

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Mecatrónica

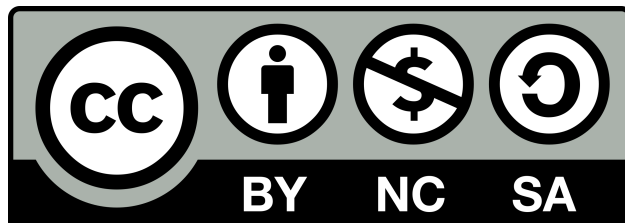


Rediseño de enjambre robótico Atta-Bot

Informe de Proyecto de Graduación para optar por el título de
Ingeniero en Mecatrónica con el grado académico de Licenciatura

Francisco Aguilera Quesada

Cartago, Costa Rica, 19 de Noviembre de 2024



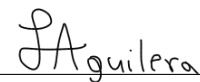
Rediseño de enjambre robótico Atta-Bot © 2024 por Francisco Aguilera Quesada está licenciado bajo una Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, Costa Rica, 19 de Noviembre de 2024



Francisco Aguilera Quesada

Céd: 1-1749-0209

INSTITUTO TECNOLÓGICO DE COSTA RICA
PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA
PROYECTO FINAL DE GRADUACIÓN
ACTA DE APROBACIÓN

El profesor asesor del presente trabajo final de graduación, indica que el documento presentado por el estudiante cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica del Instituto Tecnológico de Costa Rica para ser defendido ante el jurado evaluador, como requisito final para aprobar el curso Proyecto Final de Graduación y optar así por el título de Ingeniero(a) en Mecatrónica, con el grado académico de Licenciatura.

Estudiante: Francisco Jose Aguilera Quesada

Proyecto: Rediseño de enjambre robótico Atta-Bot



MSc. Ing. Juan Carlos Brenes Torres

Asesor

Cartago, 19 de noviembre de 2024

INSTITUTO TECNOLÓGICO DE COSTA RICA
PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA
PROYECTO FINAL DE GRADUACIÓN
ACTA DE APROBACIÓN

Proyecto final de graduación defendido ante el presente jurado evaluador como requisito para optar por el título de Ingeniero(a) en Mecatrónica con el grado académico de Licenciatura, según lo establecido por el programa de Licenciatura en Ingeniería Mecatrónica, del Instituto Tecnológico de Costa Rica.

Estudiante: Francisco Jose Aguilera Quesada

Proyecto: Rediseño de enjambre robótico Atta-Bot

Miembros del jurado evaluador

JUAN LUIS CRESPO MARIÑO (FIRMA)
PERSONA FISICA, CPF-08-0113-0166.
Fecha declarada: 19/11/2024 03:08:06 PM
Esta es una representación gráfica únicamente,
verifique la validez de la firma.

Dr. Ing. Juan Luis Crespo Mariño

Jurado



Firmado digitalmente por
RONALD JOSE LOAIZA BALDARES
(FIRMA)
Fecha: 2024.11.19 15:41:52 -06'00'

Ing. Ronald Loaiza Baldares

Jurado

Los miembros de este jurado dan fe de que el presente proyecto final de graduación ha sido aprobado y cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica.

Cartago, 19 de noviembre de 2024

Resumen

Este proyecto se centra en el rediseño del enjambre robótico Atta-Bot, una plataforma desarrollada para realizar tareas autónomas de exploración y mapeo en entornos complejos. La primera versión de los Atta-Bots presentaba limitaciones en maniobrabilidad, comunicación y tamaño, dificultando su eficacia y capacidad de expansión. El objetivo principal del proyecto es mejorar estos robots para crear un sistema de enjambre más preciso y cómodo de usar. Para ello, se simplificó su arquitectura de hardware mediante la integración de un único microprocesador, se optimizó el sistema de comunicación, entre otras cosas. Asimismo, se introdujo un nodo base que facilita la coordinación y comunicación dentro del enjambre, permitiendo una mayor eficiencia en sus operaciones.

El desarrollo inició con un diagnóstico exhaustivo de la versión original de los Atta-Bots para identificar sus limitaciones. Con esta información, se empleó la metodología de Ulrich y Eppinger para seleccionar la solución óptima, definir las necesidades del cliente y establecer métricas de diseño para esta nueva versión. Finalmente, se construyó un prototipo funcional del sistema de enjambre, que fue sometido a validaciones para verificar que cumpliera con las expectativas y requisitos definidos, asegurando así su capacidad de satisfacer las demandas tanto técnicas como investigativas del laboratorio.

Palabras clave: robots enjambre, Atta-Bot, exploración aleatoria, comunicación UDP

Abstract

This project focuses on the redesign of the Atta-Bot robotic swarm, a platform developed to perform autonomous exploration and mapping tasks in complex environments. The first version of the Atta-Bots had limitations in maneuverability, communication, and size, hindering its efficiency and expandability. The main goal of the project is to improve these robots to create a swarm system that is more accurate and comfortable to use. To achieve this, their hardware architecture was simplified by integrating a single microprocessor, and the communication system was optimized, among other improvements. Additionally, a base node was introduced to facilitate coordination and communication within the swarm, allowing for greater operational efficiency.

The development began with a thorough diagnosis of the original Atta-Bots version to identify its limitations. With this information, Ulrich and Eppinger's methodology was employed to select the optimal solution, define customer needs, and establish design metrics for this new version. Finally, a functional prototype of the swarm system was built and subjected to validation tests to ensure it met the defined expectations and requirements, thus ensuring its ability to meet both the technical and research demands of the laboratory.

Keywords: swarm robots, Atta-Bot, random walk, UDP communication

A mi madre, que siempre creyó en mí y cuya memoria me inspira cada día.

Agradecimientos

A Dios, por brindarme las fuerzas y las oportunidades para lograr esta meta.

A mis padres, quienes me formaron con los valores que hoy me definen. A mi mamá, cuya memoria me acompaña, y a mi papá, que ha sido un apoyo fundamental en cada paso.

A mis hermanas y a mi sobrina, por su apoyo constante a lo largo de todos estos años y a toda mi familia, por estar presente y brindar su ayuda siempre que la he necesitado.

A mis amigos, por su compañía y motivación, y a todas las personas que conocí en este camino, quienes contribuyeron de distintas maneras para hacer realidad este logro.

Francisco Aguilera Quesada

Cartago, Costa Rica, 19 de Noviembre de 2024

Índice general

Índice de figuras	iii
Índice de tablas	v
Lista de símbolos y abreviaciones	vii
1 Introducción	1
1.1 Contexto	1
1.2 Descripción del problema	2
1.3 Síntesis del problema	2
1.4 Objetivos	3
1.4.1 Objetivo general	3
1.4.2 Objetivos específicos	3
1.5 Estructura del documento	4
2 Marco Teórico	5
2.1 Robótica de enjambres	5
2.1.1 Principios de la robótica de enjambre	5
2.1.2 Sistemas actuales de robótica de enjambre	6
2.2 Protocolos de comunicación inalámbricos	6
2.2.1 WiFi	7
2.2.2 Bluetooth	7
2.2.3 ESP-NOW	8
2.3 Sensores de Detección	8
2.3.1 Sensores Infrarrojos	8
2.3.2 Sensores de Tiempo de Vuelo	9
2.3.3 Sensores Ultrasónicos	10
3 Metodología	11
3.1 Etapa 0: Diagnóstico del Atta-Bot	11
3.2 Etapa 1: Identificar necesidades del cliente	15
3.2.1 Recopilación de información e interpretación de necesidades	15
3.2.2 Categorización de las necesidades interpretadas	16
3.2.3 Jerarquización de necesidades	17
3.3 Etapa 2: Establecer especificaciones objetivo	18

3.4	Etapa 3: Generación de conceptos	20
3.4.1	Descomposición funcional del sistema	20
3.4.2	Búsqueda interna y externa de conceptos	22
3.4.3	Elaboración de conceptos	25
3.5	Etapa 4: Selección de conceptos	25
3.5.1	Filtrado de conceptos	26
3.5.2	Selección del concepto ganador	27
4	Desarrollo del Concepto Ganador	29
4.1	Consideraciones de diseño	29
4.2	Distribución de componentes en el robot	32
4.3	Detección de obstáculos	34
4.4	Clasificación de obstáculos	37
4.5	Control de movimiento	38
4.6	Gestión de información y comunicación del robot	41
4.6.1	Máquina de estados finitos	41
4.6.2	Comunicación y Lectura de Sensores	45
4.7	Diseño de la base	48
4.7.1	Obtención de posiciones de los robots	48
4.7.2	Funcionamiento de la base	49
5	Resultados y Análisis	51
5.1	Validación de la exactitud del robot	51
5.1.1	Exactitud en movimientos lineales	51
5.1.2	Exactitud en giros	54
5.2	Validación de la duración de la batería del robot	57
5.3	Validación del sistema de comunicación	58
5.3.1	Comunicación base a robots	58
5.3.2	Comunicación entre robots	59
5.4	Validación de la satisfacción del usuario	61
5.5	Validación del tiempo de procesado de un fotograma	62
5.6	Validación de la identificación y clasificación de obstáculos	64
5.7	Validación de los comportamientos de enjambre	65
5.8	Análisis económico	68
6	Conclusiones y Recomendaciones	71
6.1	Conclusiones	71
6.2	Recomendaciones	72
	Bibliografía	73
A	Gráficas de funcionamiento del control PID	76
B	Código de los Robots y la Base	79

Índice de figuras

1.1	Síntesis del problema a resolver.	3
2.1	Principio de funcionamiento de los sensores infrarrojos. Fuente [1]	9
2.2	Principio de funcionamiento de los sensores ToF. Fuente [2]	9
2.3	a) Configuración de pines para el HC-SR04. b) Diagrama de tiempos y forma de onda. Fuente [3]	10
3.1	Etapas de la metodología de diseño a desarrollar. Fuente [4]	11
3.2	Imágenes del Atta-Bot. Fuente [5]	12
3.3	Descomposición funcional del problema a resolver.	21
4.1	Diagrama electrónico del Atta-Bot STEM. Fuente [6]	30
4.2	Vistas del nuevo Atta-Bot.	33
4.3	Diagrama eléctrico del nuevo Atta-Bot.	34
4.4	Distribución de sensores para detectar obstáculos.	35
4.5	Sensor HW-488. Fuente [7]	35
4.6	Implementación gráfica de la medida 1.	36
4.7	Implementación gráfica de algoritmo de clasificación.	38
4.8	Implementación del control PID por motor.	39
4.9	Gráfica del control PID en funcionamiento para desplazamiento de 300mm.	40
4.10	FSM implementada para el control del robot.	42
4.11	Diagrama de flujo	45
4.12	Identificador utilizado por los robots en diferentes colores. Fuente [8]	48
4.13	Diagrama de flujo del sistema de visión. Fuente [8]	49
4.14	Diagrama de flujo de la base.	50
5.1	Imagen del proceso de validación de la exactitud en movimientos lineales.	52
5.2	Diagrama de cajas y bigotes para el error de prueba de desplazamiento.	52
5.3	Imagen del proceso de validación de la exactitud en giros.	55
5.4	Diagrama de cajas y bigotes para el error de prueba de giro.	55
5.5	Campo de pruebas utilizado para pruebas de validación.	57
5.6	Criterios y variables nominales de usabilidad. Fuente [9].	61
5.7	Diagrama de cajas y bigotes para las tres pruebas realizadas de tiempo de procesado de un fotograma.	63
5.8	Resultados obtenidos en la segunda prueba del comportamiento de enjambre.	66
5.9	Máscara utilizada para medir de forma correcta el área explorada por los robots.	67

A.1	Gráfica del control PID en funcionamiento para desplazamiento de 150mm. .	76
A.2	Gráfica del control PID en funcionamiento para desplazamiento de 200mm. .	77
A.3	Gráfica del control PID en funcionamiento para desplazamiento de 250mm. .	77
A.4	Gráfica del control PID en funcionamiento para desplazamiento de 400mm. .	78
A.5	Gráfica del control PID en funcionamiento para desplazamiento de 500mm. .	78

Índice de tablas

2.1	Robots para uso en robótica de enjambre.	6
3.1	Características de los Robots Atta-Bot.	12
3.2	Errores obtenidos para el sistema de control de navegación de la primera versión del Atta-Bot.	13
3.3	Tasa de éxito en la recepción paquetes para el sistema de comunicación de la primera versión del Atta-Bot.	14
3.4	Características de la primera versión del Atta-bot.	14
3.5	Necesidades determinadas tras el proceso de entrevistas.	16
3.6	Categorización de las necesidades interpretadas.	16
3.7	Necesidades finales y su importancia en la solución.	17
3.8	Métricas y valores ideales e marginales.	18
3.9	Matriz de correlación entre las métricas y las necesidades.	20
3.10	Soluciones de subproblemas para la formulación de conceptos.	25
3.11	Posibles soluciones candidatas al problema a resolver.	25
3.12	Matriz de filtrado de conceptos.	26
3.13	Resultados de la combinatoria de B y D.	27
3.14	Evaluación de conceptos.	28
4.1	Componentes utilizados en la primera versión del Atta-Bot STEM.	30
4.2	Parámetros del control PID.	41
5.1	Resultados de las pruebas de desplazamiento.	53
5.2	Resultados de las pruebas de giro.	56
5.3	Resultados de la duración de la batería.	58
5.4	Resultados obtenidos en la prueba de comunicación base a robots.	59
5.5	Resultados obtenidos en la prueba de comunicación entre robots.	60
5.6	Cuestionario utilizado para medir satisfacción del usuario.	61
5.7	Resultados de la validación de la satisfacción del usuario.	62
5.8	Clasificación de la detección y clasificación de obstáculos.	64
5.9	Resultados de la primera prueba de detección de obstáculos.	65
5.10	Resultados de la segunda prueba de detección de obstáculos.	65
5.11	Resultados del comportamiento de exploración aleatoria.	67
5.12	Coste unitario de la nueva versión del robot Atta-Bot.	69
5.13	Costos de los componentes de la base.	69

5.14 Análisis económico del proyecto.	69
---	----

Lista de símbolos y abreviaciones

Abreviaciones

BLE	Bluetooth Low Energy
FSM	Finite State Machine
GIRoM	Grupo de Investigación en Robótica Múltiples
IMU	Inertial Measurement Unit
IR	Infrarrolo
ITCR	Instituto Tecnológico de Costa Rica
PID	Proporcional, Integral y Derivativo
PROE	Planificación de Rutas Óptimas mediante Enjambres
RSSI	Received Signal Strength Indicator
SD	Sistema diseñado
TCP/IP	Transmission Control Protocol/Internet Protocol
TDMA	Acceso Múltiple por División de Tiempo
ToF	Time of Flight
UDP	User Datagram Protocol

Notación general []

(x, y, α) Componentes de la pose de un robot

Capítulo 1

Introducción

1.1 Contexto

El proyecto Atta-Bot pertenece al grupo GIRoM del ITCR. Este proyecto comenzó en 2022 como una colaboración interdisciplinaria entre la Escuela de Matemática y la Escuela de Ingeniería Mecatrónica. Su objetivo principal es mejorar las funcionalidades y capacidades de los Atta-Bots, una serie de robots terrestres desarrollados como parte del proyecto PROE del mismo grupo. Estos robots están diseñados para la exploración y mapeo de entornos desconocidos, así como para la determinación de rutas óptimas en diversas circunstancias. Los investigadores principales incluyen a Cindy Calderón Arce, directora del proyecto, Rebeca Solís Ortega y Juan Carlos Brenes Torres.

Actualmente, derivado de este proyecto, se ha desarrollado una nueva iniciativa denominada Atta-Bot STEM [6], enfocada en la creación de una plataforma de robótica educativa. Con base en la arquitectura de hardware desarrollada en Atta-Bot STEM, esta será reutilizada y adaptada para el desarrollo de este nuevo proyecto.

1.2 Descripción del problema

El proyecto Atta-Bot, aunque exitoso en su fase inicial, enfrenta limitaciones significativas debido a su hardware actual. Los robots Atta-Bots utilizan dos microprocesadores, lo que complica su programación y limita la capacidad de expansión para nuevas funcionalidades. Además, el diseño físico de estos robots es grande y voluminoso, lo que restringe su maniobrabilidad y eficiencia en entornos complejos, contribuyendo a la aparición de puntos muertos y choques ocasionales entre ellos. Estos problemas se ven agravados por la falta de comunicación entre los robots.

Además, aunque se dispone de un sistema de visión para determinar las posiciones de los robots, este no está integrado en el sistema de enjambre, ya que se utiliza de manera diferida a las pruebas, lo que significa que no se cuenta con un nodo base que sirva de apoyo para el sistema de enjambre. Estos hallazgos subrayan la necesidad de actualizar el hardware para mejorar el rendimiento y la capacidad de los Atta-Bots, así como la integración de un nodo base para facilitar la comunicación con los robots y coordinar sus acciones.

El nuevo proyecto tiene como objetivo reutilizar el hardware del proyecto Atta-Bots STEM [6], mediante el diseño de un sistema que incorpore un único microprocesador más potente y un formato más compacto. Esta estrategia simplificará el desarrollo y la programación de los robots, facilitando la implementación de nuevas funcionalidades y mejorando su maniobrabilidad. La modernización del hardware no solo permitirá una mayor capacidad de procesamiento, sino que también facilitará la expansión para futuras mejoras. Adicionalmente, se propone la creación de un nodo base, aprovechando el trabajo previo realizado en [8], lo que fortalecerá la coordinación y comunicación entre los robots en el sistema de enjambre.

Este avance es crucial para el proyecto Atta-Bot, ya que permitirá seguir innovando en el campo de los robots de enjambre. La implementación de un nodo base, será fundamental para optimizar el rendimiento del sistema. Esta capacidad de actualizar y expandir fácilmente los robots asegurará que el proyecto Atta-Bot se mantenga a la vanguardia de la investigación y el desarrollo en robótica. Además, el nodo base facilitará la integración de nuevas funcionalidades y mejorará la capacidad de los robots para operar en entornos complejos y dinámicos, fomentando así la creación de soluciones avanzadas y eficientes para la exploración y la operación.

1.3 Síntesis del problema

El desafío principal consiste en adaptar el hardware de los robots educativos del proyecto Atta-Bot STEM para transformarlos en un sistema de robots de enjambre que cumpla con los requisitos del proyecto original Atta-Bot. Esta transformación es fundamental para implementar soluciones innovadoras y eficientes en la exploración y operación de entornos complejos. Para lograrlo, es necesaria la incorporación de nuevos sensores y el desarrollo de un protocolo de comunicación que permita la coordinación efectiva entre los robots, así

como la integración de un nodo base que centralice la información y optimice la interacción dentro del sistema. En la figura 1.1, se presenta de manera gráfica la problemática planteada y cómo se relacionan los diferentes proyectos del grupo GIROM.

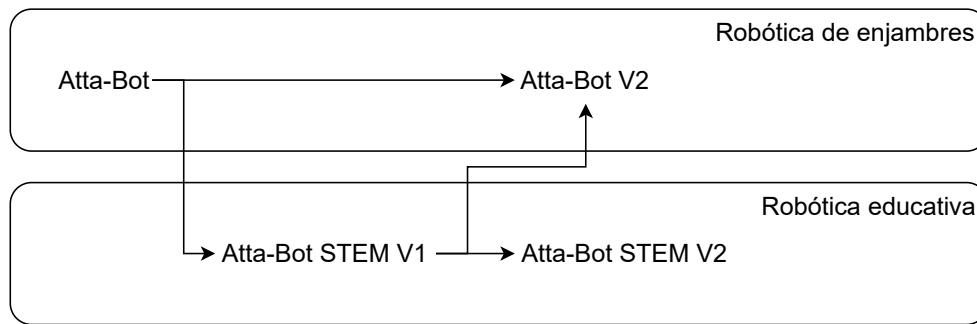


Figura 1.1: Síntesis del problema a resolver.

1.4 Objetivos

Los objetivos para este proyecto se encuentran definidos a continuación.

1.4.1 Objetivo general

Adaptar las capacidades de los robots Atta-Bots mediante tecnologías de localización, detección de obstáculos y desplazamientos con el fin de desarrollar un sistema de enjambre preciso y competente.

1.4.2 Objetivos específicos

- Evaluar la condición actual del robot Atta-Bot para determinar sus capacidades y limitaciones actuales.
- Diseñar una arquitectura de comunicación para que los robots se comuniquen entre sí y con la base.
- Evaluar los comportamientos y realizar modificaciones eléctricas como mecánicas que permitan al robot esquivar obstáculos de manera autónoma y reconocer si el obstáculo es otro robot.
- Validar la implementación conjunta de los sistemas creados mediante el desarrollo de un comportamiento de enjambre.

1.5 Estructura del documento

A continuación, se presenta un esquema de las secciones que conforman este documento, junto con una breve descripción de su contenido.

- Capítulo 1: Se introduce el contexto del proyecto, definiendo el problema a resolver y los objetivos planteados.
- Capítulo 2: Se presenta el marco teórico, proporcionando los conceptos fundamentales que facilitarán la comprensión del diseño y su análisis posterior.
- Capítulo 3: Se expone la metodología de diseño empleada, justificando cada decisión tomada en el proceso para seleccionar la mejor solución posible.
- Capítulo 4: Se detallan las consideraciones previas al diseño de la solución, junto con los la implementación del concepto ganador.
- Capítulo 5: Se presentan y analizan los resultados del concepto ganador, junto con un análisis económico de los recursos empleados en el proyecto.
- Capítulo 6: Se presentan las conclusiones y recomendaciones finales derivadas del desarrollo del proyecto.

Capítulo 2

Marco Teórico

Previo al desarrollo del proyecto, es importante primero introducir una serie de conceptos, con la finalidad de brindar mayor claridad al lector a la hora de abordar el texto. Dichos conceptos se desarrollan a continuación:

2.1 Robótica de enjambres

La robótica de enjambre es una rama de la robótica que se inspira en los comportamientos colectivos de la naturaleza, particularmente en insectos sociales como las hormigas, termitas y las abejas. Su objetivo es coordinar un gran número de robots sencillos para que, a través de la interacción y cooperación local, puedan llevar a cabo tareas complejas de manera eficiente [10]. A diferencia de los sistemas centralizados, los robots en un enjambre actúan de forma autónoma y distribuida, lo que les confiere ventajas clave como la robustez ante fallos individuales, flexibilidad ante cambios y escalabilidad en distintos entornos. [11]

2.1.1 Principios de la robótica de enjambre

La robótica de enjambre se basa en una serie de principios fundamentales que guían el comportamiento y las interacciones entre los robots. Estos principios aseguran que los robots puedan colaborar de manera eficiente y alcanzar objetivos en común. Entre los principios clave se encuentran [10]-[12]:

- **Descentralización:** No existe un robot central que organice el comportamiento de los demás. Cada robot toma decisiones autónomamente, basadas en la información de su entorno y de otros robots cercanos.
- **Simplicidad Individual, Complejidad Colectiva:** Aunque los robots, de manera individual, poseen capacidades limitadas, logran llevar a cabo tareas complejas a través de interacciones entre ellos. Es a partir de esta colaboración que surge un comportamien-

to emergente, permitiendo al sistema realizar acciones más complejas y eficaces de lo que sería posible para cada robot por sí solo.

- Escalabilidad: El sistema puede adaptarse a diferentes tamaños de grupos de robots. A medida que aumenta el número de robots, la eficiencia del sistema no se ve afectada negativamente y en algunos casos, hasta puede mejorar, sin necesidad de modificar la forma en que los robots interactúan.
- Tolerancia a Fallos: La pérdida o el fallo de un robot no compromete el funcionamiento global del sistema. Otros robots pueden asumir las funciones del robot fallido, garantizando que el sistema siga operando de manera eficiente.

