o entornos se mantienen bajo validación humana. El objetivo es doble: preservar la confianza del aula como espacio de aprendizaje y evitar que la potencia de la herramienta erosione la expectativa razonable de privacidad que asiste a estudiantes y docentes. Este enfoque, coherente con la lógica de “privacidad por defecto y desde el diseño”, se refleja en las guías de operación y en el rastro documental que acompaña a cada práctica.

Respecto a los sesgos algorítmicos, el proyecto asume que los modelos ligeros, los que posibilitan detección o seguimiento en hardware contenido, no son neutros: heredan las virtudes y carencias de sus datos de entrenamiento y de las condiciones de operación. Por ello, se analizan errores cualitativamente buscando patrones que revelen fallos sistemáticos (por ejemplo, degradaciones ante cambios de iluminación, texturas o colores dominantes). Además, se contrastan resultados en escenarios variados para evitar sobreajuste a un único “pasillo ideal” de laboratorio, y se registra con transparencia lo que el sistema hace bien, lo que hace peor y lo que no hace. Ese registro, unido a métricas básicas de rendimiento, enseña a comunicar limitaciones con honestidad y a decidir cuándo conviene simplificar la tarea o devolver la decisión al operario. La autonomía responsable queda, así, anclada en una operativa diaria: control remoto disponible, límites de velocidad y de área, y protocolos de validación incremental cuando se contemplan ampliaciones, como pruebas en exteriores. Estos hábitos, documentados e inspeccionables, enseñan que la fiabilidad no es una propiedad abstracta del algoritmo, sino una cualidad emergente de un proceso de diseño, prueba y revisión que mantiene siempre la seguridad y la equidad como criterios de salida. La coherencia de esta metodología con la organización del software en contenedores y con las bitácoras de pruebas refuerza, además, la reproducibilidad necesaria para auditar sesgos y progresos.

La transparencia y la trazabilidad se garantizan con disciplina de ingeniería: versionado de código y contenidos, inventario de parámetros de configuración, cadenas de experimentos descritas con suficiente granularidad para ser repetidas por terceros, y encapsulamiento de dependencias en contenedores que reducen el “drift” entre máquinas. Este ecosistema facilita que dos grupos distintos puedan reproducir un resultado con un esfuerzo razonable y detectar de inmediato si un cambio de versión en el sistema operativo, una librería o un controlador altera los comportamientos observados. Esa misma disciplina sostiene prácticas básicas de ciberseguridad y seguridad operacional: cambio de credenciales por defecto, aislamiento de la red del robot respecto a otras redes del centro, actualización responsable de paquetes y gestión de vulnerabilidades de forma planificada, control de acceso físico al hardware y verificación de orígenes para dependencias críticas, reduciendo el riesgo de incidentes de cadena de suministro. No se trata de securizar una infraestructura crítica, sino de enseñar a pensar en superficies de ataque, mínimos privilegios, “fail-safe” y recuperación: virtudes que escalan desde el aula a entornos productivos. Esta manera de trabajar no solo protege la actividad docente; también refuerza la confianza de la institución en el uso de tecnologías autónomas al demostrar que el proyecto se rige por normas claras, auditablemente cumplidas y comunicadas en la memoria técnica con evidencias y referencias cruzadas a capítulos y anexos.

La sostenibilidad y la responsabilidad ambiental enriquecen la reflexión ética al extenderla en el tiempo y sobre el entorno material. El proyecto minimiza consumos y residuos mediante la reparación y el reúso preferentes frente al reemplazo, el mantenimiento preventivo periódico y la entrega de baterías y electrónica como RAEE en puntos autorizados. Cuando se diseñan carcasas o protecciones, se priorizan polímeros reciclables y geometrías que faciliten el desmontaje, y se documentan decisiones para que cohortes futuras puedan iterarlas sin reinventar. El consumo energético se racionaliza ajustando servicios activos a lo que realmente aporta al aprendizaje y planificando prácticas que eviten picos innecesarios. Esta sobriedad no solo reduce la huella ambiental: educa en una noción coherente de coste total, donde cada ciclo de batería preservado y cada componente recuperado equivalen a tiempo, presupuesto y recursos disponibles para experimentar mejor. En el plano de la propiedad intelectual y del software abierto, el proyecto cumple licencias y términos de uso, cita correctamente, evita incorporar material sin permisos y explica la razón de ser de las licencias elegidas al publicar artefactos derivados; cuando se modifica una herramienta, se documentan cambios y se ofrece, si procede, la devolución de mejoras a la comunidad. Trabajar con tecnologías abiertas no es únicamente aprovechar su disponibilidad: es entrenarse en una ética de colaboración, transparencia y responsabilidad compartida que la ingeniería contemporánea necesita para sostener su propio ecosistema. Este modo de hacer, sistematizado en los materiales de curso y en el repositorio de prácticas, prepara al estudiantado para contribuir con solvencia y respeto a proyectos reales que se construyen a muchas manos.