2.1.2 Sistemas actuales de robótica de enjambre

En los últimos años, la robótica de enjambre ha experimentado avances significativos, impulsados por el desarrollo de proyectos en diversas empresas y universidades de todo el mundo. Cada uno de estos proyectos presenta características únicas, adaptadas a la realización de comportamientos colectivos específicos. A continuación, se muestran algunos ejemplos destacados de robots de enjambre, con sus respectivas características. La Tabla 2.1 resume estos sistemas, proporcionando una visión general de este tipo de sistemas [13], [14].

Tabla 2.1: Robots para uso en robótica de enjambre.

Robot	Fecha de salida	Sensores de detección	Comunicación	Tamaño	Disponible comercialmente	Costo
E-puck	2004	8 sensores de proximidad IR 3 micrófonos Cámara	Bluetooth	7,5 cm	✓	USD 1000
Khepera IV	2015	8 sensores de proximidad IR 5 sensores ultrasónicos Micrófono Cámara	Bluetooth WiFi	14 cm	✓	USD 3200
Kilobot	2010	Sensores de proximidad IR	IR corto alcance	1,6 cm	✓	USD 130
Colias	2014	Sensores de proximidad IR	IR corto alcance	4 cm	X	USD 32

2.2 Protocolos de comunicación inalámbricos

Como se ha mencionado previamente, la comunicación desempeña un papel crucial en la robótica móvil, ya que es esencial para la coordinación eficiente de movimientos y acciones colectivas. Por esta razón, a continuación se presentan y describen varios protocolos de comunicación inalámbrica comúnmente empleados en este campo para la transferencia de información.

2.2.1 WiFi

WiFi es un estándar de comunicación inalámbrica definido por la serie IEEE 802.11, ampliamente empleado para interconectar dispositivos a través de redes de área local (LAN) y redes de área amplia (WAN), con la capacidad de proporcionar acceso a internet. Este estándar ha evolucionado significativamente desde su creación, y ahora incluye varios subestándares, como 802.11a/b/g/n/ac, cada uno con características y mejoras específicas que optimizan el rendimiento en diferentes escenarios de aplicación [15]. En robótica de enjambre WiFi es valorado por varias razones técnicas que lo hacen adecuado para aplicaciones complejas, entre las cuales:

- **Compatibilidad y disponibilidad:** Los protocolos 802.11b/g/n son ampliamente compatibles con la mayoría de los dispositivos WiFi existentes, esto debido al tiempo que llevan en el mercado, lo que facilita la integración de los robots a redes locales sin necesidad de equipos especializados [15]. Esta accesibilidad es crucial para proyectos que requieren comunicación entre múltiples robots y servidores centralizados.
- **Alcance:** En particular los protocolos, 802.11b y 802.11g, que operan en la banda de 2,4 GHz, ofrecen un buen equilibrio entre velocidad y alcance [15]. Esto permite que los robots de enjambre se comuniquen de manera eficiente incluso a distancias relativamente largas dentro de entornos de trabajo amplios.
- **Escalabilidad:** Los protocolos 802.11b/g/n permiten conectar varios robots en una misma red, aunque el número de dispositivos que pueden operar de manera eficiente en una sola red está limitado por la capacidad de la red local y la posible congestión [15].

2.2.2 Bluetooth

Bluetooth es un estándar de comunicación inalámbrica de corto alcance, diseñado principalmente para conectar dispositivos cercanos. Aunque su rango es limitado comparado con WiFi, Bluetooth destaca por su alta eficiencia energética, lo que lo convierte en una opción ideal para dispositivos que requieren un bajo consumo de energía [16]. Además, permite tanto conexiones punto a punto como multi-punto, lo que lo hace especialmente adecuado para enjambres pequeños de robots, donde se necesita una coordinación directa entre los elementos sin un alto consumo energético.

Dentro del estándar Bluetooth, existe una variante llamada Bluetooth Low Energy (BLE), diseñada específicamente para reducir significativamente el consumo de energía mientras mantiene una comunicación eficiente [16]. En el contexto de la robótica de enjambre, BLE se puede utilizar para la comunicación entre robots cercanos, facilitando la detección de proximidad y la sincronización de acciones con un consumo energético muy bajo [17] [18].

2.2.3 ESP-NOW

ESP-NOW es un protocolo de comunicación desarrollado por Espressif, la empresa detrás de los microcontroladores ESP32 o ESP8266. Este protocolo permite la transmisión de datos directa entre dispositivos ESP únicamente sin necesidad de una red WiFi o emparejamiento como en Bluetooth, lo que la hace capaz de enviar mensajes rápidos y su bajo consumo energético lo convierten en una excelente opción para la comunicación en enjambres robóticos [19]. ESP-NOW permite la creación de redes mesh de bajo costo y alta eficiencia, lo que facilita la coordinación entre múltiples robots en tiempo real.

2.3 Sensores de Detección

En el campo de la robótica de enjambre, la capacidad de detectar obstáculos y el entorno es de vital importancia para garantizar la seguridad y la eficiencia en la navegación. Los robots usualmente como se puede observar en la tabla 2.1 suelen emplear una variedad de sensores que les permiten obtener información sobre su entorno y reaccionar en consecuencia. A continuación, se describen los tres tipos de sensores más comúnmente utilizados en robótica para la detección de obstáculos.

2.3.1 Sensores Infrarrojos

Los sensores infrarrojos o mejor conocidos como IR son dispositivos ampliamente utilizados en el ámbito de la robótica para la detección de objetos cercanos y la medición de distancias a corto alcance. Estos sensores operan mediante un sistema que consiste en un emisor de luz infrarroja y un receptor que capta la luz reflejada por los objetos en su proximidad. Al emitir un haz de luz infrarroja, el sensor puede detectar la presencia de un objeto. Además, al analizar la cantidad de luz reflejada, es posible estimar la distancia entre el sensor y el objeto, ya que la intensidad de la reflexión varía según la proximidad [1] [3]. En la figura [1] se puede observar el principio de funcionamiento de este tipo de sensores.

Entre las principales ventajas de los sensores IR se encuentran su bajo coste y su eficiencia energética, lo que los convierte en una opción atractiva para aplicaciones en robótica de bajo coste y donde el consumo de energía son consideraciones críticas. Sin embargo, presentan algunas desventajas. Su rendimiento puede verse afectado por superficies transparentes o altamente reflectantes, lo que limita su rango de detección y precisión [1]. Además, los sensores infrarrojos son susceptibles a interferencias de otros dispositivos IR, lo que puede causar lecturas erróneas y afectar la fiabilidad del sistema de detección.

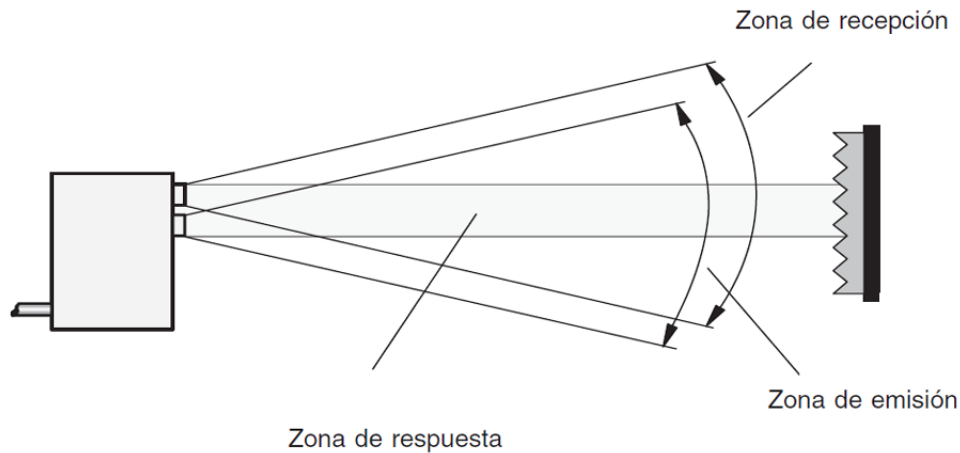


Figura 2.1: Principio de funcionamiento de los sensores infrarrojos. Fuente [1]

2.3.2 Sensores de Tiempo de Vuelo

Los sensores de Tiempo de Vuelo o ToF son sensores que miden la distancia a un objeto calculando el tiempo que tarda un pulso de luz, generalmente en el espectro infrarrojo o láser, en viajar desde el sensor hasta el objeto y regresar. Su principio de funcionamiento se basa en emitir un haz de luz y registrar el tiempo de retorno tras reflejarse en el objeto, un dato que es directamente proporcional a la distancia [2]. Esto permite obtener mediciones precisas en un amplio rango. En la figura 2.2, se puede observar el principio de funcionamiento de este tipo de sensores.

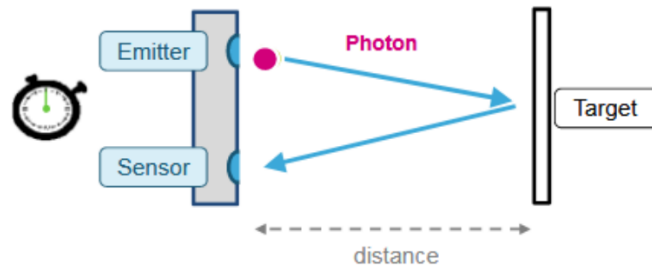


Figura 2.2: Principio de funcionamiento de los sensores ToF. Fuente [2]

A diferencia de los sensores infrarrojos convencionales, los sensores ToF ofrecen mayor precisión y un rango de medición superior, llegando a detectar objetos a varios metros de distancia. Además, su alta velocidad de respuesta y capacidad para funcionar en diversas condiciones de luz los hace ideales para aplicaciones que requieren precisión y rapidez [2]. Sin embargo, estos sensores suelen ser más costosos que los IR convencionales y pueden verse afectados por interferencias en entornos con abundante luz ambiental.

2.3.3 Sensores Ultrasónicos

Los sensores ultrasónicos operan emitiendo ondas de sonido de alta frecuencia, midiendo el tiempo que estas tardan en reflejarse en un objeto cercano para calcular la distancia. El principio de funcionamiento consiste en emitir un pulso ultrasónico que viaja a través del aire, se refleja en un objeto, y es recibido nuevamente por el sensor. Al medir el tiempo transcurrido entre la emisión y la recepción del eco, se puede determinar con precisión la distancia al objeto [1] [3]. En la figura 2.3, se muestra un sensor ultrasónico típico, así como una ilustración del principio de funcionamiento en acción.

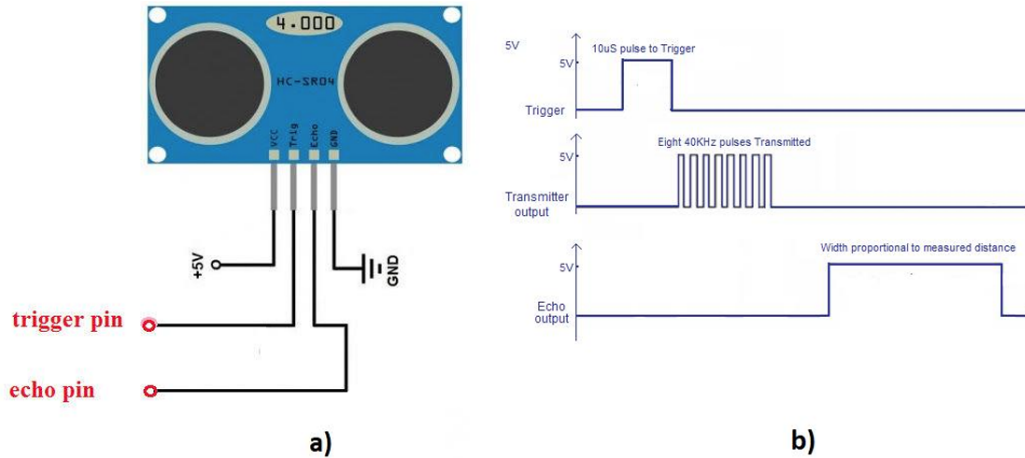


Figura 2.3: a) Configuración de pines para el HC-SR04. b) Diagrama de tiempos y forma de onda. Fuente [3]

Una de las principales ventajas de estos sensores es su alta precisión en la detección de objetos sólidos, incluso en condiciones de baja visibilidad, ya que no se ven afectados por la luz ambiental ni por la transparencia de los objetos. Además, suelen ofrecer un rango de detección más amplio que los sensores infrarrojos, con capacidad para medir distancias de varios metros. Sin embargo, presentan dificultades para detectar objetos con superficies irregulares o muy pequeñas y el ruido ambiental puede interferir en sus mediciones, reduciendo su efectividad en entornos ruidosos [3].

Capítulo 3

Metodología

Para el desarrollo de este proyecto, se ha adoptado como guía la metodología de diseño propuesta por Ulrich y Eppinger [4]. Esta metodología ha sido seleccionada debido a su capacidad para estructurar de manera eficiente el proceso de diseño a lo largo del proyecto. En este capítulo, se describirá detalladamente cómo se aplicará dicha metodología en el contexto del diseño de ingeniería.

La metodología consta de un total de siete etapas, todas de carácter iterativo, lo que permite retroceder y ajustar cada fase según sea necesario. El proceso abarca desde la identificación de las necesidades del cliente hasta el desarrollo de una solución completa para la problemática planteada. En el contexto de este proyecto, nos enfocaremos en las primeras cinco etapas, que se consideran esenciales para el desarrollo del diseño propuesto. El esquema completo de la metodología y su flujo se muestra en la figura 3.1.



Figura 3.1: Etapas de la metodología de diseño a desarrollar. Fuente [4]

A continuación, se describen en detalle las distintas etapas que se seguirán en el desarrollo de este proyecto. Es importante destacar que en este capítulo solo se abordarán las primeras cuatro etapas de la metodología, mientras que el desarrollo de la quinta etapa será tratado en un capítulo posterior.

3.1 Etapa 0: Diagnóstico del Atta-Bot

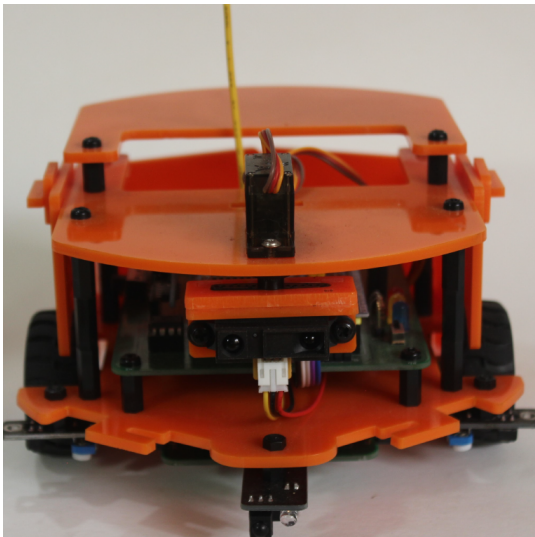
Antes de comenzar formalmente con la implementación de la metodología, es importante tener clara la problemática a solucionar. Por ello, se decidió iniciar con una evaluación exhaustiva del estado actual del Atta-Bot. Tal como se mencionó en el capítulo 1, este proyecto tiene como objetivo adaptar las capacidades de los robots Atta-Bot para desarrollar

un sistema de enjambre preciso y eficiente. En consecuencia, es esencial identificar las capacidades y limitaciones de la versión actual, lo que proporcionará un punto de referencia claro.

El Atta-Bot es un robot terrestre de tipo diferencial, diseñado específicamente para investigaciones en el campo de la robótica de enjambres. Este robot se destaca por su bajo costo, alrededor de USD 216.18 [6] y por ser de código abierto. Su estructura está compuesta principalmente de acrílico cortado con láser y espaciadores de nylon, lo que lo convierte en un robot fácil de replicar y ensamblar (figura 3.2). Sin embargo, uno de los retos de su construcción es la necesidad de utilizar una placa de circuito impreso (PCB) personalizada, aunque este componente, al igual que el resto del diseño y software, se encuentra disponible de manera pública en la página oficial del proyecto [20]. En la Tabla 3.1 se presentan las principales características que definen a este robot [5].

Tabla 3.1: Características de los Robots Atta-Bot.

Característica	Valor
Dimensiones	155 x 178 x 105 mm
Masa	455,3 g
Velocidad	108,6 mm/s
Autonomía	> 45 min



a) Vista frontal



b) Vista lateral

Figura 3.2: Imágenes del Atta-Bot. Fuente [5]

La arquitectura del Atta-Bot está basada en el uso de dos microprocesadores, cada uno con funciones específicas. Uno de los microprocesadores se encarga exclusivamente de la lectura de sensores y la detección de obstáculos, mientras que el otro gestiona tanto el control del movimiento del robot como la comunicación [5].

El microprocesador principal es el Adafruit Feather M0 RFM69, seleccionado debido a su bajo coste y a la integración de un módulo de radiofrecuencia, lo que facilita la comunicación inalámbrica entre los robots. Este microprocesador es el núcleo del sistema, encargándose de coordinar las acciones del robot y la transmisión de datos. El segundo microprocesador es el Blue Pill STM32, conocido por ser extremadamente económico. Además, debido a su simplicidad y eficiencia lo hacen ideal para manejar tareas como la lectura de sensores sin sobrecargar el sistema principal [5].

En cuanto a los sensores, el Atta-Bot está equipado con tres sensores IR. Dos de estos se utilizan para la detección de bordes en el suelo, ayudando al robot a evitar caídas o salirse de su área de operación. El tercer sensor IR está ubicado en la parte frontal y está diseñado para detectar obstáculos en su trayectoria. Adicionalmente, cuenta con un sensor Sharp GP2Y0A21YK, el cual es capaz de medir distancias y detectar obstáculos tanto en los laterales como en el frente del robot, esto debido a que está montado sobre un servomotor que puede girar hasta 180°, permitiendo una cobertura completa de la parte frontal del robot para una detección más precisa y dinámica [5].

Para el control del movimiento, este dispone de varios sensores propioceptivos. Cada una de sus ruedas está equipada con un encoder de efecto Hall, lo que permite medir el desplazamiento angular de cada rueda de manera independiente. Esto facilita un control más preciso del movimiento diferencial del robot. Además, está equipado con una IMU MPU6050, que cuenta con 6 grados de libertad, permitiendo medir la velocidad angular y la aceleración en los tres ejes. También incorpora un magnetómetro HMC5883L, que permite determinar la orientación del robot con respecto al campo magnético terrestre, esencial para el control de giros y navegación [5].

Estos sensores propioceptivos juegan un papel clave en el control preciso de los movimientos del Atta-Bot. Al ser un robot de tracción diferencial, solo cuenta con dos motores y una rueda libre tipo **ball caster**. Cada uno de los motores está controlado mediante un algoritmo PID discreto, lo que asegura un control preciso y exacto en sus movimientos, mejorando la estabilidad y eficiencia del robot en sus tareas de navegación [5]. En la tabla 3.2 se presentan los errores obtenidos con este sistema de control [21].

Tabla 3.2: Errores obtenidos para el sistema de control de navegación de la primera versión del Atta-Bot.

Experimento	Error (%)
Avanzar 30 cm	2,48
Avanzar 15 cm	2,74
Girar 90°	1,57
Girar -90°	1,58

En cuanto a la comunicación, el Atta-Bot está equipado con un módulo de radiofrecuencia que se utiliza para establecer una comunicación unidireccional con una estación base. A través de este enlace, el robot puede transmitir información clave, como su posición, estado, o si ha detectado algún obstáculo en su entorno. Para manejar esta comunicación de manera

eficiente y evitar colisiones o interferencias entre múltiples robots, el sistema emplea el protocolo TDMA. Esto significa que, antes de iniciar una prueba o tarea, los robots deben pasar por un proceso de configuración que asigna intervalos de tiempo específicos a cada uno para la transmisión, lo que garantiza que cada robot pueda enviar su información sin interferir con los demás [5]. En la tabla 3.3 se encuentra la tasa de éxito en la recepción de mensajes de este sistema [22].

Tabla 3.3: Tasa de éxito en la recepción paquetes para el sistema de comunicación de la primera versión del Atta-Bot.

Nodo	Tasa de éxito (%)
Base	96,85
Atta-Bot	93,96

A modo de resumen, en la tabla 4.1 se presenta una lista de las principales características que conforman esta versión del Atta-bot.

Tabla 3.4: Características de la primera versión del Atta-bot.

Característica	Componentes
Microcontrolador	Adafruit Feather M0 RFM69 Packet Radio
	Blue Pill STM32
Sensores de movimiento	Magnetómetro de triple eje HMC5883L
	IMU 6050
	Encoders
Sensores de detección	Sensor infrarrojo HW-488
	Sensor de medición Sharp GP2Y0A21YK
	2 sensores de seguimiento
Comunicación	Radiofrecuencia

Actualmente, el Atta-Bot cuenta con un comportamiento de enjambre implementado para la exploración aleatoria. Este comportamiento permite que los robots se desplacen de manera autónoma, explorando su entorno de forma aleatoria en busca de objetivos o zonas de interés. Además, el proceso de exploración se monitorea mediante un sistema de visión, el cual permite validar el movimiento y el desempeño de los robots. Este comportamiento ha demostrado una efectividad del 50% de área explorada utilizando tres robots durante un período de 5 minutos [5].

Este sistema de visión, desarrollado en Python, permite medir la efectividad del comportamiento de los robots al procesar un video grabado durante la prueba y calcular las posiciones de los robots para determinar el área de cobertura alcanzada [8]. Sin embargo, al utilizarse de forma diferida, tiene la desventaja de no proporcionar información en tiempo real al sistema, lo que dificulta la retroalimentación inmediata y la toma de decisiones durante las pruebas.

A pesar de las ventajas que ofrece el Atta-Bot en términos de exploración autónoma y bajo coste, también presenta varias desventajas que limitan su rendimiento actual. Un problema

recurrente es su geometría y la disposición de los sensores, lo que provoca que el robot se quede atrapado con elementos del escenario de pruebas con relativa frecuencia. Además, el sistema de detección de obstáculos, especialmente cuando varios robots interactúan en el mismo entorno, es deficiente lo que resulta en colisiones frecuentes entre los robots.

Otra limitación importante es la inestabilidad que se ha observado durante las pruebas, donde algunos robots se detienen inesperadamente sin razón aparente, lo que obliga a reiniciar las pruebas. Asimismo, aunque la presencia de dos microprocesadores (uno para el control de movimiento y otro para la detección) tiene sus ventajas, también complica la implementación de nuevos comportamientos y características, al requerir una mayor coordinación y complejidad en el desarrollo del software. Esto añade un desafío adicional en términos de integración y mantenimiento.

Como resultado de esta evaluación inicial, se ha obtenido una comprensión clara de las fortalezas y debilidades del Atta-Bot en su versión actual. Esto no solo establece un punto de referencia para el desarrollo de mejoras, sino que también permite identificar áreas críticas en las que es necesario intervenir para alcanzar los objetivos planteados en este proyecto.

3.2 Etapa 1: Identificar necesidades del cliente

En esta primera etapa de la metodología de Ulrich y Eppinger, el enfoque se centra en comprender a fondo las necesidades y expectativas del cliente. El proceso consta de tres pasos. Primero, se recopilará información relevante sobre el problema y la solución esperada por el cliente. Una vez obtenidos estos datos, se procederá a transformarlos en necesidades específicas. En el segundo paso, se realizará una categorización de estas necesidades, dividiéndolas en primarias y secundarias, agrupando aquellas que compartan características comunes. Finalmente, se llevará a cabo una encuesta al cliente para validar las necesidades identificadas y asegurar que se cubren todos los aspectos que desea solucionar. El cliente también asignará un valor de importancia a cada necesidad, lo que permitirá priorizar el trabajo de manera más organizada.

3.2.1 Recopilación de información e interpretación de necesidades

Para este paso, se realizó inicialmente una reunión introductoria con Cindy Calderón, asesora del grupo de investigación, con el fin de obtener una visión más clara y detallada del problema a resolver. Posteriormente, se llevó a cabo una segunda entrevista en la que se plantearon una serie de preguntas destinadas a profundizar en la comprensión del problema y las necesidades a abordar. Con base en las respuestas proporcionadas por la asesora durante esta etapa, se identificaron las necesidades, las cuales se presentan en la tabla 3.5.

Tabla 3.5: Necesidades determinadas tras el proceso de entrevistas.

Número de necesidad	Necesidad interpretada
1	El SD debe ser escalable y eficiente con diferentes cantidades de robots.
2	El robot debe de ser compacto en cuanto a dimensiones físicas.
3	EL robot es preciso con sus movimientos.
4	El robot tiene mejor o igual autonomía en cuanto a batería que la versión anterior.
5	El robot se puede comunicar con la base.
6	El robot se puede comunicar con otros robots.
7	El robot es capaz de evitar obstáculos.
8	El robot es capaz de identificar otros robots.
9	El SD es fácil de utilizar y configurar.
10	El SD es capaz de ejecutar al menos un comportamiento de enjambre.
11	El robot debe de reutilizar el hardware del Atta-Bot STEM, en la medida que se pueda.
12	El robot basa su funcionamiento en los comportamientos del Atta-Bot original.

3.2.2 Categorización de las necesidades interpretadas

Una vez se hayan definido e interpretado las necesidades, como segundo paso se realiza una categorización de estas, con el objetivo de organizarlas en necesidades primarias y secundarias, agrupando aquellas que compartan características en común. En la tabla 3.6 se presenta dicha categorización. Esto es de gran utilidad, ya que permite una visión más clara y estructurada de las prioridades del proyecto, facilitando así el enfoque en los aspectos clave que requieren mayor atención y garantizando un desarrollo más eficiente y ordenado del diseño.

Tabla 3.6: Categorización de las necesidades interpretadas.

Número de necesidad	Necesidad interpretada
1	El SD debe ser escalable y eficiente con diferentes cantidades de robots.
2	El robot debe ser compacto en cuanto a sus dimensiones físicas reutilizando el hardware del Atta-Bot STEM en la medida de lo posible.
2.a	El robot debe de ser compacto en cuanto a dimensiones físicas.
2.b	El robot debe de reutilizar el hardware del Atta-Bot STEM, en la medida que se pueda.
3	El robot es exacto con sus movimientos.
4	El robot tiene mejor o igual autonomía en cuanto a batería que la versión anterior.
5	El robot se puede comunicar con la base y con otros robots.
5.a	El robot se puede comunicar con la base.
5.b	El robot se puede comunicar con otros robots.
6	El robot es capaz de evitar obstáculos e identificar si es otro robot.
6.a	El robot es capaz de evitar obstáculos.
6.b	El robot es capaz de identificar otros robots.
7	El SD es fácil de utilizar y configurar.
8	El SD debe ser capaz de ejecutar al menos un comportamiento de enjambre, basándose en los comportamientos originales del Atta-Bot.
8.a	El SD es capaz de ejecutar al menos un comportamiento de enjambre.
8.b	El robot basa su funcionamiento en los comportamientos del Atta-Bot original.

3.2.3 Jerarquización de necesidades

Como último paso de esta primera etapa, se envió una encuesta al cliente con el propósito de validar si las necesidades planteadas en la tabla 3.6 cumplen con sus expectativas. Además, se solicitó al cliente que indicara la importancia de cada necesidad, con el fin de jerarquizarlas y determinar a cuáles se les debe prestar mayor atención durante el desarrollo del proyecto. Para ello, se utilizó una escala del 1 al 5, detallada a continuación:

1. La función es indeseable. No consideraría una solución con esta función.
2. La función no es importante, pero no me importaría tenerla.
3. Sería bueno tener esa función, pero no es necesaria.
4. La función es altamente deseable, pero consideraría una solución sin ella.
5. La función es de importancia crítica. No consideraría una solución sin esta función.

A continuación, se presenta en la tabla 3.7 la lista final de necesidades, detallada junto con la importancia asignada a cada una de ellas por el cliente.

Tabla 3.7: Necesidades finales y su importancia en la solución.

Número de necesidad	Necesidad interpretada	Importancia
1	El SD debe ser escalable y eficiente con diferentes cantidades de robots.	5
2	El robot debe ser compacto en cuanto a sus dimensiones físicas reutilizando el hardware del Atta-Bot STEM en la medida de lo posible.	3
3	El robot es exacto con sus movimientos.	5
4	El robot tiene mejor o igual autonomía en cuanto a batería que la versión anterior.	4
5	El robot se puede comunicar con la base y con otros robots.	5
6	El robot es capaz de evitar obstáculos e identificar si es otro robot.	5
7	El SD es fácil de utilizar y configurar.	4
8	El SD debe ser capaz de ejecutar al menos un comportamiento de enjambre, basándose en los comportamientos originales del Atta-Bot.	5

Por último, tal y como se puede observar en la tabla 3.7, ninguna de las necesidades tiene una importancia tan baja como para no ser tomada en cuenta para el desarrollo de la solución, por este motivo se utilizará esta tabla como base para priorizar el desarrollo de las funcionalidades, garantizando que se cumplan las expectativas del cliente de manera adecuada y eficiente.

3.3 Etapa 2: Establecer especificaciones objetivo

Una vez se identifican las necesidades del cliente y se asigna su respectiva importancia, es crucial establecer métricas que permitan cuantificar el grado de cumplimiento de cada una. Estas métricas serán fundamentales para evaluar las soluciones propuestas en las etapas posteriores, ayudando a reducir el número de opciones viables. A cada métrica se le definirá un valor ideal y un valor marginal, los cuales servirán para determinar si la solución satisface o no las necesidades del cliente. Por lo tanto, es esencial que cada necesidad tenga al menos una métrica asociada, asegurando así la capacidad de medir objetivamente el cumplimiento de los requerimientos del cliente.

Por consiguiente, se han definido métricas específicas para evaluar cada una de las necesidades descritas en la tabla 3.7, las cuales se detallan en la tabla 3.8. Muchos de los valores establecidos como márgenes provienen de la versión anterior del Atta-Bot, mientras que otros fueron obtenidos directamente del cliente durante reuniones o a través de investigación adicional. Cabe resaltar que, como se muestra en la tabla 3.8, cada métrica está asociada a un nivel de importancia, el cual es heredado de la necesidad de la que procede.

Tabla 3.8: Métricas y valores ideales e marginales.

Número de métrica	Necesidad asociada	Métrica	Importancia	Unidades	Valor marginal	Valor ideal
1	1	Tiempo de procesado de un fotograma	5	<i>ms</i>	≤ 333	< 200
2	2	Tamaño del robot	3	<i>mm</i>	$\leq 155 \times 178$	$< 108 \times 124$
3	3	Error de posición promedio	5	%	≤ 10	$< 2,74$
4	3	Error de ángulo promedio	5	%	≤ 10	$< 1,58$
5	4	Duración de la batería en uso	4	<i>min</i>	≥ 45	> 90
6	5	Tasa de éxito en la transmisión de mensajes	5	%	≥ 95	$> 96,9$
7	6	Tasa de éxito en la identificación y evasión de obstáculos	5	%	≥ 90	> 95
8	7	Nivel de satisfacción del usuario al usar el SD	4	número	≥ 3	≥ 4
9	8	Número de comportamientos de enjambre implementados	5	número	1	≥ 2

La primera métrica es el tiempo de procesamiento de un fotograma, que está directamente relacionado con la escalabilidad del sistema (necesidad número uno). Esta métrica es crucial porque determina la capacidad de la base para gestionar simultáneamente varios robots sin crear cuellos de botella. Si la base no puede procesar las posiciones y comunicarlas a los robots en el tiempo estipulado, la sincronización y coordinación del enjambre se verán comprometidas, afectando negativamente la eficiencia global del sistema.

Para mantener el sistema funcionando de manera óptima, se debe garantizar que el tiempo de procesamiento esté dentro de un margen adecuado, por lo que para el valor ideal se realizó una estimación basada en un escenario con una tasa de actualización de la posición de 5 veces por segundo por cada robot. El valor marginal se definió con una tasa de actualización de 3

veces por segundo, lo que ofrece un equilibrio entre precisión y capacidad de procesamiento, asegurando el buen desempeño del enjambre.

La segunda métrica es el tamaño del robot, relacionada con la necesidad número dos. Durante una entrevista, el cliente manifestó su preferencia por un robot más compacto que el modelo anterior. Por este motivo, se estableció como valor marginal las dimensiones del robot actual, mientras que el valor ideal se fijó en una reducción del 30%. Es importante destacar que solo se prestará atención a las medidas de ancho y largo del robot, ya que la altura no representa un problema significativo para el cliente.

Las métricas tres y cuatro se refieren al error de posición tras mover el robot una distancia determinada y el error tras rotar un ángulo específico, respectivamente. Ambas están relacionadas con la necesidad número tres, que aborda la exactitud de los movimientos del robot. Para estas métricas, se consideró como valor ideal los resultados obtenidos con la versión 1 del Atta-Bot, ya que estos proporcionan una referencia en términos de exactitud, asegurando que el nuevo diseño mantenga el nivel de exactitud en sus desplazamientos y rotaciones.

La quinta métrica, que mide la duración de la batería en uso, está vinculada a la necesidad número cuatro, la cual establece que el robot debe tener una autonomía igual o superior a la de su predecesor. Por esta razón, se ha seleccionado como valor marginal la duración de la batería de la versión anterior, utilizándola como referencia para asegurar que el nuevo diseño cumpla con el requisito mínimo de autonomía, mientras que el valor ideal se orientará a mejorar dicha duración.

La sexta métrica evalúa la tasa de éxito en la transmisión de mensajes y está directamente vinculada con la necesidad número cinco, que aborda la comunicación tanto entre los robots como con la base. Esta métrica es fundamental, ya que una comunicación deficiente podría comprometer el comportamiento coordinado del enjambre. Al igual que la métrica relacionada con el tiempo de procesamiento, esta también podría convertirse en un factor crítico para el funcionamiento óptimo del sistema. Como valor ideal, se tomó como referencia los resultados obtenidos con la primera versión del Atta-Bot.

La métrica que sigue aborda la necesidad número seis, que se refiere a la capacidad del robot para evitar obstáculos y diferenciar si estos son otros robots. Esta métrica evalúa la tasa de éxito en la identificación y evasión de obstáculos, lo cual es crucial para asegurar un comportamiento autónomo eficiente y evitar colisiones dentro del entorno operativo.

La siguiente métrica se enfoca en medir el nivel de satisfacción del usuario al utilizar el sistema, vinculada a la necesidad número siete, que hace referencia a la facilidad de uso y configuración del sistema. Un sistema intuitivo y sencillo de configurar es esencial para garantizar una experiencia positiva y eficiente para el usuario.

Finalmente, la métrica número nueve se refiere al número de comportamientos de enjambre implementados en el sistema. Esta métrica está vinculada a la necesidad número nueve, que establece que el sistema debe ser capaz de ejecutar al menos un comportamiento de enjambre, tomando como base los comportamientos originales de la primera versión del Atta-Bot.

A modo de resumen, la tabla 3.9 presenta una matriz de correlación entre las necesidades identificadas y las métricas previamente explicadas. Esta matriz tiene como objetivo proporcionar una visión clara de las métricas asignadas a cada necesidad, asegurando que todas las necesidades estén adecuadamente cubiertas por al menos una métrica, sin dejar ningún aspecto sin evaluar.

Tabla 3.9: Matriz de correlación entre las métricas y las necesidades.

		Métrica								
		1	2	3	4	5	6	7	8	9
Necesidad	1	X								
	2		X							
	3			X	X					
	4					X				
	5						X			
	6							X		
	7								X	
	8									X

3.4 Etapa 3: Generación de conceptos

En esta etapa, el primer paso consiste en descomponer el problema principal en subproblemas específicos. Este enfoque permite desarrollar soluciones distintas para cada subproblema, lo que resulta en un diseño más integral al ofrecer múltiples combinaciones de soluciones orientadas a satisfacer las necesidades del cliente. Además, al dividir el problema en componentes más manejables, se minimiza el riesgo de errores que puedan surgir de decisiones incorrectas durante el diseño. Si es necesario realizar ajustes, solo se requerirá modificar la sección específica del diseño en lugar de rehacer el sistema completo.

El segundo paso implica una búsqueda exhaustiva, tanto interna como externa, de posibles soluciones para cada subproblema. Este proceso busca ampliar el rango de conceptos disponibles y fomentar la innovación. Finalmente, se desarrollan varios conceptos de soluciones potenciales, los cuales serán evaluados y filtrados en una fase posterior para seleccionar la mejor opción.

3.4.1 Descomposición funcional del sistema

Para desarrollar un diseño más completo, es fundamental descomponer el problema principal en subproblemas específicos. Este enfoque permite crear soluciones personalizadas para cada aspecto del problema, reduciendo así el riesgo de cometer errores por malas decisiones durante el proceso de diseño.

Es importante destacar que el problema se divide en dos componentes independientes que trabajan en conjunto: por un lado, el robot, encargado de ejecutar los comportamientos de enjambre, y por otro, la base, responsable de determinar y transmitir la posición a cada robot. Tal como se puede observar en la figura 3.3, en ella se ilustran los subproblemas que abarcan cada una de estas partes.

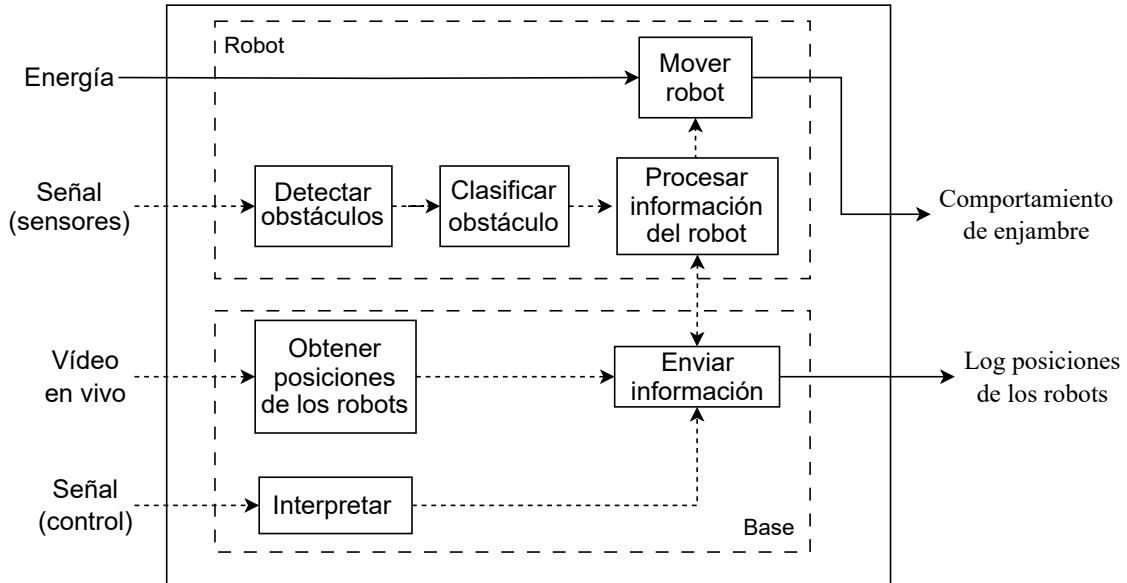


Figura 3.3: Descomposición funcional del problema a resolver.

Como se muestra en la figura 3.3, la descomposición funcional del sistema incluye cuatro entradas principales: energía, señales de los sensores, vídeo en tiempo real y señales de control enviadas por el usuario. Las salidas del sistema son dos: el comportamiento de enjambre y un registro de las posiciones de los robots.

El sistema se ha dividido en siete subproblemas, sin embargo, para efectos del desarrollo de este proyecto, los tres subproblemas relacionados con la base se abordarán como uno solo. Esto se debe a que el único aspecto que se trabajará es el envío de información, mientras que los otros dos subproblemas se basarán en un trabajo previo realizado por el grupo de investigación. Dicho trabajo será el sistema de visión que se mencionó anteriormente [8]. A continuación, se proporciona una breve descripción de cada subproblema.

- **Detectar obstáculos:** Este subproblema se centra en el uso de sensores para identificar objetos en el entorno que puedan bloquear el camino del robot. Los sensores recogen información sobre el entorno inmediato, lo que permite al sistema determinar la presencia de cualquier objeto frente al robot.
- **Clasificar obstáculo:** Una vez detectado el obstáculo, este subproblema se enfoca en clasificarlo. Aquí, el sistema debe diferenciar si el objeto es otro robot o un elemento del entorno. Esta clasificación es fundamental para ajustar la respuesta del robot, ya que la estrategia de evasión puede variar según el tipo de obstáculo.

- **Mover robot:** Este subproblema está relacionado con el control preciso de los motores que permiten los desplazamientos del robot. Involucra la regulación de los motores para que el robot se mueva con precisión según las órdenes recibidas, ya sea para desplazarse linealmente o girar.
- **Procesar información del robot:** En este subproblema se aborda la gestión de la información interna del robot, incluidas las posiciones, el estado de los sensores y otros datos cruciales. Además, se coordina la comunicación entre los robots para permitir una cooperación efectiva.
- **Enviar información:** Este subproblema se encarga de la comunicación entre la base y los robots. Incluye el envío de órdenes desde la base a los robots y la recepción de información de retorno sobre el estados de los robots.

3.4.2 Búsqueda interna y externa de conceptos

En esta sección se exploran diversas alternativas y conceptos que podrían ofrecer soluciones efectivas a los cinco subproblemas planteados previamente. A continuación, se detallan las opciones consideradas para abordar cada subproblema.

Detectar obstáculos

Para este subproblema, se llevó a cabo únicamente una búsqueda interna de conceptos, en la que se evaluaron alternativas conocidas tanto por el estudiante responsable del proyecto como por el cliente. Estas alternativas provienen de implementaciones previas utilizadas en otras versiones del Atta-Bot o de su popularidad en aplicaciones similares. A continuación, se presenta una lista de los sensores considerados mediante esta búsqueda interna para abordar el problema, junto con una breve descripción de cada uno:

1. **APDS9960:** Sensor de proximidad, gestos y color que utiliza un sensor óptico para detectar objetos cercanos y reconocer movimientos gestuales.
2. **HW-488:** Módulo sensor de obstáculos infrarrojo, capaz de detectar la presencia de objetos cercanos mediante la reflexión de luz infrarroja.
3. **Sharp GP2Y0A21YK:** Sensor de distancia infrarrojo que mide la distancia a objetos utilizando un haz de luz infrarroja.
4. **HC-SR04:** Sensor ultrasónico que mide distancias mediante el uso de ondas de sonido.
5. **VL53L0X:** Sensor de distancia basado en tecnología ToF, que ofrece mediciones precisas de corta distancia.

Clasificar obstáculo

Para este subproblema, se decidió utilizar tanto la búsqueda interna como la búsqueda externa. En el caso de la búsqueda externa, solo surgió una opción la de usar BLE para detectar dispositivos cercanos. El resto de las alternativas provienen de la búsqueda interna. A continuación, se presenta una lista de los resultados obtenidos:

1. **APDS9960:** Utilizar este sensor para detectar el color del obstáculo. Dependiendo de si el color coincide con un valor específico, se podría determinar si el obstáculo es otro robot.
2. **Compartir pose actual:** Este método consiste en que el robot que detecta el obstáculo comparte su ubicación con el resto del grupo. A través de cálculos geométricos, se determina si el obstáculo es otro robot. La pose se refiere a su posición en (x, y) y el ángulo del robot respecto a la horizontal.
3. **BLE:** El microcontrolador utilizado cuenta con esta tecnología y según las referencias [17] [18], se determinó que teóricamente es posible identificar si el obstáculo es otro robot mediante el uso de esta tecnología.

Es importante destacar que no se consideró la opción de utilizar Wi-Fi para hacer algo similar a BLE. Esto se debe a que Wi-Fi está diseñado para distancias más largas y su uso de RSSI en cortas distancias no es preciso, lo que es fundamental para este subproblema [23].

Mover robot

Para este subproblema, se realizó únicamente una búsqueda interna de conceptos, evaluando alternativas provenientes de implementaciones previas en otras versiones del Atta-Bot. A continuación, se presenta una lista de los resultados obtenidos en esta búsqueda interna, junto con una breve descripción de cada opción:

1. **Encoder + PID:** Este método combina un encoder, que mide la posición o velocidad de los motores del robot, con un controlador PID. El encoder proporciona retroalimentación sobre el movimiento del robot, mientras que el PID ajusta la velocidad de los motores.
2. **IMU 6050 + Magnetómetro:** Esta alternativa incluye IMU que mide la aceleración y la rotación del robot, junto con un magnetómetro que proporciona información sobre la orientación del robot.
3. **IMU ICM-20948:** Una IMU que combina acelerómetro, giroscopio y magnetómetro en un solo módulo.

Procesar información del robot

Como se mencionó anteriormente, este subproblema se encarga tanto de la gestión del robot y además, de la comunicación entre ellos. Se decidió utilizar el IDE de Arduino para gestionar el robot, ya que las versiones anteriores del Atta-Bot fueron desarrolladas con este entorno, lo que facilita la programación al contar con una base sólida desde la cual partir. Para la comunicación entre los robots, se realizó una búsqueda externa para evaluar los posibles protocolos de comunicación que ofrece el microcontrolador utilizado. A continuación se presentan los resultados:

1. **WiFi TCP/IP**: Protocolo que permite conexiones estables y confiables. Requiere una configuración de cliente-servidor.
2. **WiFi UDP**: Protocolo rápido y eficiente, pero menos confiable en cuanto a la pérdida de datos. No requiere una configuración de cliente-servidor.
3. **Bluetooth**: Protocolo de corto alcance que permite conexiones punto a punto o entre grupos pequeños.
4. **ESP-NOW**: Protocolo de comunicación rápido y eficiente, diseñado para conexiones directas entre dispositivos. Requiere conocer la dirección MAC de los otros robots para establecer la comunicación.

Enviar información

Para este subproblema en particular, se optó por seleccionar un subconjunto de las soluciones propuestas en el subproblema anterior, dado que es crucial garantizar que tanto los robots como la base puedan manejar los mismos protocolos de comunicación. Los resultados seleccionados son los siguientes:

1. **WiFi TCP/IP**
2. **WiFi UDP**
3. **Bluetooth**

A modo de resumen, en la tabla 3.10 se presentan todas las posibles soluciones identificadas para cada uno de los subproblemas. A partir de esta tabla, se llevarán a cabo las combinaciones pertinentes para generar los distintos conceptos de solución.

Tabla 3.10: Soluciones de subproblemas para la formulación de conceptos.

Detectar obstáculos	Clasificar obstáculo	Mover robot	Procesar información del robot	Enviar información
APDS9960	APDS9960	Encoder + PID	WiFi TCP/IP	WiFi TCP/IP
HW-488	Compartir pose actual		WiFi UDP	WiFi UDP
Sharp GP2Y0A21YK		IMU 6050 + Magnetómetro	Bluetooth	
HC-SR04	BLE	IMU	ESP-NOW	Bluetooth
VL53L0X		ICM-20948		

3.4.3 Elaboración de conceptos

Una vez definidos varios conceptos para cada subproblema, se procede a presentar los posibles conceptos a solución que puedan cumplir de cierta manera con las métricas previamente planteadas. Para ello, se utilizó la tabla 3.10 como referencia y aplicando el criterio experto, se generaron seis conceptos a solución. Estos serán evaluados y filtrados posteriormente para identificar la mejor opción. En la tabla 3.11 se presentan los seis conceptos propuestos.

Tabla 3.11: Posibles soluciones candidatas al problema a resolver.