Este cierre no pretende clausurar la discusión ética, sino mantenerla viva: cada práctica deja rastros técnicos y argumentos que la siguiente iteración podrá pulir; cada cohorte hereda un estándar y, a la vez, la libertad responsable de elevarlo. Lo decisivo es que el proyecto demuestra, en la forma misma de su trabajo, que la excelencia técnica y el buen juicio no compiten: se exigen mutuamente. Si el alumnado aprende a medir con precisión y, simultáneamente, a preguntarse por el sentido y el efecto de lo que mide; si aprende a versionar el código y, a la vez, a versionar sus propios criterios a la luz de datos y razones, entonces el objetivo de este anexo se habrá cumplido. Habremos formado no solo operadores de sistemas autónomos, sino profesionales capaces de integrar capacidades y valores, de sostener su práctica ante sus pares y de responder ante la sociedad por lo que ponen en marcha. Esa madurez, documentada en capítulos y anexos de la memoria y verificada en la reproducibilidad de los resultados, es la mejor garantía de que la robótica con IA en el aula puede ser, de verdad, una escuela de responsabilidad.

ANEXO IV: CONTENIDO AUDIOVISUAL

En el siguiente anexo se recogen todos los enlaces a vídeos de la práctica de las funciones básicas y avanzadas que se han llevado a cabo con el ROSMASTER A1. Los vídeos están subidos y compartidos en Google Drive, para obtener una visualización de buena calidad se recomienda descargarlos en el propio ordenador o dispositivo personal, de lo contrario, si se ven tras clicar el enlace y sin descargar, se obtendrá una visualización preliminar de baja calidad de imagen. Todo el contenido audiovisual del proyecto se encuentra en la Tabla A4-1.

Función	Apartado en el que se desarrolla	Enlace
Movimiento RC	3.4.3	https://drive.google.com/file/d/1Skpl5MfMDpr_P4edYsLEg9j-geMOYOtx/view?usp=sharing
Activación de la Cámara	3.4.4	https://drive.google.com/file/d/1M2KjMdMZMQ6Rc-VByv6GevojmhmMH3Kn/view?usp=sharing
Activación del LiDAR	3.4.5	https://drive.google.com/file/d/1NkmOp_FdPD8Jck0vTBxZ7L6dUMCwlmvN/view?usp=sharing
Activación del Altavoz	3.4.6	https://drive.google.com/file/d/12AL-201BQtGcn2yggasgapm-47yqkMVx/view?usp=sharing
Mapeado de una superficie	3.5.1	https://drive.google.com/file/d/10aITUyPOdSvTC9P4t6VhKHxJk6RQmUYS/view?usp=sharing
Evitación de Obstáculos	3.5.2	https://drive.google.com/file/d/12CKqV_JXcBObOU7qaEtgTKcnS5g_UvWB/view?usp=sharing
SLAM	3.5.2	https://drive.google.com/file/d/1mIdufACGnsGNILNBZfsZ7p673RqKRJHB/view?usp=sharing
Seguimiento por Gestos	3.5.3	https://drive.google.com/file/d/1-eeIIVaoRNIJTFiHrjpkUNepF19pfne0/view?usp=sharing
Seguimiento de Personas	3.5.3	https://drive.google.com/file/d/1tVtKYPZsELSHHzZnN9b4t3EIosfA43mq/view?usp=sharing
Identificación de Objetos	3.5.3	https://drive.google.com/file/d/1JSOEXKft1EWCCv68cjlFJkBK-pCVbnny/view?usp=sharing
Seguimiento por colores	3.5.3	https://drive.google.com/file/d/1vdlSAmAJvA78CloxInN3O2nSVMhE02Ur/view?usp=sharing
Seguimiento de Línea	3.5.4	https://drive.google.com/file/d/1TO6-ApYqI8TGANln-25QZv4nbuV1091/view?usp=sharing
Enjambre RC	3.5.5	https://drive.google.com/file/d/1jP1zjpRX9n2GxUd1dsmcdSljbRO8JS8U/view?usp=sharing

Tabla A4-1 Contenido audiovisual del proyecto