Concepto	Detectar obstáculos	Clasificar obstáculo	Mover robot	Procesar información del robot	Enviar información
A	VL53L0X	Compartir pose actual	Encoder + PID	ESP-NOW	WiFi UDP
B	HW-488	Compartir pose actual	IMU ICM-20948	WiFi UDP	WiFi UDP
C	Sharp GP2Y0A21YK	BLE	IMU 6050 + Magnetómetro	WiFi UDP	WiFi TCP/IP
D	APDS9960	BLE	Encoder + PID	Bluetooth	WiFi UDP
E	APDS9960	Compartir pose actual	Encoder + PID	WiFi TCP/IP	WiFi TCP/IP
F	HC-SR04	BLE	IMU 6050 + Magnetómetro	Bluetooth	Bluetooth

3.5 Etapa 4: Selección de conceptos

En esta etapa de la metodología, se llevará a cabo un proceso de filtrado para seleccionar el concepto más adecuado para resolver el problema planteado. Este proceso se fundamentará en las necesidades previamente definidas y mediante el uso de matrices de evaluación, se identificará el concepto que mejor se ajuste a la solución esperada.

3.5.1 Filtrado de conceptos

Como primer paso se tiene el filtrado de conceptos, para lo cual se estableció una lista de criterios basada en las necesidades identificadas en la tabla 3.7. El objetivo de esta fase fue evaluar cada concepto de manera exhaustiva, asegurando que se consideraran todas las necesidades y expectativas del cliente.

La evaluación se realizó a través de una matriz de decisión, donde se utilizó un concepto de referencia para comparar los conceptos, siendo en este caso la primera versión del Atta-bot. A cada criterio se le asignó un + si el concepto superaba al de referencia, un 0 si era comparable, y un - si el concepto de referencia resultaba superior. Al finalizar este proceso, se analizaron los resultados para identificar posibles combinaciones de conceptos y determinar cuáles avanzarían a la siguiente etapa. La tabla 3.12 muestra la matriz de filtrado empleada en esta evaluación.

Tabla 3.12: Matriz de filtrado de conceptos.

Criterio de selección	Conceptos						
	A	B	C	Atta-Bot V1	D	E	F
El SD debe ser escalable y eficiente con diferentes cantidades de robots.	+	+	-	0	-	-	-
El robot debe ser compacto en cuanto a sus dimensiones físicas reutilizando el hardware del Atta-Bot STEM en la medida de lo posible.	0	+	0	0	+	+	0
El robot es preciso con sus movimientos.	+	-	-	0	0	+	-
El robot tiene mejor o igual autonomía en cuanto a batería que la versión anterior.	0	0	0	0	0	0	0
El robot se puede comunicar con la base y con otros robots.	+	+	+	0	+	+	+
El robot es capaz de evitar obstáculos e identificar si es otro robot.	+	+	+	0	+	+	+
El SD es fácil de utilizar y configurar.	+	+	+	0	+	+	+
El SD debe ser capaz de ejecutar al menos un comportamiento de enjambre, basándose en los comportamientos originales del Atta-Bot.	0	0	0	0	0	0	0
Suma +	5	5	3	0	4	5	3
Suma 0	3	2	3	8	3	2	3
Suma -	0	1	2	0	1	1	2
Evaluación neta	5	4	1	0	3	4	1
Lugar	1	2	5	7	4	2	5
¿Continúa?	Si	Combinar	No	No	Combinar	Si	No

En la tabla anterior se puede observar que el concepto *A* fue el ganador con cuatro puntos, mientras que los conceptos *B* y *E* empataron en el segundo lugar. El concepto *D* ocupó el cuarto puesto, y los conceptos *C* y *F* fueron los peor evaluados. A partir de estos resultados, se decidió avanzar con los conceptos *A* y *E*, además de una combinación de los conceptos *B* y *D*, ya que ambos presentaron características interesantes que, al fusionarse, podrían formar una solución superior.

Este nuevo concepto, denominado *BD*, se presenta en la tabla 3.13. Cabe destacar que, para el subproblema detectar obstáculos, se decidió implementar ambas soluciones propuestas, ya que no son mutuamente excluyentes y, al combinarse, mejoran la funcionalidad del sistema. Lo mismo se aplicó al subproblema clasificar obstáculo. Esta combinación asegura que se aprovechen las fortalezas de cada concepto original, resultando en un diseño más robusto y eficiente.

Tabla 3.13: Resultados de la combinatoria de B y D.

Concepto	Detectar obstáculos	Clasificar obstáculo	Mover robot	Procesar información del robot	Enviar información
BD	HW-488 + APDS9960	Compartir pose actual + BLE	Encoder + PID	WiFi UDP	WiFi UDP

3.5.2 Selección del concepto ganador

Una vez filtrados los conceptos, se procede a la selección del concepto final que será desarrollado. Para esta selección, se volverán a emplear los mismos criterios utilizados en el paso de filtrado, pero en esta ocasión se asignará un peso relativo a cada criterio, basado en la importancia definida en la tabla 3.7.

A diferencia de la etapa anterior, en la que se utilizaba un único concepto como referencia, en esta fase cada criterio podrá tener su propia referencia. Esto permitirá realizar una evaluación más precisa y detallada de cada concepto, considerando sus características específicas. La escala utilizada para evaluar los conceptos será la siguiente:

1. Es notablemente menor que la referencia.
2. Es ligeramente menor que la referencia.
3. Es la referencia.
4. Es ligeramente superior que la referencia.
5. Es notablemente superior que la referencia.

En la tabla 3.14 se presenta la evaluación detallada de los conceptos. Según esta evaluación, la combinación *BD* se destacó como la opción más robusta y efectiva, siendo seleccionada como la solución final. La tabla 3.13 detalla la solución propuesta en este concepto.

Tabla 3.14: Evaluación de conceptos.

		Concepto					
		A		BD		E	
Criterio de selección	Peso	C	C_w	C	C_w	C	C_w
El SD debe ser escalable y eficiente con diferentes cantidades de robots	14%	3	0,42	4	0,56	2	0,28
El robot debe ser compacto en cuanto a sus dimensiones físicas reutilizando el hardware del Atta-Bot STEM en la medida de lo posible	8%	3	0,24	3	0,24	3	0,24
El robot es preciso con sus movimientos	14%	3	0,42	3	0,42	3	0,42
El robot tiene mejor o igual autonomía en cuanto a batería que la versión anterior	11%	3	0,33	3	0,33	3	0,33
El robot se puede comunicar con la base y con otros robots	14%	3	0,42	3	0,42	3	0,42
El robot es capaz de evitar obstáculos e identificar si es otro robot	14%	3	0,42	5	0,7	2	0,28
El SD es fácil de utilizar y configurar	11%	3	0,33	3	0,33	3	0,33
El SD debe ser capaz de ejecutar al menos un comportamiento de enjambre, basándose en los comportamientos originales del Atta-Bot	14%	3	0,42	3	0,42	3	0,42
Total puntos		3		3,42		2,72	
Lugar		2		1		3	
¿Continua?		No		Desarrollar		No	

Por lo tanto, este concepto será desarrollado para abordar el problema planteado. El desarrollo detallado de la solución se abordará en el próximo capítulo del documento.

Capítulo 4

Desarrollo del Concepto Ganador

Esta sección describe detalladamente el procedimiento y el trabajo realizado para desarrollar el diseño de la solución propuesta al problema. Inicialmente, se presentan algunas consideraciones previas que son esenciales para el correcto desarrollo de la solución. Luego, se detalla la propuesta, abordando la implementación de cada subproblema individualmente.

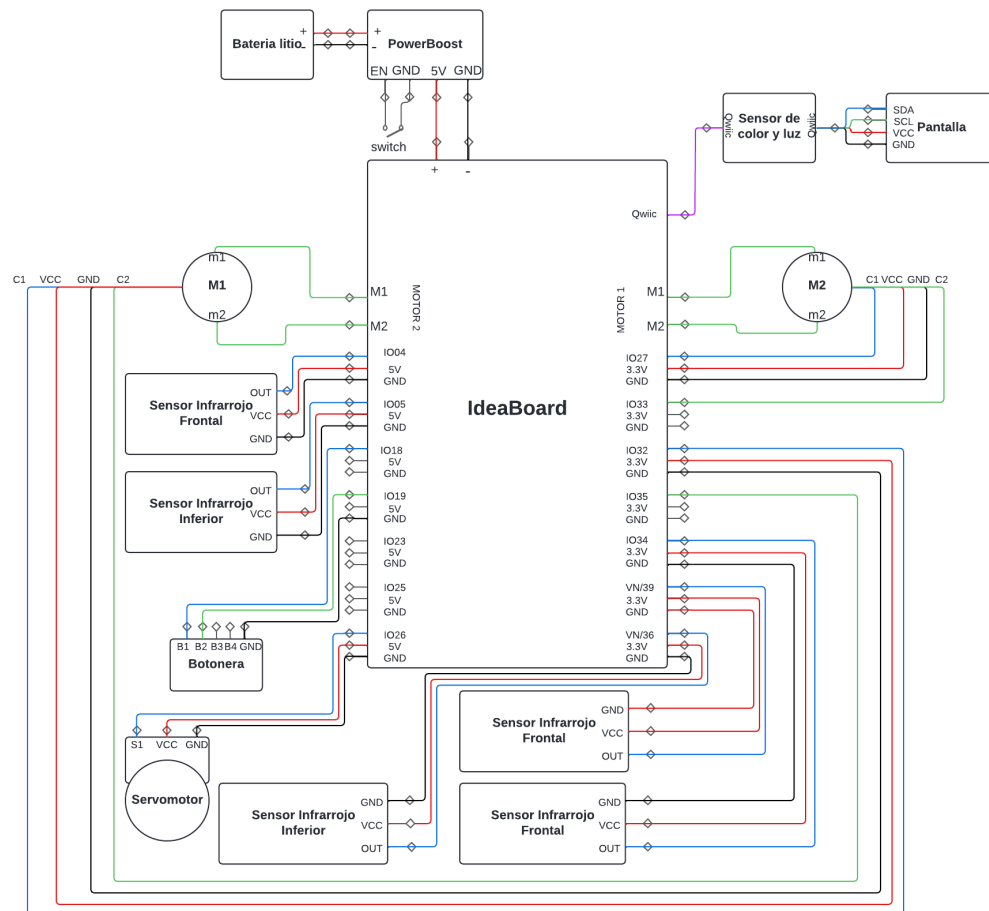
4.1 Consideraciones de diseño

Como se mencionó en el capítulo 1, el objetivo principal de este proyecto es reutilizar la arquitectura de hardware desarrollada para la versión 1 del Atta-Bot STEM. Para lograr esto, es fundamental realizar un análisis detallado de la arquitectura actual, con el fin de identificar qué componentes pueden ser reutilizados sin modificaciones y cuáles deben ser ajustados o reemplazados para cumplir con los nuevos requisitos del sistema de enjambre propuesto. Este análisis también permitirá determinar qué elementos deben incorporarse o eliminarse para optimizar el rendimiento del sistema.

En la tabla 4.1 se presenta una lista detallada de todos los componentes utilizados en la construcción de la versión 1 del Atta-Bot STEM [6]. Además, en la figura 4.1 se muestra la arquitectura de hardware seleccionada para dicha versión, la cual servirá como base para el desarrollo de esta nueva iteración. El objetivo de esta nueva versión no es solo optimizar el uso de los recursos existentes, sino también garantizar que el sistema sea escalable y eficiente en un entorno de robots en enjambre.

Tabla 4.1: Componentes utilizados en la primera versión del Atta-Bot STEM.

Cantidad	Componente	Precio
2	Motor DC con enconder y rueda N20	\$ 14.95
2	Suporte de motor	\$ 0.75
1	Batería de litio de 3.7 V y 2500mAh	\$ 17.95
1	PowerBoost de 5V y 1000mA	\$ 10.95
1	Ball Caster	\$ 2.75
1	IdeaBoard	\$ 39.95
1	Micro Servomotor MG90S	\$ 6.95
2	Módulo de sensor infrarrojo HW-488	\$ 2.95
2	Módulo de sensor de seguimiento	\$ 1.95
1	Sensor de APDS9960	\$ 11.95
1	Pantalla	\$ 5.95
1	Interruptor general	\$ 1.25
1	Keypad	\$ 1.95
1	Kit conectores	\$ 8.95
1	Kit de espaciadores y tornillos	\$ 15.95
1	Cable	\$ 3.95
1	STEMMA QT cable	\$ 1.49
1	STEMMA QT cable macho-macho	\$ 1.95

**Figura 4.1:** Diagrama electrónico del Atta-Bot STEM. Fuente [6]

A partir de la arquitectura presentada anteriormente y con base en la selección de componentes realizada en el capítulo anterior, se detalla a continuación el listado final de los elementos que se utilizarán para el desarrollo de la nueva versión del Atta-Bot:

- 1 IdeaBoard como unidad principal de control y donde encuentran los controladores para los motores.
- 2 motores CD con encoder, junto con sus respectivas bases de montaje, para proporcionar el movimiento al robot.
- 1 ball caster para proporcionar estabilidad al robot durante sus desplazamientos sin afectar la maniobrabilidad de las ruedas.
- 1 batería de litio, acompañada de un módulo PowerBoost para la gestión de la energía y garantizar una adecuada autonomía del sistema.
- 1 interruptor para encender y apagar el robot.
- 1 servo motor para controlar el movimiento del sensor frontal.
- 1 sensor APDS9960 para la detección y medición de distancias.
- 2 sensores HW-488 para la detección de obstáculos.

Este conjunto de componentes constituye la base que asegura que el robot mantenga un rendimiento óptimo mientras se adapta a las nuevas funcionalidades necesarias para el sistema de enjambre. Como se puede observar, se logra una reutilización del 100% de los componentes respecto a la versión 1 del Atta-Bot STEM, lo que significa que no es necesario incorporar nuevos elementos para esta actualización. Esta decisión no solo optimiza la eficiencia en términos de costos, sino que también simplifica el proceso de implementación. Por lo que, con la selección de componentes definida, se puede proceder a dar inicio al desarrollo de la solución propuesta.

Con respecto, al diseño de la carcasa y el chasis del robot, se procederá a hacer una desde cero, ya que no es posible reutilizar los de la versión 1 del Atta-Bot. Esto se debe a que los nuevos componentes requieren un rediseño para adaptarse, y la estructura anterior resulta demasiado voluminosa. Es importante señalar que en la versión 1 del Atta-Bot STEM no se realizó este paso, por lo que es indispensable llevarlo a cabo en esta nueva versión para cumplir con las expectativas del cliente.

Para el diseño del chasis, se optará por un robot de tipo diferencial, ya que los componentes del Atta-Bot STEM están seleccionados específicamente para este tipo de configuración. Además, el diseño diferencial ofrece varias ventajas, como la simplicidad en el control de movimiento y una mayor maniobrabilidad en entornos complejos. Con estos elementos en consideración, se avanzará hacia la etapa de diseño mecánico.

4.2 Distribución de componentes en el robot

Una vez seleccionados los componentes y definido el tipo de robot a desarrollar, el siguiente paso consistió en la búsqueda de modelos 3D para cada uno de los elementos elegidos. Sin embargo, en el caso del IdeaBoard, fue necesario crear un modelo desde cero, ya que al ser una placa relativamente nueva, no se pudo encontrar su modelo 3D disponible en línea. Esto se realizó con el objetivo de lograr un diseño lo más compacto posible y obtener una visión clara de cómo resultaría la disposición final de los componentes, permitiendo además ahorrar tiempo y costos al realizar múltiples iteraciones sin la necesidad de construir físicamente cada versión.

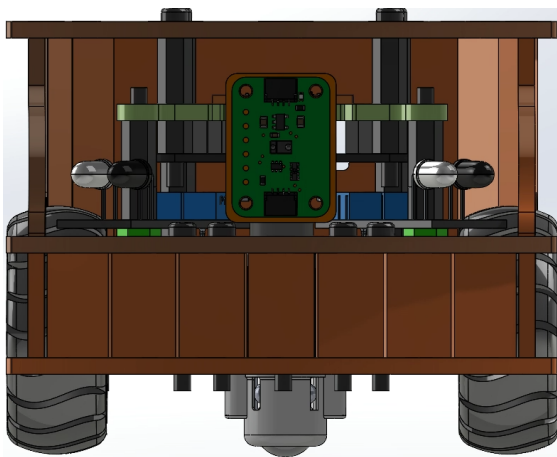
Para la creación del diseño 3D, se utilizó el software SolidWorks, ya que se contaba con la licencia necesaria y es una herramienta en la que resulta relativamente sencillo encontrar modelos 3D de componentes comunes. Esto permitió trabajar de manera eficiente en el desarrollo del prototipo.

Con los modelos 3D de todos los componentes listos, se procedió a realizar varias iteraciones del diseño del robot. El enfoque principal fue garantizar que el robot fuera lo más compacto posible, sin comprometer su funcionalidad. A lo largo del proceso, se realizaron entrevistas con el cliente para obtener retroalimentación sobre las distintas versiones del diseño. Tras varias iteraciones y ajustes, se llegó a un diseño final que cumple con los requisitos planteados, el cual se presenta en la figura 4.2.

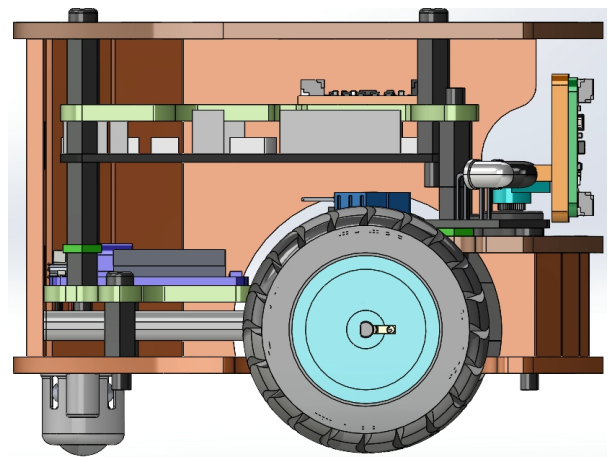
Como se puede observar en la figura anterior, el diseño propuesto está pensado para ser fabricado con piezas de acrílico de 3 mm cortadas con láser y utilizando espaciadores de nylon, siguiendo la misma línea que su versión anterior. Este enfoque no solo mantiene el costo bajo, sino que también facilita la replicación del robot de manera sencilla y eficiente. En cuanto a las dimensiones, la nueva versión presenta un tamaño compacto de 100 mm de ancho, 108 mm de largo y 82 mm de alto, por lo que esta dentro de las métricas especificadas en la tabla 3.8.

En la figura 4.2, se puede apreciar la disposición interna del robot, donde se ha diseñado una estructura en varios niveles. En el nivel más bajo se encuentran la batería y los motores, lo que permite mantener el centro de masa lo más bajo posible, favoreciendo una mayor estabilidad durante los movimientos. En el siguiente nivel está el módulo PowerBoost, mientras que en el nivel superior se ubica el IdeaBoard. Esta disposición por niveles no solo optimiza el uso del espacio, sino que también contribuye a reducir las dimensiones generales del robot. Es importante señalar que, para estas capturas, se han ocultado o eliminado algunas partes de la carcasa con el fin de ofrecer una visión más clara del interior del robot y su distribución de componentes.

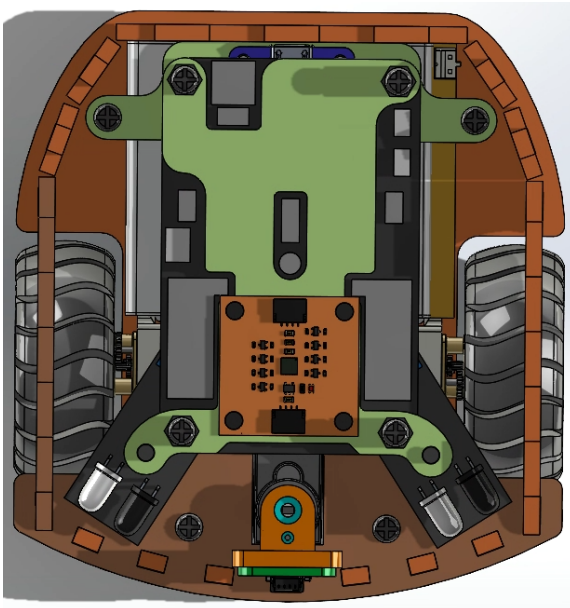
Esta disposición de los componentes optimiza el acceso a las partes esenciales del robot. Sin necesidad de desmontar el chasis, es posible acceder de manera directa al puerto de carga, así como al puerto utilizado para la programación del robot. Además, el interruptor de encendido está ubicado en una posición accesible, mejorando significativamente la usabilidad en comparación con la versión anterior, donde estas facilidades no estaban disponibles.



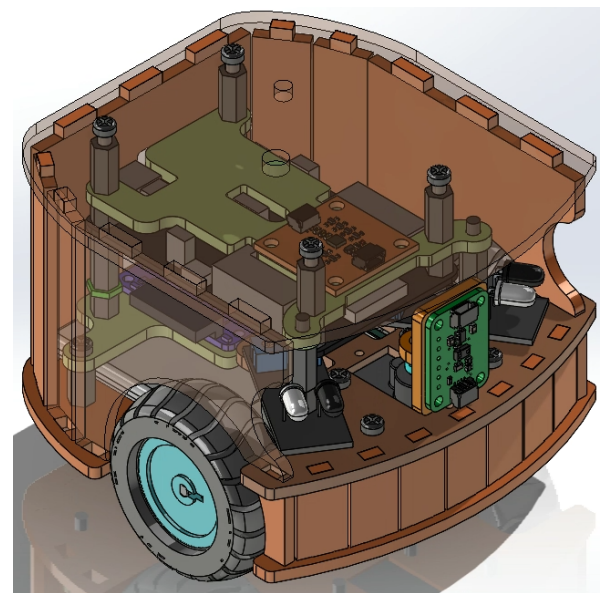
a) Vista frontal



b) Vista lateral



c) Vista superior



d) Vista isométrica

Figura 4.2: Vistas del nuevo Atta-Bot.

Si es necesario realizar tareas de mantenimiento más profundas, los paneles laterales y superior pueden retirarse con facilidad, brindando acceso directo a todos los componentes internos. Esto simplifica operaciones como el reemplazo de cables, la sustitución de la batería o cualquier ajuste necesario, mejorando la eficiencia del mantenimiento como la facilidad de actualización del robot.

La figura 4.3 muestra el diagrama de conexiones eléctricas para el nuevo Atta-Bot, detallando las conexiones físicas necesarias para su correcto funcionamiento. Este diagrama incluye la disposición de cada componente clave, como el microprocesador, sensores, actuadores, entre otros.

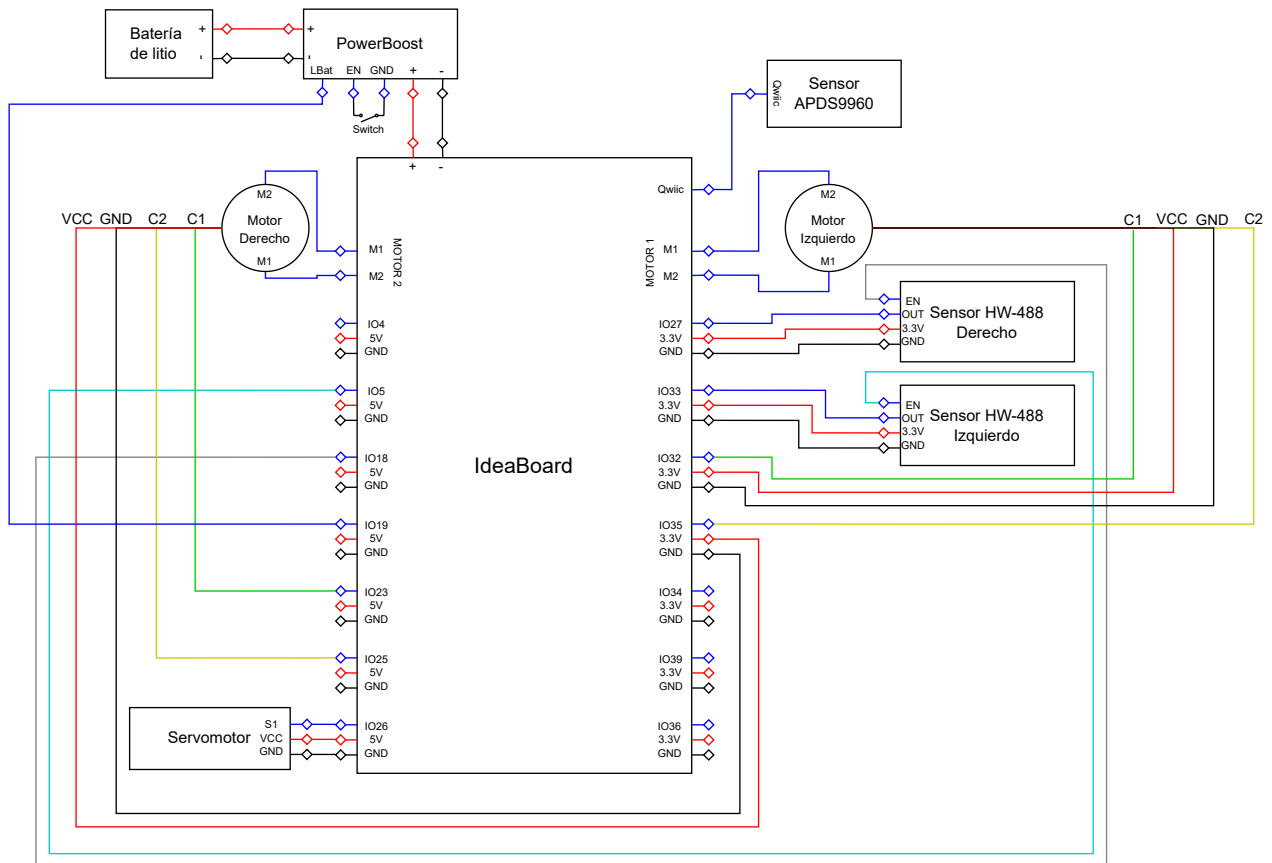


Figura 4.3: Diagrama eléctrico del nuevo Atta-Bot.

4.3 Detección de obstáculos

Para lograr una detección de obstáculos efectiva, la correcta ubicación de los sensores es fundamental. En este subproblema, como se puede ver en la tabla 3.13, se seleccionó como solución óptima el uso de los sensores HW-488 en combinación con un sensor APDS9960. La disposición de estos sensores se muestra en la figura 4.4. En cada extremo del robot se ubicó un sensor HW-488, que tiene un ángulo de detección efectivo de $\pm 35^\circ$ [7], mientras que el sensor APDS9960 fue montado sobre un servomotor en el centro del frontal del robot.

Esta configuración permite que el APDS9960 pueda detectar obstáculos tanto en el centro como en los extremos del campo visual cuando sea necesario, gracias a la capacidad del servomotor de rotar. El objetivo principal fue maximizar la cobertura de detección para reaccionar ante obstáculos lo antes posible. Con esta disposición, se logra una cobertura frontal teórica de 135° , la cual puede expandirse hasta 180° si es necesario, ajustando el servomotor para cubrir el área adicional en los extremos. Esto asegura una detección rápida y eficiente, minimizando riesgos de colisiones.

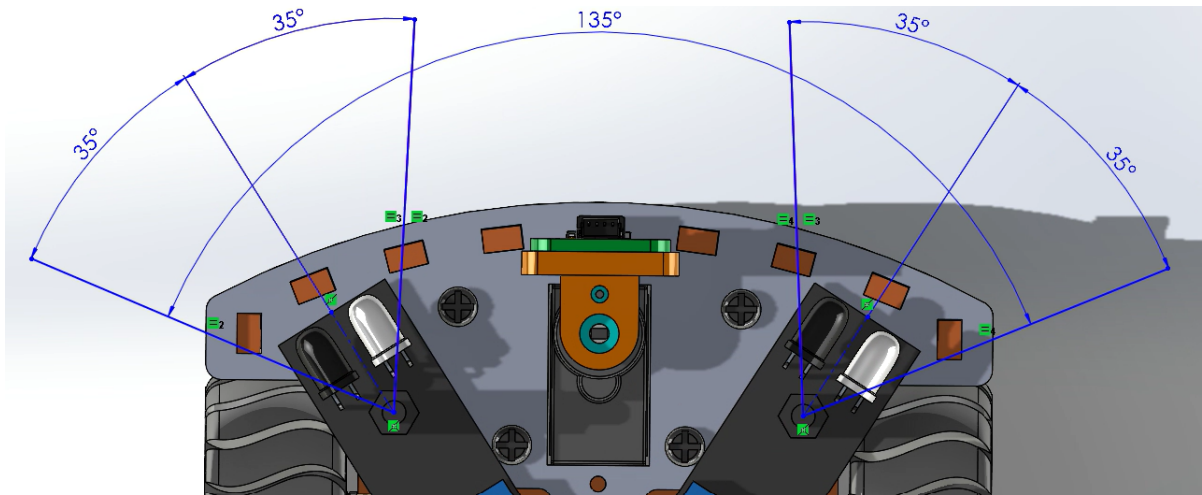


Figura 4.4: Distribución de sensores para detectar obstáculos.

En cuanto a la configuración de los sensores, el sensor APDS9960 no requiere calibración previa a su uso. Solo es necesario conocer su rango de funcionamiento que para detectar obstáculos a una distancia menor a 10 cm, el valor de proximidad del sensor debe ser mayor a 2. Por otro lado, los sensores HW-488 sí requieren una calibración adecuada para su correcto funcionamiento.

La calibración de los HW-488 consiste en ajustar el potenciómetro R6 (figura 4.5) hasta obtener una salida de 38 kHz en la salida del NE555, el cual está conectado al cátodo del LED IR. Para realizar esta calibración de manera precisa, se puede utilizar un osciloscopio para medir la frecuencia en el cátodo del LED IR [7]. Además, es necesario configurar la distancia de detección ajustando el potenciómetro R5 hasta que el sensor detecte objetos a la distancia deseada, que en este caso son objetos con distancias menores a 10 cm. Este proceso de calibración asegura que los sensores funcionen dentro de los parámetros esperados, optimizando su capacidad para detectar obstáculos de forma eficiente.

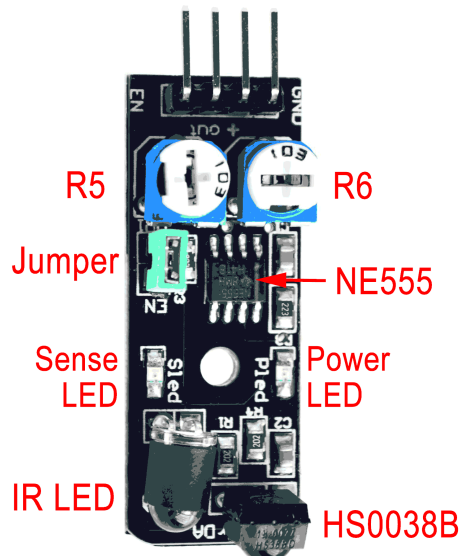


Figura 4.5: Sensor HW-488. Fuente [7]

Con respecto a la programación de los sensores, es importante destacar que estos son propensos a sufrir interferencias, ya sea por la luz ambiental o por otros robots cercanos. Para mitigar estos problemas, se implementaron dos medidas, ya que, de no hacerlo, el sistema de detección fallaría frecuentemente cuando varios robots se encuentran en proximidad.

La primera medida consiste en dividir el tiempo en bloques, para este caso se establecieron bloques de 28,8 ms, los cuales se subdividen en segmentos de no más de 2 ms. Esto es necesario porque si se excede este tiempo, los sensores HW-488 pueden saturar su receptor IR, lo que ocasionaría una pérdida de sensibilidad y en consecuencia, una detección incorrecta [7]. Con este esquema, cada robot selecciona de forma aleatoria dos de estos subbloques: uno para los sensores HW-488 que al estar en lados opuestos no se causan interferencia y otro para el APDS9960. Esto es esencial, ya que ambos tipos de sensores pueden interferirse entre sí. De este modo, los sensores solo estarán activos durante una fracción de tiempo en cada ciclo, lo que reduce significativamente la probabilidad de que dos robots interfieran entre sí. En la figura 4.6, se muestra esta medida de forma gráfica para facilitar su comprensión, donde el intervalo 1 indica cuándo está activado el sensor HW-488 y el intervalo 2 cuándo lo está el sensor APDS9960.

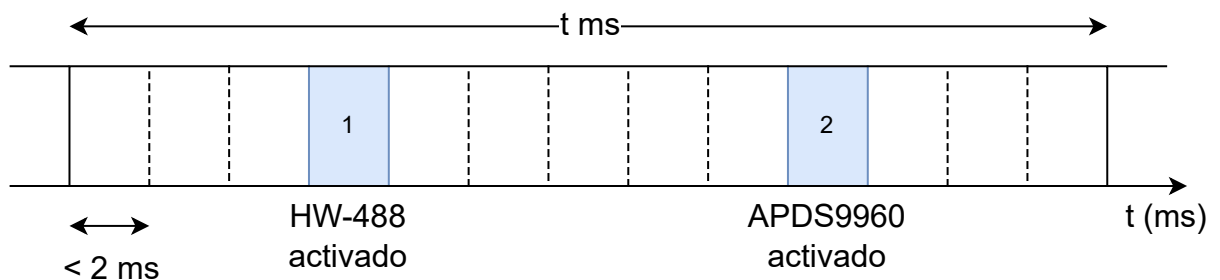


Figura 4.6: Implementación gráfica de la medida 1.

La segunda medida implementada fue la utilización de un "debouncing" por software en cada sensor. Esta técnica consiste en registrar el tiempo cada vez que un sensor se activa, si el tiempo supera un umbral predefinido, se considera que hay un obstáculo real. En caso de que el tiempo se reinicie debido a ruido o inestabilidades, el sistema espera hasta la siguiente ventana de activación para confirmar la detección. En este caso, el margen de tiempo es de 1,35 ms, mientras que la ventana de activación es de 1,6 ms, lo que deja poco margen para errores y ayuda a evitar falsos positivos.

Cabe destacar que estas medidas solo funcionan cuando el robot está en movimiento. Si el robot está detenido, los sensores se apagan para evitar interferir con otros robots en operación. Esto optimiza la eficiencia del sistema y reduce el riesgo de interferencia entre los robots.

4.4 Clasificación de obstáculos

Una vez que los robots son capaces de identificar obstáculos, el siguiente paso es clasificarlos y determinar si el obstáculo es otro robot o no. Para este subproblema, la solución elegida fue utilizar BLE y compartir la pose actual del robot. La idea es que, cuando un robot detecta un obstáculo, escanee su entorno utilizando BLE y envíe su pose actual solo a los robots cercanos. De esta manera, se puede determinar si el obstáculo detectado es otro robot, lo que además mejora la eficiencia, ya que solo se intercambian mensajes entre robots cercanos.

No obstante, aunque esta solución se probó correctamente de forma aislada, durante la integración con el código principal surgió un problema de capacidad de memoria. Las bibliotecas de BLE y las de comunicación ocupaban demasiado espacio, lo que impedía compilar y cargar el código en el microcontrolador. Ante este inconveniente, se decidió utilizar únicamente la parte de envío de la pose actual a todos los robots, eliminando la optimización basada en BLE, ya que esta era vista como una mejora adicional, pero no indispensable para el funcionamiento correcto del sistema.

Con esta aclaración, se procede a explicar la implementación utilizada. Cuando un robot detecta un obstáculo, toma su pose actual y el estado de sus tres sensores de detección. Esta información se envía por broadcast al resto de los robots. El robot luego espera durante un tiempo determinado, si no recibe un mensaje de otro robot confirmando que es el obstáculo, se concluye que el obstáculo no es otro robot.

Si un robot (R1) recibe este mensaje de otro robot (R2) con su pose y estado de sensores, lo primero que hace es calcular la distancia entre ambos robots. Si la distancia es mayor a un umbral predefinido, se descarta que R1 sea el obstáculo de R2. Sin embargo, si la distancia está dentro del umbral, se procede a calcular el ángulo entre ambos robots en relación con la horizontal. Este ángulo se compara con el ángulo de orientación de R2 para determinar si R1 está dentro del área de detección de R2. Los datos de los sensores se utilizan para mejorar la precisión en la identificación del obstáculo. Si R1 confirma que es el obstáculo, envía un mensaje a R2 confirmando esta situación. El código correspondiente a esta implementación puede encontrarse en los anexos, bajo la función **IsRobotObstacle**.

En la figura 4.7 se presenta una representación gráfica de lo que se busca con esta implementación. El proceso comienza cuando el robot R2 detecta un obstáculo y envía su posición al resto de los robots. A continuación, el robot R1 verifica si está dentro de la circunferencia M, definida por un umbral de distancia. Si está dentro de este límite, se procede a calcular el ángulo α , que es el ángulo entre la línea que une a R2 y R1 respecto a la orientación de R2. Con esta información, y basándose en el estado de los sensores, se determina si R1 es efectivamente un obstáculo para R2. Por ejemplo, en la figura representativa, el sensor lateral derecho de R2 debe estar activado obligatoriamente para que se envíe la confirmación de que R1 es el obstáculo. Es importante señalar que esta figura es solo ilustrativa y las distancias mostradas no corresponden a las utilizadas en el prototipo final.

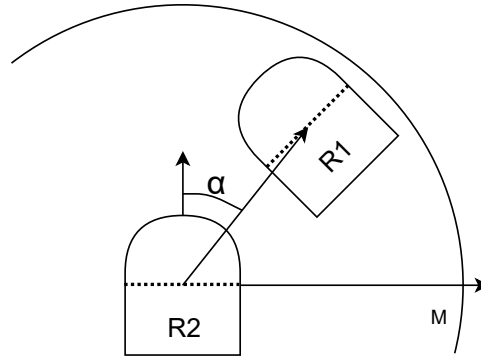


Figura 4.7: Implementación gráfica de algoritmo de clasificación.

Además, cabe destacar que todos los cálculos relacionados con las coordenadas y el ángulo de la pose se realizan en un marco de coordenadas global, compartido por todos los robots del grupo, lo que garantiza que las posiciones y orientaciones sean consistentes entre ellos.

4.5 Control de movimiento

Para resolver el subproblema de mover robot, se optó por utilizar encoders junto con un controlador PID, lo que permite un control preciso y exacto de los movimientos del robot. Por lo que, se emplean encoders de efecto Hall de cuadratura, integrados en cada motor, que ofrecen mayor precisión y la capacidad de indicar la dirección de giro de las ruedas.

La clave para aprovechar los encoders es establecer una relación entre los pulsos generados y la distancia recorrida en el mundo real. Para maximizar la precisión, los encoders se gestionan mediante interrupciones en el código, lo que permite capturar cada pulso sin pérdida de información, evitando así errores en la medición de la distancia recorrida o la velocidad instantánea.

En cuanto a la implementación del controlador PID, se decidió modificar el control utilizado en la versión anterior del Atta-Bot. La versión base consistía en un PID discretizado para mantener una velocidad constante, junto con un observador que vigilaba el desplazamiento. Una vez que el robot recorría la distancia deseada, los motores se apagaban automáticamente [21]. Este enfoque funcionaba bien para movimientos lineales, mientras que para los giros se utilizaba un PID de posición, donde el ángulo deseado era la referencia y el controlador se encargaba de realizar el giro.

En esta nueva versión, se unificó el control PID de velocidad para utilizarlo tanto en movimientos lineales como en giros. Esto se logró al convertir el ángulo de giro en una distancia que cada rueda debe recorrer, utilizando la fórmula de longitud de arco. Básicamente, el ángulo se multiplica por la distancia entre el centro del robot y la rueda. Esta simplificación permitió reducir de dos controladores PID (uno para velocidad y otro para posición) a solo uno, lo que también facilita el mantenimiento del código, ya que la misma función puede manejar ambos tipos de movimientos.

Además, se añadió un mecanismo de reducción progresiva de la velocidad cuando el robot está cerca de completar el movimiento. A medida que se aproxima a la distancia final, la velocidad objetivo comienza a disminuir gradualmente hasta un mínimo, lo que ayuda a contrarrestar los efectos de la inercia. En versiones anteriores, los motores simplemente se apagaban al llegar a la distancia objetivo, lo que provocaba que el robot avanzara más de lo deseado debido a la inercia. Con esta mejora, el robot puede detenerse con mayor exactitud.

En la figura 4.8 se presenta el control utilizado para cada uno de los motores. En este esquema, la letra D representa el desplazamiento requerido para cada rueda. La velocidad objetivo (V_O) se calcula como la diferencia entre la velocidad máxima (V_{max}) y la velocidad de frenado (V_f), lo que asegura una desaceleración progresiva conforme el robot se aproxima a su objetivo. El resto del diagrama sigue la estructura de un controlador PID clásico, se ajustan los parámetros de proporcionalidad, integral y derivada para mantener el control preciso sobre la velocidad de cada rueda y asegurar que el robot alcance su destino de manera eficiente y controlada.

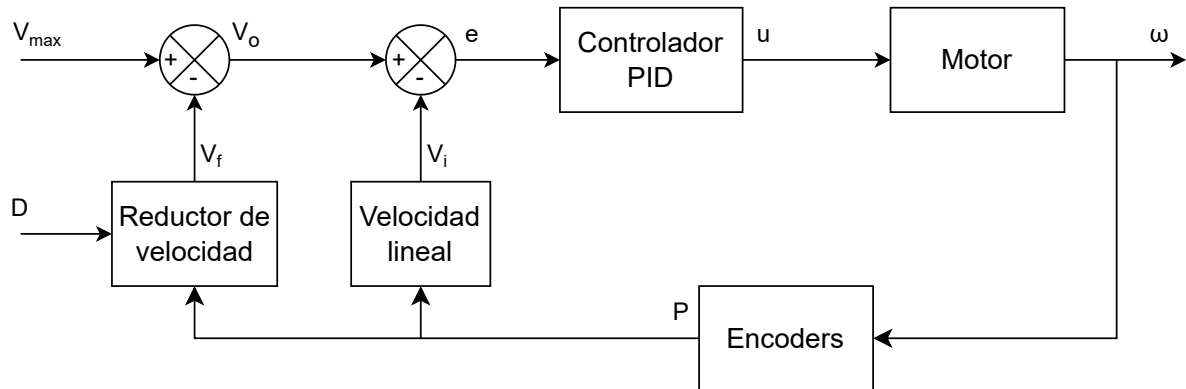


Figura 4.8: Implementación del control PID por motor.

Para la obtención de las constantes del controlador PID, se optó por una sintonización manual. Este proceso consistió en ajustar de manera iterativa los valores de las constantes hasta alcanzar la respuesta deseada. En este caso, el objetivo era lograr una respuesta críticamente amortiguada del sistema, es decir, se busca que el robot alcanzase su velocidad objetivo sin oscilaciones o sobreimpulso. Como resultado de este proceso iterativo se obtuvieron las siguientes constantes.

- K_P : 110
- K_I : 375
- K_D : 2

Es importante mencionar que, para esta implementación del controlador PID, se aumentó la resolución del PWM del microcontrolador de los 8 bits predeterminados a una resolución de 14 bits. Este cambio facilita la sintonización manual del controlador, ya que se pueden usar constantes mas grandes permitiendo un ajuste más fino en el control.

Durante este proceso, se realizaron gráficos del comportamiento dinámico de los motores, lo que permitió visualizar de manera clara la respuesta del sistema. Las gráficas mostraban cómo la velocidad de los motores respondía a las correcciones del PID, permitiendo refinar las constantes hasta obtener una respuesta estable y eficiente, en línea con los requerimientos de precisión y control del robot.

En la figura 4.9, se presenta una de estas gráficas ya con el control PID configurado, esta corresponde a una prueba en la que se programó al robot para avanzar una distancia de 300mm . En esta representación, se puede observar claramente cómo el sistema responde con un comportamiento críticamente amortiguado. Además, de como a medida que el robot se aproxima al desplazamiento deseado, el controlador PID reduce gradualmente la velocidad de los motores, lo que permite frenar de manera controlada hasta detenerse con precisión en la posición objetivo.

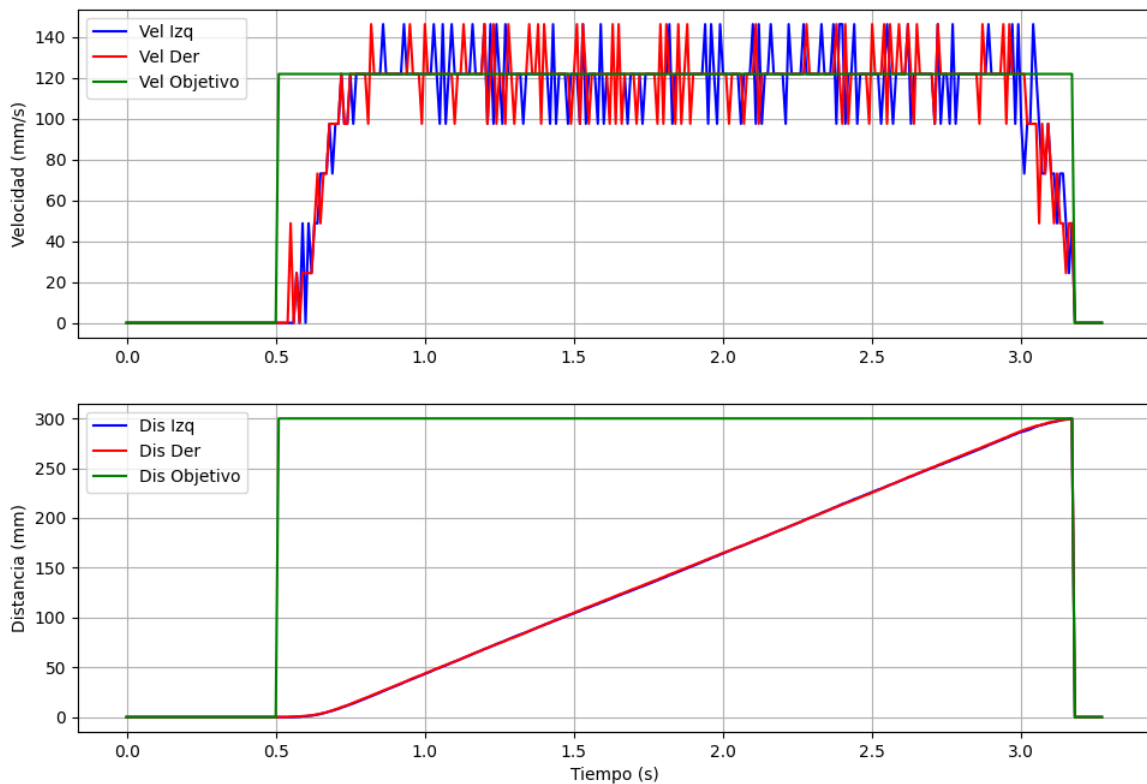


Figura 4.9: Gráfica del control PID en funcionamiento para desplazamiento de 300mm .

La gráfica también evidencia la efectividad del reductor de velocidad implementado, el cual ajusta la velocidad de manera progresiva al acercarse a la meta, minimizando el impacto de la inercia y asegurando que el robot no se desplace más allá de los 300mm programados.

Es importante destacar que los picos de velocidad observados en las gráficas de ambos motores se deben principalmente al tiempo de muestreo seleccionado y a la forma en que se mide la velocidad. Estos picos no reflejan inestabilidades significativas en el sistema, ya que, como se puede apreciar en la gráfica de la distancia, no tienen un impacto notable en el desplazamiento total del robot. Por lo tanto, estos pequeños desajustes en la medición de la velocidad no afectan la exactitud ni el rendimiento del sistema de control.

En la figura 4.9, se observan varios parámetros clave para el control PID, los cuales se resumen en la tabla 4.2. En ella se destaca que el error en estado estable es aproximadamente del 0%, lo cual indica que el control logra alcanzar con exactitud la velocidad deseada de 121,78 mm/s. Aunque se presentan algunos picos, es evidente que la tendencia mantiene la velocidad objetivo, lo que asegura un rendimiento consistente del sistema. Además, el control no muestra sobreimpulso y el tiempo de estabilización es de aproximadamente 220 ms, lo cual resulta adecuado para el rendimiento del robot. En el anexo A se incluyen gráficas adicionales que muestran el desempeño del sistema a diferentes distancias objetivo, brindando una visión más completa de la robustez y efectividad de este control.

Tabla 4.2: Parámetros del control PID.

Parámetro	Resultados
Error estado estacionario	0 %
Tiempo de estabilización	220 ms
Sobreimpulso	0 %

4.6 Gestión de información y comunicación del robot

Para esta sección, como se mencionó anteriormente, se utilizará el entorno de programación Arduino IDE. Tanto el código de la primera versión del Atta-Bot [20] como el del Atta-Bot STEM [24] fueron desarrollados en este ambiente, proporcionando una base sólida sobre la cual empezar.

Para la gestión general del robot, se decidió implementar una máquina de estados finitos (FSM), ya que ambas versiones anteriores utilizaron este enfoque con resultados exitosos, a la vez que utilizar una FSM tiene la ventaja de ser modular, lo que facilita la modificación o adición de nuevos comportamientos y funcionalidades a futuro. Además, se implementó otro programa que corre en paralelo a la FSM, encargado de gestionar la comunicación y la lectura de los sensores. A continuación, se presenta una explicación detallada del funcionamiento de ambos sistemas.

4.6.1 Máquina de estados finitos

En la figura 4.10, se muestra el diagrama de la FSM implementada para el control del robot. Esta FSM consta de nueve estados principales, cada uno diseñado para gestionar una fase específica del comportamiento del robot. La transición entre estos estados está determinada por la lógica interna del sistema, basada en entradas de los sensores, comunicaciones y eventos específicos.

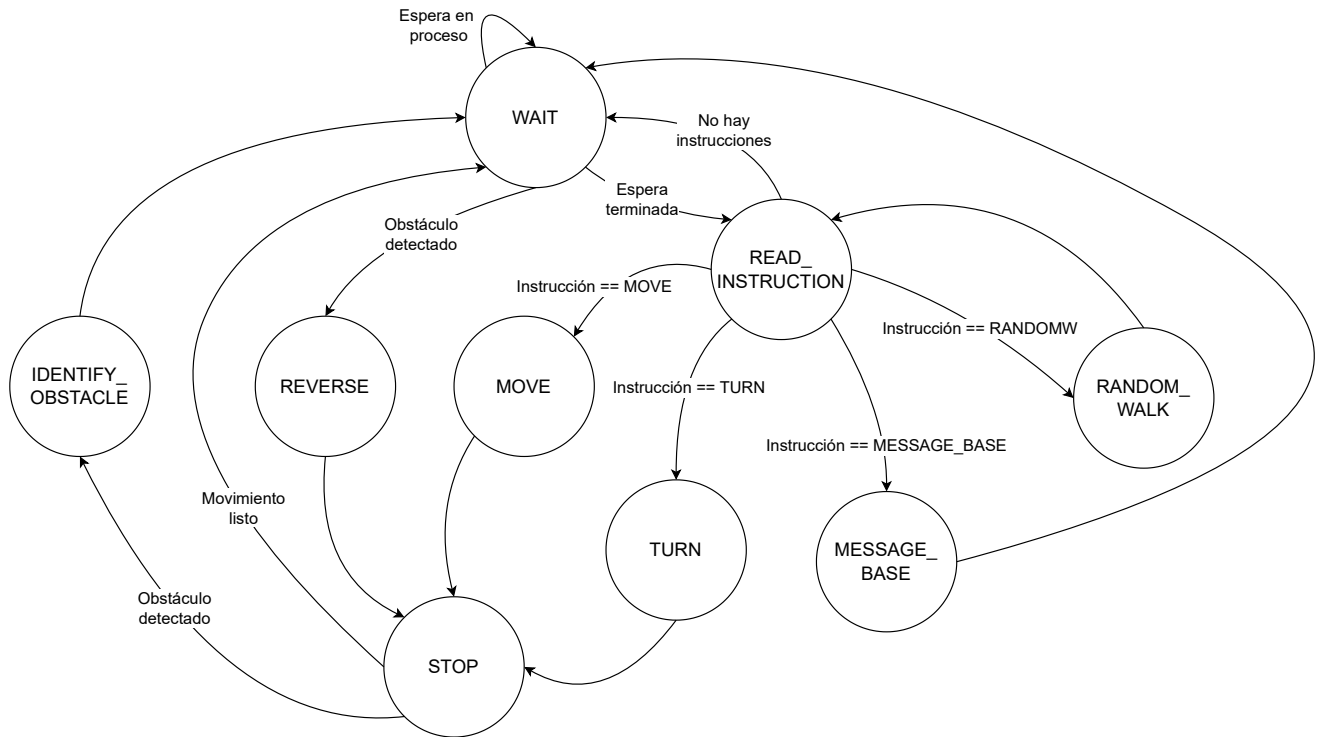


Figura 4.10: FSM implementada para el control del robot.

A continuación, se explicará en detalle cada uno de los estados de la FSM, proporcionando una comprensión completa de su funcionamiento. Esta descripción abarcará no solo las funciones de cada estado, sino también los criterios que deben cumplirse para que el robot transite de un estado a otro, asegurando así un control preciso y coherente del comportamiento global del robot.

WAIT

El estado **WAIT** es el estado inicial de la FSM y está diseñado para que el robot realice pausas sin detener la ejecución completa del programa, evitando el uso de instrucciones como **delay()**. Este estado permite al robot esperar una cantidad específica de milisegundos antes de tomar la siguiente acción. Además, este es el encargado de reiniciar las constantes del PID para que este funcione de manera correcta.

Al finalizar el periodo de espera, la FSM evalúa las condiciones del último movimiento realizado por el robot. Si el movimiento fue exitoso y no se encontró ningún obstáculo, el robot transiciona al estado **READ_INSTRUCTION** para ejecutar nuevas instrucciones. Sin embargo, si durante el último movimiento el robot detectó un obstáculo, se pasa al estado **REVERSE** para retroceder y evitar una colisión.

READ_INSTRUCTION

El estado **READ_INSTRUCTION** se encarga de procesar las instrucciones enviadas al robot, ya sea desde la base o desde otro robot. En función de la instrucción recibida, el robot transicionará a uno de los siguientes estados: **MOVE** (para ejecutar movimientos lineales), **TURN** (para realizar giros), **MESSAGE_BASE** (para enviar mensajes a la base) o **RANDOM_WALK** (para iniciar una exploración aleatoria). Si no hay nuevas instrucciones disponibles para ejecutar, el robot regresará al estado **WAIT** para esperar la llegada de nuevas órdenes.

MOVE

El estado **MOVE** es responsable de ejecutar los movimientos lineales del robot, tanto hacia adelante como hacia atrás. Durante este estado, se aplica el control PID previamente explicado para garantizar desplazamientos precisos. El robot puede salir de este estado de dos maneras: primero, si el robot completa el desplazamiento requerido y segundo, si detecta un obstáculo mediante alguno de sus sensores. En ambos casos, el robot transiciona al estado **STOP** para detener el movimiento.

TURN

El estado **TURN** se encarga de hacer que el robot gire sobre su propio eje en cualquiera de las dos direcciones. Al igual que en el estado **MOVE**, se utiliza el control PID previamente explicado para garantizar la precisión en el giro. Este estado tiene dos condiciones de salida: cuando el robot completa el giro solicitado o si detecta algún obstáculo durante el proceso. En ambos casos, el sistema transiciona al estado **STOP** para detener el movimiento del robot.

STOP

El estado **STOP** es responsable de detener los motores después de un movimiento. Si el movimiento anterior fue exitoso, el sistema transiciona al estado **WAIT**, donde se espera un tiempo antes de ejecutar la siguiente instrucción. Sin embargo, si el movimiento fue interrumpido debido a la detección de un obstáculo, este estado también se encarga de enviar la pose actual del robot al resto del enjambre, permitiendo ejecutar el algoritmo de clasificación anteriormente explicado. Tras este envío de información, el sistema pasa al estado **IDENTIFY_OBSTACLE** para clasificar el obstáculo detectado.

IDENTIFY_OBSTACLE

El estado **IDENTIFY_OBSTACLE** se encarga de clasificar el obstáculo que ha sido detectado. Como se mencionó previamente, el robot espera una confirmación de algún otro robot del enjambre para determinar si el obstáculo es otro robot o un objeto externo. Si en un tiempo determinado no recibe ninguna confirmación, o si recibe una confirmación indicando que el obstáculo es efectivamente otro robot, el sistema procede a pasar al estado **WAIT**, preparándose para hacer un retroceso.

REVERSE

El estado **REVERSE** es responsable de realizar un movimiento de retroceso cuando el robot detecta un obstáculo con alguno de sus sensores frontales. Este movimiento hacia atrás permite que el robot gane suficiente espacio para poder maniobrar y evitar el obstáculo. Una vez que el robot ha completado el desplazamiento hacia atrás, el sistema transiciona nuevamente al estado **STOP**, donde se detienen los motores y se evalúa el siguiente paso a seguir.

MESSAGE_BASE

El estado **MESSAGE_BASE** es responsable de enviar mensajes a la base de control cuando sea necesario. Un ejemplo común es informar a la base cuando el robot ha finalizado la ejecución de una secuencia de movimientos. Tras enviar el mensaje, el robot transiciona al estado **WAIT**, donde permanece a la espera de nuevas instrucciones.

RANDOM_WALK

El estado **RANDOM_WALK** es responsable de ejecutar el comportamiento de enjambre de exploración aleatoria, que fue el unico comportamiento implementado. Su funcionamiento es el siguiente: al ingresar a este estado, el robot genera una instrucción de movimiento aleatoria, donde las opciones están ponderadas, de modo que avanzar tiene una mayor probabilidad que girar. Una vez seleccionado el tipo de movimiento, se realiza una segunda selección aleatoria para determinar la magnitud del desplazamiento o el ángulo de giro, según corresponda.

Si en el último movimiento el robot detectó un obstáculo, el próximo movimiento generado será exclusivamente un giro, para evitar colisiones. Además de generar la instrucción de movimiento, este estado también crea una nueva instrucción que llame a este mismo estado, con el tiempo restante del comportamiento de exploración, asegurando que se generen secuencias de movimientos aleatorios continuos. Una vez generadas ambas instrucciones, el estado **RANDOM_WALK** transiciona al estado **READ_INSTRUCTION** para ejecutar la instrucción recién creada.

4.6.2 Comunicación y Lectura de Sensores

La comunicación y lectura de sensores sigue el flujo descrito en la figura 4.11. En primer lugar, el sistema entra en una etapa de configuración, donde se inicializan todos los elementos necesarios para el correcto funcionamiento del robot, incluyendo los módulos de comunicación. Esta fase garantiza que tanto la comunicación como el control del robot estén listos para operar de manera eficiente.

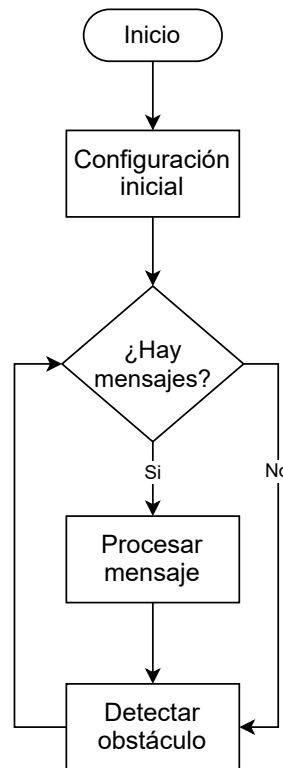


Figura 4.11: Diagrama de flujo .

Una vez completada la configuración, el robot entra en un ciclo continuo (loop) donde primero se verifica si hay mensajes pendientes por procesar. Si se detectan mensajes, estos son analizados y procesados. Después, el sistema procede a la etapa de detección de obstáculos, en la cual se ejecuta el algoritmo de detección descrito previamente en la figura 4.6. Este flujo garantiza una operación continua y eficiente, permitiendo que el robot responda a nuevos comandos y se adapte a su entorno mediante la detección de obstáculos.

Comunicación UDP

Como se muestra en la tabla 3.13, la opción seleccionada para la comunicación tanto con la base como entre los robots fue el uso de WiFi junto con el protocolo UDP. Esta elección se basó en su velocidad y simplicidad, adecuadas para los requerimientos de comunicación en tiempo real. Para implementar este protocolo, se utilizó la biblioteca de WiFiUdp incluida en el entorno de desarrollo Arduino IDE, lo que facilitó considerablemente la integración y programación.

A continuación, se presenta un listado de los comandos actualmente soportados por el robot, los cuales permiten su control. Es importante destacar que los elementos dentro de los paréntesis representan los parámetros ajustables de cada comando, los cuales pueden variar según el tipo de comando.

- *CONFIG|START|(IP_BROADCAST)*: Este comando inicia la configuración del robot. El parámetro *IP_BROADCAST* corresponde a la dirección IP que permite al robot enviar mensajes al resto de los elementos en la red, además de almacenar la IP de la base en memoria.
- *CONFIG|SAVE|(ROBOT_ID)*: Este comando guarda en memoria la información de qué dirección IP corresponde a cada *ROBOT_ID*, lo que permite identificar y gestionar las comunicaciones entre robots.
- *CONFIG|ROBOTS*: Este comando se utiliza principalmente para debug. Permite verificar qué robots están guardados en la memoria del robot ejecutando el comando, solo funciona si el modo debug serial está activado.
- *CONFIG|DEBUG|(NIVEL)*: Activa o desactiva el modo debug por UDP. Con *NIVEL = 0*, el modo debug está apagado, *NIVEL = 1* envía a la base mensajes sobre la finalización de movimientos y detección de obstáculos, *NIVEL = 2* envía datos detallados del funcionamiento interno del robot, como la velocidad y el estado del control PID, *NIVEL = 3* activa una señal visual mediante el LED del robot cuando detecta obstáculos con los sensores frontales.
- *CONFIG|(ROBOT_ID)*: Configura el ID único del robot durante la prueba.
- *RESET*: Reinicia el robot en caso de ser necesario. Este comando es útil para reiniciar el sistema sin necesidad de apagar físicamente el robot, y restablece todos los valores a su estado inicial.
- *SERVO|(ANGULO)*: Controla el servo motor frontal, moviéndolo al ángulo especificado (*ANGULO*). Este comando es utilizado cuando se necesita manipular o ajustar la posición del sensor frontal.
- *PID|(K_P)|(K_I)|(K_D)*: Reescribe las constantes del controlador PID del robot (*K_P*, *K_I*, *K_D*), ajustando los parámetros del control de movimiento. Estas constantes se pierden al reiniciar el robot, por lo que deben reconfigurarse si es necesario mantener estos ajustes específicos.
- *KFPID|(R)|(H)|(Q)*: Ajusta las constantes del filtro de Kalman que se ha implementado dentro del control PID para evitar cambios bruscos en la salida PWM. Estas constantes se pierden tras un reinicio y deben ser reconfiguradas en caso de ser necesarias.
- *POSE|(X)|(Y)|(ANGULO)*: Reescribe la pose actual del robot, actualizando las coordenadas *X*, *Y* y el ángulo de orientación respecto al sistema de coordenadas global.

- *CHECK_OBSTACLE*|(SENSORES)|(X)|(Y)|(ANGULO): Se envía cuando el robot detecta un obstáculo. Incluye la información de los sensores que realizaron la detección, junto con la pose actual del robot ($X, Y, ANGULO$).
- *OBSTACLE_DETECTED*|(ROBOT_ID): Este comando se envía cuando un robot confirma que él mismo es el obstáculo detectado por otro robot. El mensaje incluye su propio *ROBOT_ID*, permitiendo que el robot que detectó el obstáculo reconozca que se trata de otro robot dentro del enjambre y no de un objeto.
- *MOVE*|(DIST): Ordena al robot moverse una distancia específica (*DIST*). Si *DIST* es positivo, el robot avanza y si es negativo retrocede.
- *TURN*|(ANGULO): Indica al robot girar un ángulo determinado (*ANGULO*). Un valor positivo de *ANGULO* representa un giro en sentido horario, mientras que un valor negativo indica un giro en sentido antihorario.
- *WAIT*|(TIEMPO): Ordena al robot detenerse y esperar por un periodo de tiempo específico (*TIEMPO*). Durante este tiempo, el robot no ejecutará movimientos ni procesará nuevas instrucciones.
- *RANDOMW*|(TIEMPO): Activa el modo de exploración aleatoria (random walk) durante un tiempo determinado (*TIEMPO*). En este estado, el robot genera movimientos aleatorios mientras evita obstáculos y colabora con otros robots para optimizar la exploración del entorno.
- *MESSAGE_BASE*|(NUMERO): Este comando envía un mensaje a la base. Si *NUMERO* = 1, el robot enviará el mensaje **READY** indicando que ha completado una tarea o está listo para nuevas instrucciones.

Es importante destacar que entre estos dos sistemas, la FSM que gestiona los movimientos del robot y el sistema encargado de la comunicación, existe una estructura de datos clave llamada **deque**. Una **deque** es una estructura que permite agregar o quitar elementos al principio o al final de una lista, algo así como una cola de doble extremo. Esto la hace muy flexible para ciertos tipos de tareas. Aunque funciona de forma similar a un arreglo, los elementos no están siempre uno al lado del otro en memoria, lo cual permite que sea más eficiente en algunas situaciones. Es especialmente útil cuando necesitas añadir o quitar cosas al inicio y al final rápidamente [25].

Esta estructura actúa como un puente entre ambos sistemas, permitiendo almacenar las instrucciones recibidas a través del protocolo UDP y transferirlas de manera eficiente a la FSM para su procesamiento. Gracias a la **deque**, se asegura que las instrucciones se manejen en el orden correcto y que no se pierdan, incluso si llegan de manera asincrónica. Esto garantiza una interacción fluida entre la comunicación externa y la ejecución de las acciones del robot, mejorando la capacidad del sistema para responder rápidamente a las órdenes y mantener una operación continua.

4.7 Diseño de la base

Como se observa en la figura 3.3, el diseño de la base se divide en tres subproblemas principales: la obtención de las posiciones de los robots, la interpretación de las señales de control y el envío de información a los robots. Para la implementación de este subsistema se ha utilizado el lenguaje de programación Python, ya que, como se mencionó previamente, la obtención de las posiciones de los robots se basa en un trabajo previo desarrollado por el grupo de investigación [8], el cual fue implementado en Python. Esta decisión facilita la integración de la solución actual con esa base de código existente.

Para la comunicación con los robots, se ha empleado el protocolo UDP sobre WiFi, como se explicó en la sección anterior. En la base, se implementa este tipo de comunicación mediante sockets, aprovechando bibliotecas que facilitan la creación y gestión de esta conexión. Las instrucciones utilizadas y sus funciones están detalladas en la sección previa, lo que permite una interacción fluida entre la base y los robots.

4.7.1 Obtención de posiciones de los robots

La obtención de las posiciones de los robots es un componente clave del sistema, ya que permite monitorizar en tiempo real la ubicación y orientación de cada robot dentro del entorno. Para lograr esto, se implementó un sistema de visión que facilita la detección precisa de cada robot. Este sistema utiliza identificadores visuales, como el que se muestra en la figura 4.12, que consisten en dos círculos uno dentro de otro de diferentes colores. Estos colores permiten diferenciar a los robots entre sí y calcular su orientación en el espacio [8].

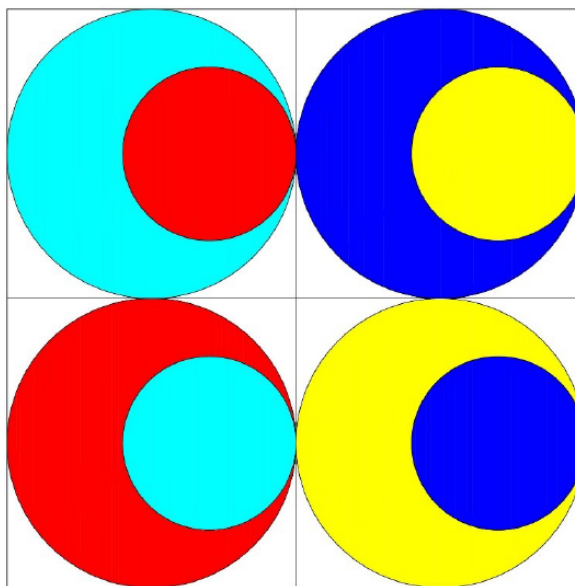


Figura 4.12: Identificador utilizado por los robots en diferentes colores. Fuente [8]

En esta ocasión, se seleccionaron colores como azul oscuro, amarillo y magenta, ya que se distinguen claramente entre sí y con el fondo del escenario, lo que facilita el procesamiento y análisis por parte del sistema de visión.

El sistema de visión se basa en el diagrama de flujo mostrado en la figura 4.13, que describe el proceso de captura de fotogramas de video, aplicando diferentes filtros para generar máscaras con cada uno de los colores utilizados en los identificadores. A partir de estas máscaras, se detectan los círculos correspondientes a los identificadores, lo que permite identificar a los robots y calcular su pose (posición y orientación) [8].

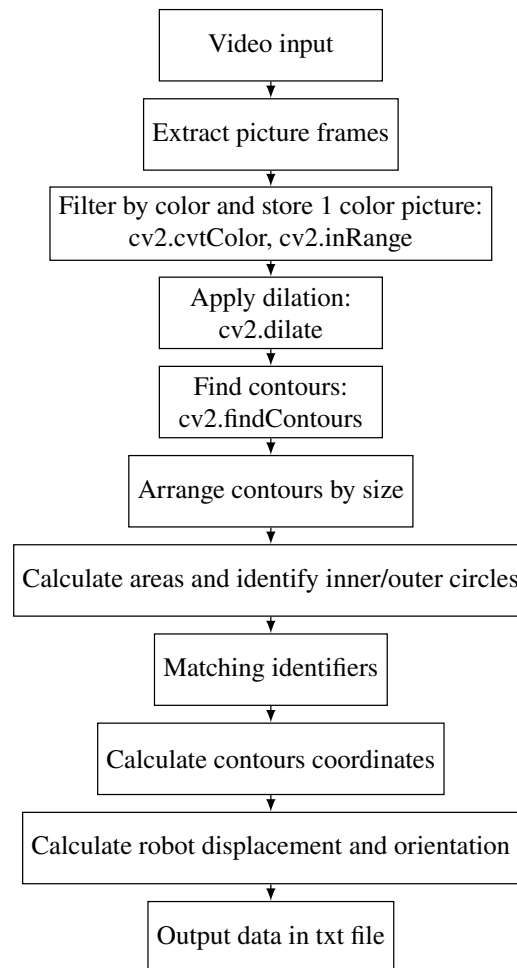


Figura 4.13: Diagrama de flujo del sistema de visión. Fuente [8]

Para este proyecto, fue necesario optimizar el sistema para que funcionara en tiempo real, ya que procesar las máscaras secuencialmente consume un tiempo considerable. Por este motivo, se implementaron optimizaciones para que las máscaras se procesaran en paralelo, lo que redujo significativamente el tiempo de procesamiento y permitió la operación eficiente del sistema.

4.7.2 Funcionamiento de la base

Una vez que las partes de la base están definidas, se integran en un sistema unificado. En la figura 4.14, se presenta un diagrama de flujo de alto nivel del sistema. El primer paso es que la base recolecte las IPs de todos los robots en la red, enviando un mensaje de broadcast a todos los elementos conectados. Aquellos que respondan serán considerados como robots.

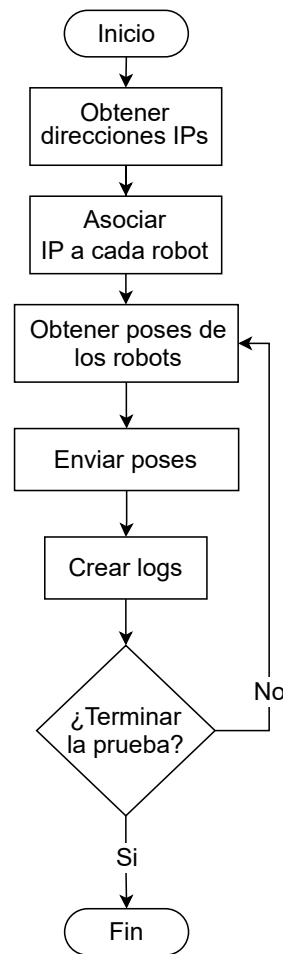


Figura 4.14: Diagrama de flujo de la base.

Posteriormente, se asigna una IP a cada robot detectado. Esto es necesario porque todos los robots comparten la misma programación y no conocen de antemano su identificador ni los colores que los distinguen. La asignación se realiza enviando una instrucción de giro (*TURN*) a un solo robot y utilizando el sistema de visión para identificar qué robot realizó el giro. Una vez identificado, se configura su ID con el comando *CONFIG|ROBOT_ID*. Este procedimiento se repite para cada robot en la prueba.

Luego, el sistema entra en un ciclo principal donde primero se obtienen las poses de los robots utilizando el sistema de visión previamente explicado. A continuación, se envía a cada robot su pose correspondiente y esta información se almacena en un archivo log. Se mantiene un registro de las posiciones de los robots en un archivo tipo CSV y además, se crea un video que guarda todos los fotogramas procesados para facilitar el análisis del sistema a posterior.

Es importante destacar que, de manera paralela a este flujo, se ejecutan otros dos flujos. Uno se encarga de mantener la comunicación abierta con los robots sin depender del procesamiento de imágenes, lo que permite enviar instrucciones a los robots en cualquier momento. El otro flujo se dedica exclusivamente a leer los mensajes enviados por los robots y a registrar estos mensajes en un log.

Capítulo 5

Resultados y Análisis

Una vez que se ha desarrollado un prototipo funcional, es necesario validar las soluciones implementadas. En este capítulo se exponen tanto los resultados obtenidos como su respectivo análisis, comparándolos con las métricas definidas en la tabla 3.8. Además, se concluye con un análisis económico para determinar la viabilidad financiera del proyecto, evaluando tanto los costos de producción como los posibles ahorros o beneficios derivados de la implementación del sistema.

5.1 Validación de la exactitud del robot

Para validar la exactitud del robots, se llevarán a cabo dos experimentos: uno para medir la exactitud de los movimientos en línea recta y otro para evaluar la exactitud de los giros.

5.1.1 Exactitud en movimientos lineales

En la prueba de movimientos en línea recta, el robot será colocado sobre una superficie plana que permita marcar su posición inicial con un lápiz o marcador. Se enviará el comando de movimiento *MOVE* desde la base, ordenando al robot avanzar una distancia específica. Una vez completado el movimiento, se marcará la posición final del robot. Posteriormente, se medirá la distancia recorrida entre los puntos inicial y final, comparando esta distancia con la esperada según el comando enviado. A partir de esta comparación, se calculará el error relativo de cada prueba.

Esta prueba se repetirá un total de 30 veces para cada uno de los dos desplazamientos de distancia definidos. La repetición de 30 mediciones por desplazamiento tiene como objetivo generar una distribución normal de los datos, permitiendo un análisis estadístico más preciso y confiable de la exactitud del sistema. Esto también facilitará la identificación de cualquier desviación sistemática o aleatoria en los movimientos del robot.

La figura 5.1 muestra una imagen capturada durante el proceso de validación de la prueba de desplazamiento.

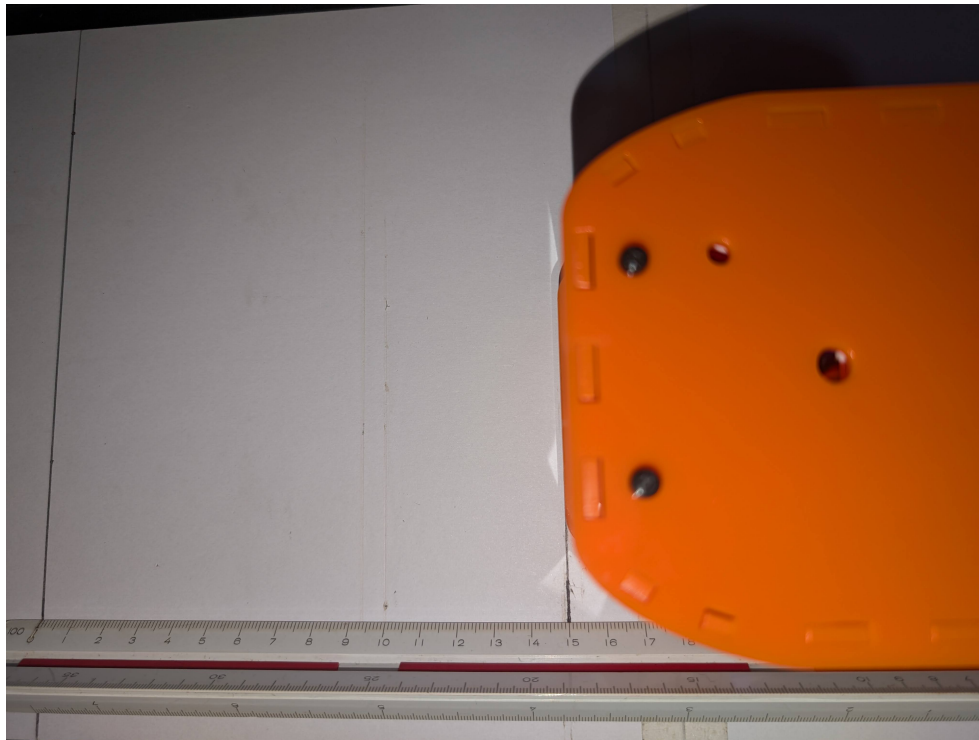


Figura 5.1: Imagen del proceso de validación de la exactitud en movimientos lineales.

La tabla 5.1 muestra los resultados de la prueba de desplazamientos, mientras que la figura 5.2 presenta un diagrama de cajas y bigotes para el error de desplazamiento en ambas pruebas. Como se puede observar en ambos elementos, los errores promedio entre los diferentes desplazamientos son similares entre si, así como la desviación estándar de las mediciones. Este comportamiento consistente se debe en gran parte a la implementación eficaz del sistema de frenado en el control del robot, lo que reduce significativamente los efectos de la inercia. Gracias a esto, los desplazamientos no solo son exactos, sino también repetibles a lo largo de múltiples pruebas.

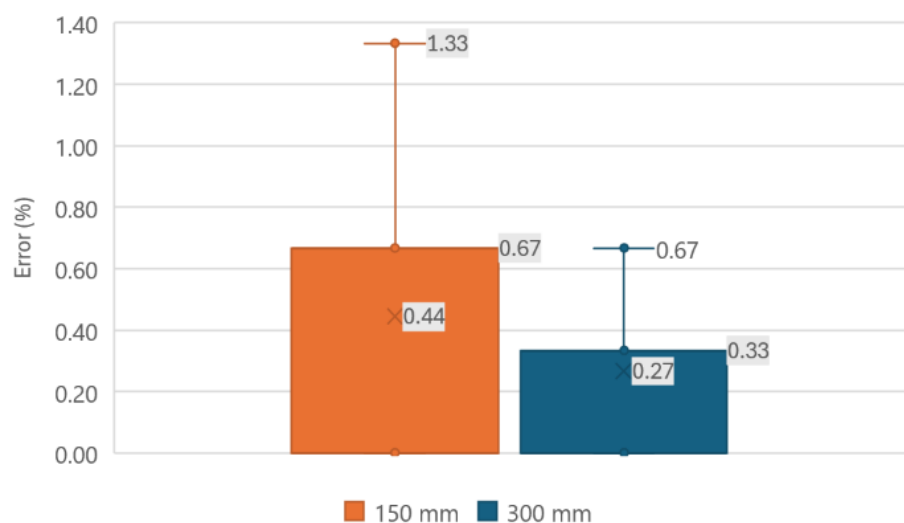


Figura 5.2: Diagrama de cajas y bigotes para el error de prueba de desplazamiento.

Tabla 5.1: Resultados de las pruebas de desplazamiento.

Medición	Desplazamiento 150 (± 1 mm)	Error (%)	Desplazamiento 300 (± 1 mm)	Error (%)
1	151	0,67	299	0,33
2	152	1,33	301	0,33
3	151	0,67	300	0,00
4	151	0,67	300	0,00
5	150	0,00	302	0,67
6	151	0,67	300	0,00
7	150	0,00	300	0,00
8	152	1,33	301	0,33
9	150	0,00	301	0,33
10	150	0,00	300	0,00
11	151	0,67	298	0,67
12	150	0,00	301	0,33
13	150	0,00	299	0,33
14	150	0,00	300	0,00
15	150	0,00	301	0,33
16	151	0,67	301	0,33
17	151	0,67	300	0,00
18	149	0,67	301	0,33
19	149	0,67	300	0,00
20	150	0,00	299	0,33
21	151	0,67	299	0,33
22	149	0,67	298	0,67
23	151	0,67	300	0,00
24	150	0,00	302	0,67
25	149	0,67	299	0,33
26	150	0,00	300	0,00
27	149	0,67	301	0,33
28	149	0,67	301	0,33
29	151	0,67	300	0,00
30	150	0,00	298	0,67
Promedio	150,27	0,44	300,07	0,27
Desviación estándar	0,85	0,40	1,06	0,23

Además, el bajo promedio de error también está relacionado con una correcta caracterización de los componentes del robot. La precisión en la medición del tamaño de las llantas y la adecuada configuración de la cantidad de pulsos de los encoders han sido fundamentales para alcanzar estos resultados, asegurando que las mediciones de distancia estén alineadas con las expectativas teóricas.

Con base en los resultados obtenidos, se seleccionó la prueba de desplazamiento de 300 mm, que presentó la mayor desviación estándar de 1,06 mm, para el cálculo de la potencia de prueba. Utilizando esta desviación estándar, se determinó una potencia estadística del 99,8%, lo cual supera ampliamente el nivel ideal del 85%. Esto indica que el tamaño de muestra de 30 mediciones es suficiente y estadísticamente representativo para validar los resultados con alta confiabilidad.

Por otro lado, los errores promedios registrados están dentro del valor ideal, que de acuerdo con la tabla 3.8 debe ser menor al 2,74%. Este resultado confirma que el sistema de control es robusto y que el robot cumple con la métrica número 3, que corresponde a la precisión en el desplazamiento.

Por lo tanto, se puede decir que el sistema implementado asegura un nivel adecuado de exactitud en los movimientos lineales del robot, manteniéndose dentro de los márgenes aceptables definidos para el proyecto.

5.1.2 Exactitud en giros

Para la prueba de giros, se seguirá un procedimiento similar al utilizado en los movimientos en línea recta. El robot se colocará en una posición inicial marcada sobre una superficie plana. Mediante el comando *TURN*, se le ordenará girar un ángulo específico. Una vez que el robot complete el giro, se marcará su nueva orientación final, es decir, la dirección en la que quedó alineado tras la rotación. Posteriormente, se medirá el ángulo realmente girado comparándolo con el ángulo enviado en el comando, para así calcular el error angular. Al igual que en la prueba de distancias, este experimento se repetirá un total de 30 veces para cada uno de los cuatro giros de ángulos definidos. Estos ángulos incluyen tanto giros positivos (en sentido de las agujas del reloj) como negativos (en el sentido contrario a las agujas del reloj), lo que permitirá evaluar la exactitud del robot en ambas direcciones de rotación.

La repetición de 30 mediciones por ángulo asegura que se obtenga una distribución normal de los datos, permitiendo un análisis estadístico preciso y la evaluación del margen de error en los giros del robot. Este enfoque permitirá identificar patrones de desviación en la respuesta del robot a los comandos de rotación y posibles diferencias entre giros positivos y negativos.

La figura 5.3 muestra una imagen capturada durante el proceso de validación de la prueba de desplazamiento.

La tabla 5.2 muestra los resultados de la prueba de giros y la figura 5.4 presenta un diagrama de cajas y bigotes que ilustra el error de giro en todas las pruebas realizadas. Los resultados muestran una consistencia similar a la observada en la prueba de desplazamientos lineales. Esto es lógico, ya que ambos tipos de movimientos emplean el mismo sistema de control, el cual incluye un mecanismo de frenado que mitiga los efectos de la inercia. Este mecanismo también tiene un impacto positivo en los movimientos de giro, proporcionando precisión y estabilidad. Además, la dirección de giro, ya sea en sentido horario o antihorario, no parece afectar ni la precisión ni la exactitud de los giros.

El bajo promedio de error también está relacionado con una correcta caracterización de las dimensiones físicas del robot. La medición precisa de la distancia entre el centro del robot y las llantas ha sido fundamental para alcanzar estos resultados, asegurando que las mediciones de los giros estén alineadas con las expectativas teóricas.

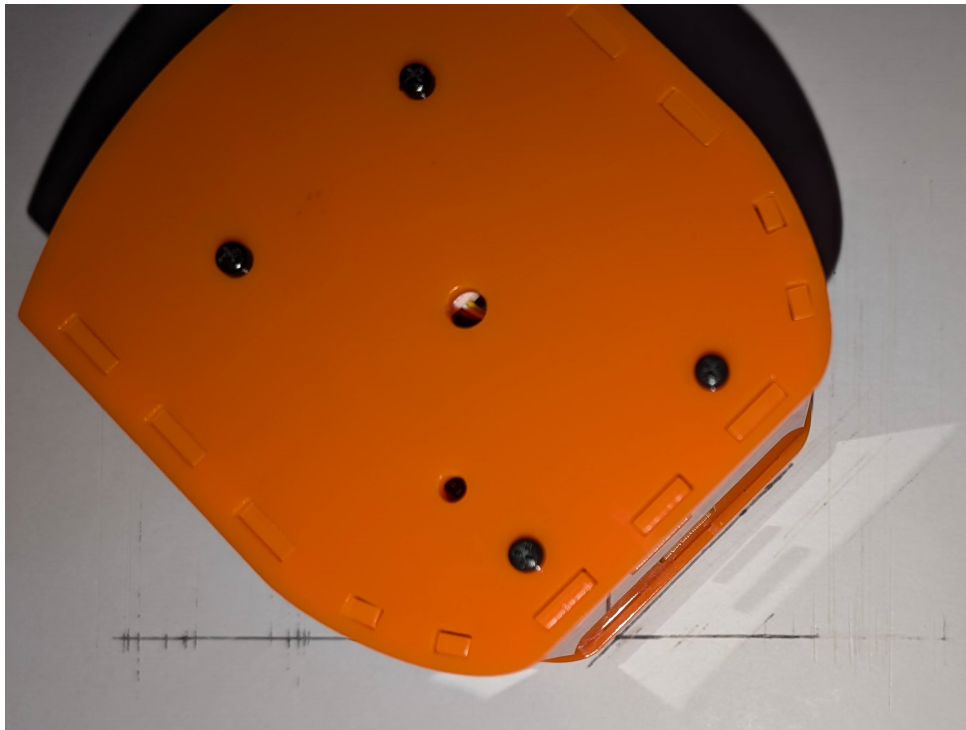


Figura 5.3: Imagen del proceso de validación de la exactitud en giros.

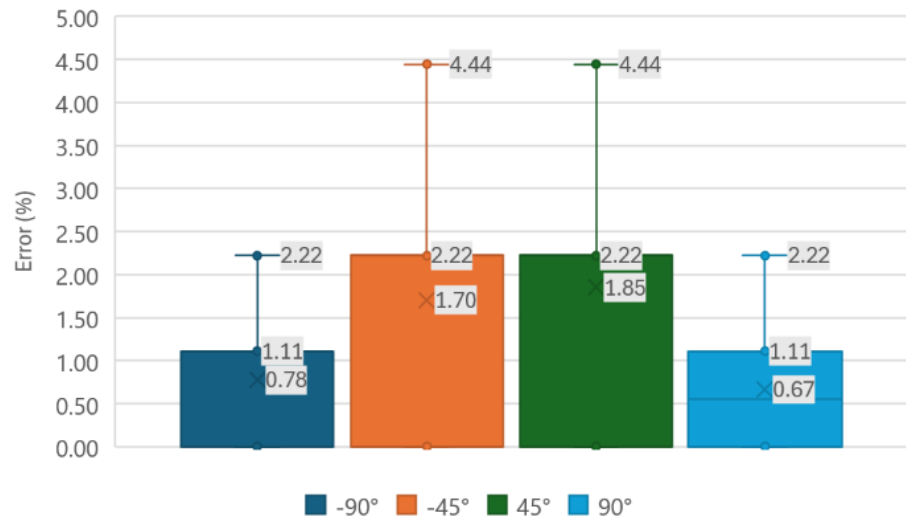


Figura 5.4: Diagrama de cajas y bigotes para el error de prueba de giro.

Sin embargo, como se observa en la tabla 5.2 y en especial en la figura 5.4, los giros con ángulos pequeños (45°) presentan un error más significativo en comparación con los giros más grandes (90°). Esto se debe principalmente a las limitaciones del instrumento de medición utilizado, que solo permitía medir diferencias de ángulo con una precisión de 1° . Por lo tanto, aunque las variaciones en los ángulos medidos oscilan entre 1 y 3 grados, en los giros más pequeños este margen de error tiene un mayor impacto relativo. Esta diferencia es observable también en las desviaciones estándar de las mediciones, que son similares para ambos tipos de giros, pero que afectan más a los ángulos pequeños debido a su menor valor nominal.

Tabla 5.2: Resultados de las pruebas de giro.

Medición	Giro -90 ($\pm 1^\circ$)	Error (%)	Giro -45 ($\pm 1^\circ$)	Error (%)	Giro 45 ($\pm 1^\circ$)	Error (%)	Giro 90 ($\pm 1^\circ$)	Error (%)
1	-89	1,11	-45	0,00	43	4,44	89	1,11
2	-89	1,11	-44	2,22	44	2,22	90	0,00
3	-88	2,22	-43	4,44	44	2,22	88	2,22
4	-89	1,11	-45	0,00	44	2,22	89	1,11
5	-88	2,22	-45	0,00	44	2,22	89	1,11
6	-90	0,00	-45	0,00	45	0,00	90	0,00
7	-89	1,11	-44	2,22	43	4,44	89	1,11
8	-89	1,11	-44	2,22	44	2,22	90	0,00
9	-90	0,00	-45	0,00	44	2,22	90	0,00
10	-89	1,11	-43	4,44	45	0,00	90	0,00
11	-89	1,11	-44	2,22	44	2,22	90	0,00
12	-90	0,00	-44	2,22	43	4,44	89	1,11
13	-90	0,00	-44	2,22	44	2,22	89	1,11
14	-90	0,00	-43	4,44	45	0,00	89	1,11
15	-90	0,00	-44	2,22	45	0,00	89	1,11
16	-89	1,11	-45	0,00	45	0,00	88	2,22
17	-90	0,00	-44	2,22	44	2,22	90	0,00
18	-88	2,22	-45	0,00	44	2,22	90	0,00
19	-90	0,00	-44	2,22	44	2,22	90	0,00
20	-89	1,11	-45	0,00	45	0,00	89	1,11
21	-89	1,11	-44	2,22	44	2,22	89	1,11
22	-90	0,00	-44	2,22	44	2,22	90	0,00
23	-90	0,00	-44	2,22	45	0,00	90	0,00
24	-90	0,00	-45	0,00	45	0,00	90	0,00
25	-89	1,11	-43	4,44	45	0,00	89	1,11
26	-90	0,00	-45	0,00	44	2,22	88	2,22
27	-89	1,11	-45	0,00	44	2,22	90	0,00
28	-89	1,11	-44	2,22	44	2,22	90	0,00
29	-88	2,22	-45	0,00	43	4,44	89	1,11
30	-90	0,00	-43	4,44	44	2,22	90	0,00
Promedio	-89,30	0,78	-44,23	1,70	44,17	1,85	89,40	0,67
Desviación estándar	0,69	0,77	0,72	1,59	0,64	1,42	0,66	0,74

A partir de los resultados obtenidos, se seleccionó la prueba de giro de -45° , que presentó la desviación estándar más alta, de $1,59^\circ$, para calcular la potencia de la prueba. Con esta desviación estándar, se obtuvo una potencia estadística del 91,39%, superando el nivel ideal del 85%. Esto sugiere que el tamaño de muestra de 30 mediciones es adecuado y estadísticamente representativo, proporcionando una validación confiable de los resultados.

Según la tabla 3.8, el valor ideal del error para los giros debería ser menor al 1,58%. En el caso de los giros de 90° , esta métrica se cumple satisfactoriamente, ya que los errores se mantienen por debajo de ese umbral. Sin embargo, para los giros de 45° , aunque el error está cerca de cumplir con la métrica, no lo logra completamente, alcanzando un máximo de 1,85% en el peor de los casos. A pesar de esto, si calculamos un promedio general del error en todos los giros evaluados, obtenemos un valor de 1,25%, que sí se encuentra dentro del valor ideal especificado en las métricas.

Por lo tanto, se puede decir que el sistema de control y la caracterización del robot permiten que, en promedio, los giros cumplan con los requerimientos de exactitud establecidos en la métrica número 4, aunque se debe tener en cuenta la ligera variación en los ángulos más pequeños debido a las limitaciones del instrumento de medición utilizado para medir los giros en esta prueba.

5.2 Validación de la duración de la batería del robot

Para medir la duración de la batería del robot, se realiza una única prueba diseñada para simular condiciones de uso real. En esta prueba, primeramente se cargan los tres robots disponibles al máximo de batería, luego estos se colocan juntos en el campo de pruebas (figura 90). Los robots se programan para ejecutar el comportamiento de exploración aleatoria de manera indefinida, con el fin de observar su desempeño continuo. Durante la prueba, se registrará el tiempo que cada robot logra operar antes de que indique que la batería está baja, esto lo realiza prendiendo el led con el que cuenta el IdeaBoard. Esta prueba se repetirá un total de dos veces, lo que permitirá obtener seis muestras en total. De este modo, se podrá analizar la consistencia en la duración de la batería entre los diferentes ensayos y obtener una estimación más precisa del rendimiento energético del robot.

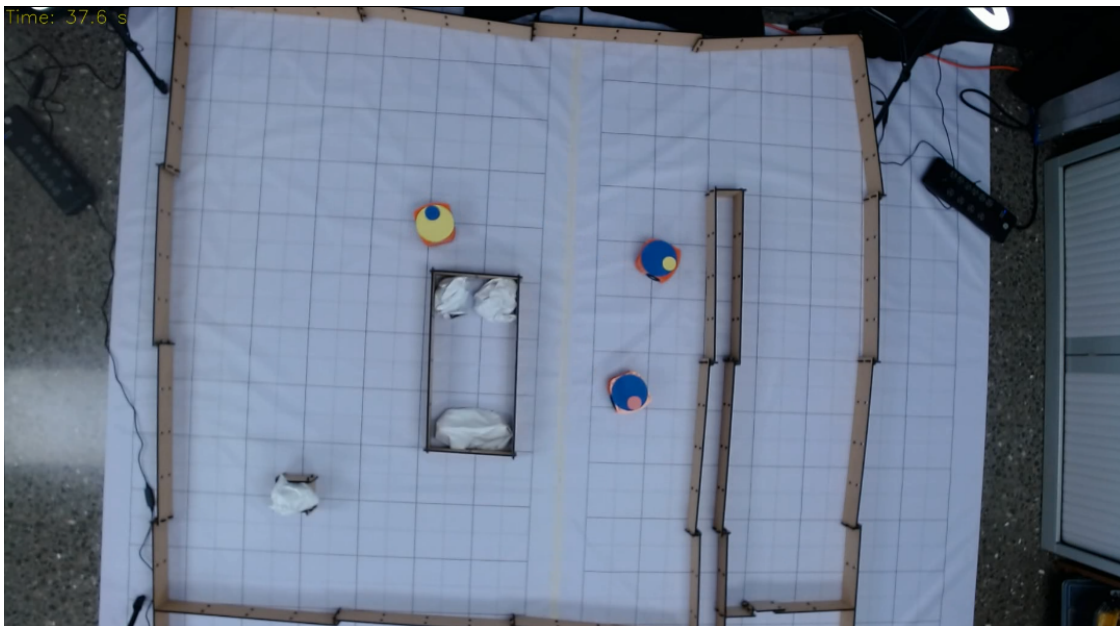


Figura 5.5: Campo de pruebas utilizado para pruebas de validación.

En la tabla 5.3 se presentan los resultados de la prueba de duración de la batería. Como se puede observar, la duración promedio de la batería fue de aproximadamente 289 minutos, con una desviación estándar de 21 minutos. A partir de estos resultados se calculó la potencia estadística de la prueba, se obtuvo una potencia estadística del 59,19%, que se queda por debajo del nivel ideal del 85%, por lo que si bien el número de datos recolectados no es suficiente para establecer un comportamiento completamente representativo del robot, se debe considerar que cada conjunto de mediciones requiere alrededor de 9 horas, distribuidas entre 4 horas de

carga y más de 5 horas de funcionamiento continuo. Debido a estas limitaciones de tiempo, no fue posible realizar más de dos pruebas completas.

Tabla 5.3: Resultados de la duración de la batería.

Medición	Duración (min)
1	271
2	283
3	305
4	326
5	294
6	260
Promedio	289,83
Desviación estándar	21,78

A pesar de esto, es importante destacar que, de acuerdo con la tabla 3.8, el valor ideal de duración de la batería debe ser superior a 90 minutos. Dado que el valor promedio obtenido es más de tres veces este umbral (289 minutos), se puede decir que la métrica 5 relacionada con la duración de la batería fue alcanzada de manera satisfactoria, superando ampliamente el valor ideal establecido.

5.3 Validación del sistema de comunicación

Para validar el sistema de comunicación, se realizarán dos pruebas fundamentales. La primera prueba se enfocará en medir la tasa de éxito en la transmisión de mensajes desde la base hacia los robots. La segunda prueba evaluará la tasa de éxito en la transmisión de mensajes entre los robots. Estas dos pruebas se consideran suficientes debido a que la transmisión de mensajes desde los robots hacia la base no se considera crucial para la funcionalidad principal del sistema, sino que está destinada principalmente para tareas de debug y monitoreo.

5.3.1 Comunicación base a robots

Esta prueba evaluará la eficiencia y confiabilidad del sistema en la transmisión de mensajes desde la base hasta los robots. Para ello, se utilizarán los tres robots disponibles en el campo de pruebas mostrado en la figura 5.5, donde cada robot ejecutará el comportamiento de exploración aleatoria.

El procedimiento consistirá en enviar 1000 mensajes desde la base a cada robot mientras estos se encuentran en exploración. Este número elevado de mensajes permitirá obtener una muestra representativa del desempeño del sistema y evaluar su comportamiento en condiciones prolongadas de operación. Esta cantidad es suficiente para detectar no solo fallos esporádicos, sino también cualquier tendencia recurrente o problema de rendimiento

a largo plazo. Al finalizar el envío de los 1000 mensajes, se contabilizará la cantidad de mensajes recibidos por cada robot, generando así un conjunto de resultados para el análisis.

La prueba se repetirá tres veces para asegurar la consistencia y la replicabilidad de los resultados, minimizando la influencia de cualquier variación provocada por factores externos. Al finalizar, se contará con un total de nueve mediciones. Además, la prueba se ejecuta mientras los robots exploran aleatoriamente para simular un entorno realista, replicando las condiciones de una operación de enjambre donde los robots están en movimiento constante, explorando y comunicándose de forma continua.

Para facilitar el conteo de mensajes y obtener resultados precisos, se han implementado comandos especiales en el sistema que registran la recepción de mensajes por parte de cada robot.

En la tabla 5.4 se presentan los resultados obtenidos en esta prueba, donde se observa una tasa promedio de éxito en la recepción de mensajes por parte de los robots de un 99,92%. Este resultado supera lo establecido en la métrica número 6, que exige un porcentaje de éxito superior al 96,9%, indicando que la comunicación de la base a los robots cumple con los requisitos de confiabilidad en la comunicación.

Tabla 5.4: Resultados obtenidos en la prueba de comunicación base a robots.

Medición	Mensajes recibidos	Tasa de éxito (%)
1	1000	100,00
2	999	99,90
3	1000	100,00
4	998	99,80
5	1000	100,00
6	999	99,90
7	1000	100,00
8	1000	100,00
9	997	99,70
Promedio	999,22	99,92

Al calcular la potencia de la muestra a partir de los resultados obtenidos, se alcanza un valor del 100%, lo que demuestra que los datos son completamente representativos del sistema. Esta potencia máxima indica una fiabilidad estadística alta, asegurando que los resultados reflejan con precisión el rendimiento y comportamiento del sistema en pleno funcionamiento.

5.3.2 Comunicación entre robots

Esta prueba tiene como objetivo evaluar la eficiencia y confiabilidad en la transmisión de mensajes entre los robots. Al igual que en la prueba de comunicación entre la base y los robots, se utilizarán los tres robots disponibles, los cuales ejecutarán el comportamiento de exploración aleatoria en el área definida en la figura 5.5.

El procedimiento consiste en que, durante la elección de cada nuevo movimiento, cada robot enviará un mensaje a los otros dos, esto un total de 500 mensajes enviados a cada robot. Esto genera un total combinado de 1000 mensajes recibidos por cada robot, un volumen que permite obtener una muestra representativa para evaluar el rendimiento del sistema de comunicación en condiciones cercanas a las reales. La elección de 500 mensajes por robot asegura que el sistema sea probado en escenarios de alta demanda, evaluando cómo maneja la carga simultánea de múltiples emisores.

Al igual que en la prueba anterior, esta se repetirá tres veces para garantizar la consistencia de los datos y validar los resultados, eliminando variaciones que puedan surgir por factores externos, lo que permitirá obtener un total de nueve mediciones. Ejecutar esta prueba mientras los robots realizan exploración aleatoria simula un entorno dinámico y proporciona resultados representativos del rendimiento en escenarios de operación reales.

Para facilitar el conteo de mensajes y obtener resultados precisos, se han implementado comandos especiales en el sistema que registran la recepción de mensajes por parte de cada robot, además del código que envía estos mensajes cada vez que el robot elige una instrucción.

La tabla 5.5 muestra los resultados de la prueba, con una tasa promedio de éxito en la recepción de mensajes entre robots de un 99.24%. Este porcentaje supera el mínimo del 96.9% establecido en la métrica número 6, lo cual confirma que el sistema de comunicación entre robots satisface los requisitos de confiabilidad del sistema.

Tabla 5.5: Resultados obtenidos en la prueba de comunicación entre robots.

Medición	Mensajes recibidos	Tasa de éxito (%)
1	992	99,20
2	994	99,40
3	987	98,70
4	993	99,60
5	996	99,60
6	987	98,70
7	996	99,60
8	995	99,50
9	992	99,20
Promedio	992,44	99,24

Al realizar el cálculo de la potencia de la muestra basada en los datos obtenidos, se obtuvo una potencia del 100%. Esto confirma que los datos recolectados representan completamente el sistema, evidenciando una alta confiabilidad estadística. Este nivel de potencia asegura que los resultados reflejan de manera precisa el rendimiento y comportamiento del sistema en condiciones de operación.

Por lo tanto, se puede decir que en esta prueba como en la anterior, el sistema de comunicación cumple con la métrica número 6. A pesar de que el protocolo UDP no garantiza la entrega de paquetes, los resultados demuestran que este no representa un problema en la implementación actual.

5.4 Validación de la satisfacción del usuario

Para validar la satisfacción del usuario al utilizar el sistema de enjambre, se decidió aplicar un cuestionario a un grupo de diez personas, entre los que se encontraban asistentes del laboratorio y estudiantes del ITCR de carreras afines al proyecto. Previo a la aplicación del cuestionario, los participantes recibieron una breve capacitación sobre el uso del sistema para asegurar que comprendieran su funcionamiento básico y pudieran evaluarlo de manera objetiva.

La creación del formulario se basó en la norma **ISO 9241-11:2018**, que se refiere a la usabilidad de los sistemas. Según esta norma, la usabilidad se define como el grado de eficacia, eficiencia y satisfacción con el que determinados usuarios alcanzan sus objetivos en contextos específicos [26]. La figura 5.6 muestra las características y atributos utilizados para medir la usabilidad según esta norma. Estos conceptos sirvieron de guía en la elaboración del cuestionario.

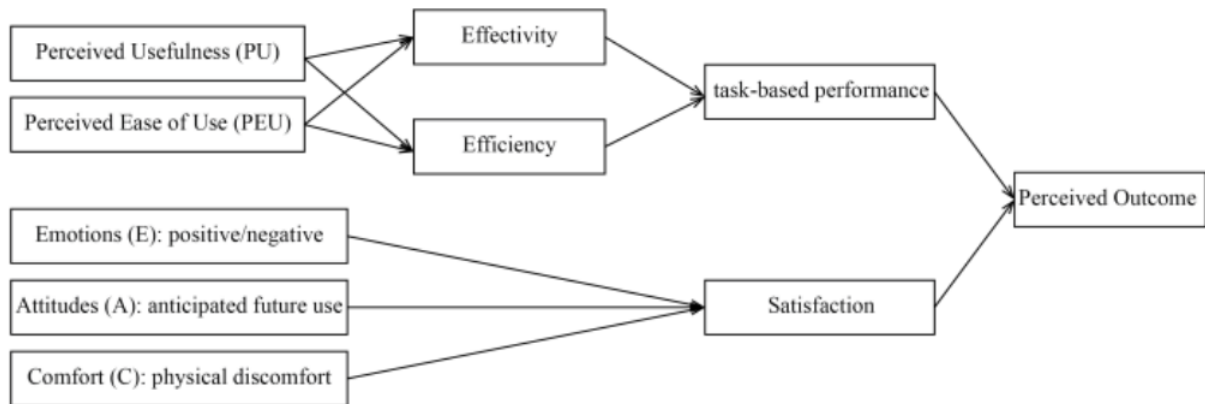


Figura 5.6: Criterios y variables nominales de usabilidad. Fuente [9].

Además de utilizar la ISO 9241-11:2018 como referencia, se consulto cuestionarios ya validados que miden estas características y atributos [27], los cuales fueron adaptados para este caso específico. En la tabla 5.6 se presentan las afirmaciones utilizadas en el cuestionario, junto con el atributo y la característica asociada que se pretende medir.

Tabla 5.6: Cuestionario utilizado para medir satisfacción del usuario.

Característica	Atributo	Afirmaciones
Eficiencia y efectividad	PU: Utilidad percibida	El sistema de enjambre es útil.
	PEU: Facilidad de uso percibida	El sistema de enjambre es fácil de usar.
		Fue fácil aprender a usar el sistema de enjambre.
Satisfacción	E: Emociones	Me siento cómodo utilizando el sistema.
	A: Actitudes	Usaría este sistema en tareas futuras.

Estas afirmaciones deberán ser evaluadas en base a una escala Likert de 1 a 5, donde 1 representa "completamente en desacuerdo" y 5 representa "completamente de acuerdo".

La tabla 5.7 muestra los resultados de la evaluación de los cuatro atributos descritos en la tabla 5.6. En general, las calificaciones promedio de los cuatro atributos se encuentran en la mitad superior de la escala, con una puntuación promedio igual o superior a 4,5. Los atributos con calificaciones promedio más bajas fueron "actitudes" y "utilidad percibida," siendo "actitudes" el que presentó mayor variación. En contraste, los atributos mejor evaluados fueron "facilidad de uso percibida" y "emociones."

Tabla 5.7: Resultados de la validación de la satisfacción del usuario.

Atributo	Promedio	Desviación estándar
PU: Utilidad percibida	4,6	0,49
PEU: Facilidad de uso percibida	4,7	0,46
E: Emociones	4,8	0,40
A: Actitudes	4,5	0,67

A partir de los resultados obtenidos, se empleó el atributo de actitudes para calcular la potencia de la muestra, ya que este presentó la desviación estándar más elevada (0.67). La potencia de la muestra resultante fue del 69.56%, lo cual sugiere que habrían sido necesarios aproximadamente 5 encuestados adicionales para alcanzar el 85% recomendado. Sin embargo, debido a la limitada disponibilidad de participantes, no fue posible lograr dicha cifra.

Los comentarios de los encuestados resaltan aspectos positivos del sistema y algunas sugerencias para futuras mejoras. Los usuarios consideran que el diseño de los robots es atractivo y su tamaño adecuado y destacan la fluidez en la comunicación entre el programa principal y los robots. También valoran que el sistema de enjambre es más eficiente en comparación con versiones anteriores, mostrando movimientos más rápidos y un control mejorado, además de una notable mejora en la detección de obstáculos. Además, entre las sugerencias, se menciona que sería útil agregar un menú para facilitar el uso de comandos sin necesidad de memorizarlos.

En relación con la métrica número 8 en la tabla 3.8, que establece un valor ideal de satisfacción del usuario igual o superior a 4, se observa que los cuatro atributos evaluados cumplen con este valor. Esto indica que la métrica se cumple en su totalidad.

5.5 Validación del tiempo de procesamiento de un fotograma

Para validar el tiempo de procesamiento de un fotograma, se llevará a cabo una prueba en la que los tres robots disponibles ejecutarán el comportamiento de exploración aleatoria durante un período de 5 minutos en el campo de pruebas de la figura 5.5. Durante la prueba, se registrará en un log el tiempo que tarda en procesarse cada fotograma capturado por el sistema de visión. Estos tiempos de procesamiento se almacenarán para su posterior análisis.

Al finalizar la prueba, se procederá a analizar los tiempos registrados para evaluar la consistencia y eficiencia del sistema en la gestión de los fotogramas en tiempo real. Esta prueba se repetirá un total de tres veces para obtener un conjunto de datos más robusto y así poder identificar posibles variaciones o patrones en el procesamiento de imágenes bajo diferentes condiciones.

La figura 5.7 muestra los resultados obtenidos en las tres pruebas realizadas, representados mediante diagramas de cajas y bigotes para cada una. En total, se midió el tiempo de procesamiento de 2764 fotogramas: 927 para la primera prueba, 935 para la segunda y 902 para la tercera, en ese mismo orden se presentan en la figura.

Se observa que los promedios de tiempo de procesamiento entre las pruebas son bastante cercanos, el tiempo promedio mínimo registrado fue de 50,5 ms, mientras que el máximo promedio fue de 51,96 ms. Además, aunque se identificaron valores atípicos en las tres pruebas, ninguno superó los 136,9 ms. Esto confirma que se cumple con la métrica número 1 en su valor ideal, ya que, según la tabla 3.8, el tiempo ideal de procesamiento debe ser menor a 200 ms.

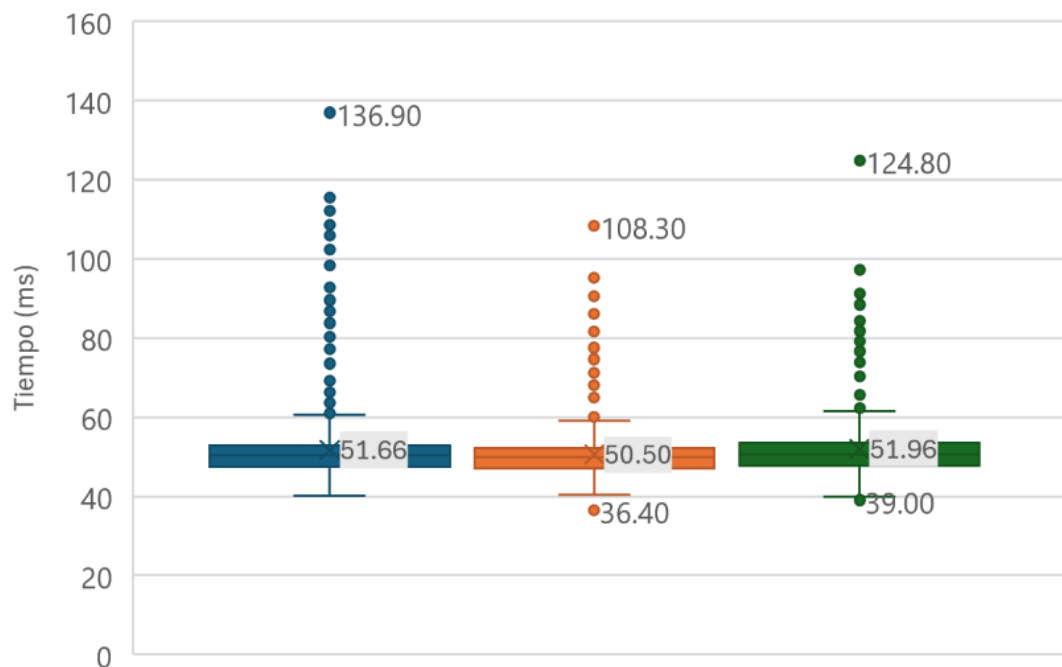


Figura 5.7: Diagrama de cajas y bigotes para las tres pruebas realizadas de tiempo de procesamiento de un fotograma.

Al analizar todos los tiempos de prueba en conjunto, se obtuvo una desviación estándar de 8.33 ms. Basándose en este valor, se calculó una potencia estadística del 100%, lo cual supera ampliamente el estándar ideal del 85%. Este alto nivel de potencia indica que los resultados obtenidos son altamente representativos y confiables, brindando evidencia estadística sólida de la consistencia del sistema bajo condiciones de uso real.

5.6 Validación de la identificación y clasificación de obstáculos

Para validar la identificación y clasificación de obstáculos, se llevará a cabo una prueba en el área de pruebas representada en la figura 5.5. Se colocarán los tres robots disponibles en el campo y se enviará un mensaje por broadcast con el comando ‘*CONFIG|DEBUG|1*’ para activar el modo de debug en los robots a través de UDP. A continuación, se iniciará una prueba en la que los robots ejecutarán el comportamiento de exploración aleatoria durante 5 minutos.

Una vez finalizada la prueba, se procederá a revisar el log generado por la consola junto con el video grabado durante la misma. Cada interacción entre los robots y los obstáculos será clasificada en una de las cuatro categorías descritas en la tabla 5.8. Estas categorías reflejan las posibles situaciones de detección y clasificación de obstáculos. Como se indica en la tabla, la única opción que se considera exitosa es cuando el sistema logra detectar y clasificar correctamente el obstáculo. Esta clasificación es clave para evaluar tanto la precisión como la consistencia del sistema en su capacidad para identificar y distinguir los obstáculos en tiempo real. La prueba se repetirá dos veces para garantizar que el comportamiento de los robots sea consistente, proporcionando así un conjunto de datos más robusto y representativo del rendimiento general del sistema de detección.

Tabla 5.8: Clasificación de la detección y clasificación de obstáculos.

Posibles Situaciones	Puntuación
No detectó obstáculo	0
Detectó obstáculo, pero fue un falso positivo	0
Detectó obstáculo, pero no lo clasificó correctamente	0
Detectó y clasificó correctamente el obstáculo	1

En las tablas 5.9 y 5.10 se presentan los resultados de esta prueba. En ellas, se observa que el total de situaciones analizadas entre ambas pruebas fue de aproximadamente 267. La primera prueba alcanzó una tasa de éxito en la detección y clasificación de un 97,46%, mientras que la segunda obtuvo un 97,18%. Comparando estos valores con la métrica objetivo (mayor a 95%) especificada en la tabla 3.8, se puede afirmar que se cumple satisfactoriamente, ya que ambas pruebas superan el porcentaje mínimo de éxito.

Al analizar los resultados de ambas pruebas, se obtiene una potencia de muestra del 87,04%, lo cual supera el umbral habitual del 85%. Esto indica que los datos recopilados son representativos del funcionamiento del sistema de detección de obstáculos en condiciones de operación real.

Tabla 5.9: Resultados de la primera prueba de detección de obstáculos.

Posibles situaciones	Cantidad
No detectó obstáculo	0
Detectó obstáculo, pero fue un falso positivo	1
Detectó obstáculo, pero no lo clasificó correctamente	2
Detectó y clasificó correctamente el obstáculo	115
Total	120
Tasa éxito	97,46%

Tabla 5.10: Resultados de la segunda prueba de detección de obstáculos.

Posibles situaciones	Cantidad
No detectó obstáculo	2
Detectó obstáculo, pero fue un falso positivo	1
Detectó obstáculo, pero no lo clasificó correctamente	1
Detectó y clasificó correctamente el obstáculo	138
Total	147
Tasa éxito	97,18%

Un análisis más detallado de los resultados también sugiere que la estrategia utilizada, ilustrada en la figura 4.6, ha sido altamente efectiva. En situaciones donde la interferencia de los sensores infrarrojos podría provocar falsos positivos en la detección o fallos en la identificación de obstáculos, se observó que aunque estos casos aún pueden ocurrir, su frecuencia es lo suficientemente baja para no impactar significativamente el rendimiento global de los robots. Esto indica que el sistema está gestionando adecuadamente las interferencias, lo cual resulta esencial para la fiabilidad y eficiencia en escenarios de operación real.

En cuanto al sistema de clasificación, se observa que este logra clasificar exitosamente los obstáculos en la mayoría de los casos. Aunque pueden presentarse errores ocasionales en la clasificación, estos no afectan significativamente el comportamiento general del sistema, por lo que no representan un problema crítico. En la mayoría de las situaciones, el sistema demuestra una capacidad adecuada para realizar una clasificación efectiva de los obstáculos, contribuyendo así al buen desempeño del robot en su entorno operativo.

5.7 Validación de los comportamientos de enjambre

Como se mencionó en el capítulo anterior, solo se logró implementar un comportamiento de enjambre, que es el de exploración aleatoria. Para validar este comportamiento, se llevará a cabo una prueba en la que se colocarán los tres robots en el campo de pruebas, mostrado en la figura 5.5. Luego, se les ordenará ejecutar el comportamiento de exploración aleatoria durante un periodo de 5 minutos.

Al finalizar la prueba, se analizará el último fotograma del video grabado y a partir de

este, se calculará el área total de cobertura explorada por los robots en ese tiempo. Este análisis permitirá verificar cuánta área del entorno fue explorada colectivamente por los robots, lo que sirve como medida de la efectividad del comportamiento de enjambre en términos de distribución espacial y eficiencia. La prueba se repetirá un total de 10 veces para obtener una muestra representativa y garantizar la consistencia de los resultados, lo que permitirá identificar patrones en el comportamiento de los robots y evaluar si el algoritmo de exploración es robusto y adecuado para diferentes escenarios.

La figura 5.8 muestra el resultado de la segunda prueba. En la parte superior, se visualiza el video grabado por la cámara, mientras que en la parte inferior, se destaca en color azul el recorrido realizado por los robots durante la prueba. Al observar la trayectoria, se identifican ciertas áreas del campo de pruebas que los robots no lograron alcanzar. Para obtener un cálculo más preciso del área realmente explorada, se aplicó una máscara, como se muestra en la figura 5.9, en la parte inferior de la imagen. Esta máscara permite delimitar de forma efectiva el área accesible en el campo de pruebas, facilitando así una medición más exacta del área explorada y mejorando la precisión de los resultados obtenidos en esta prueba.

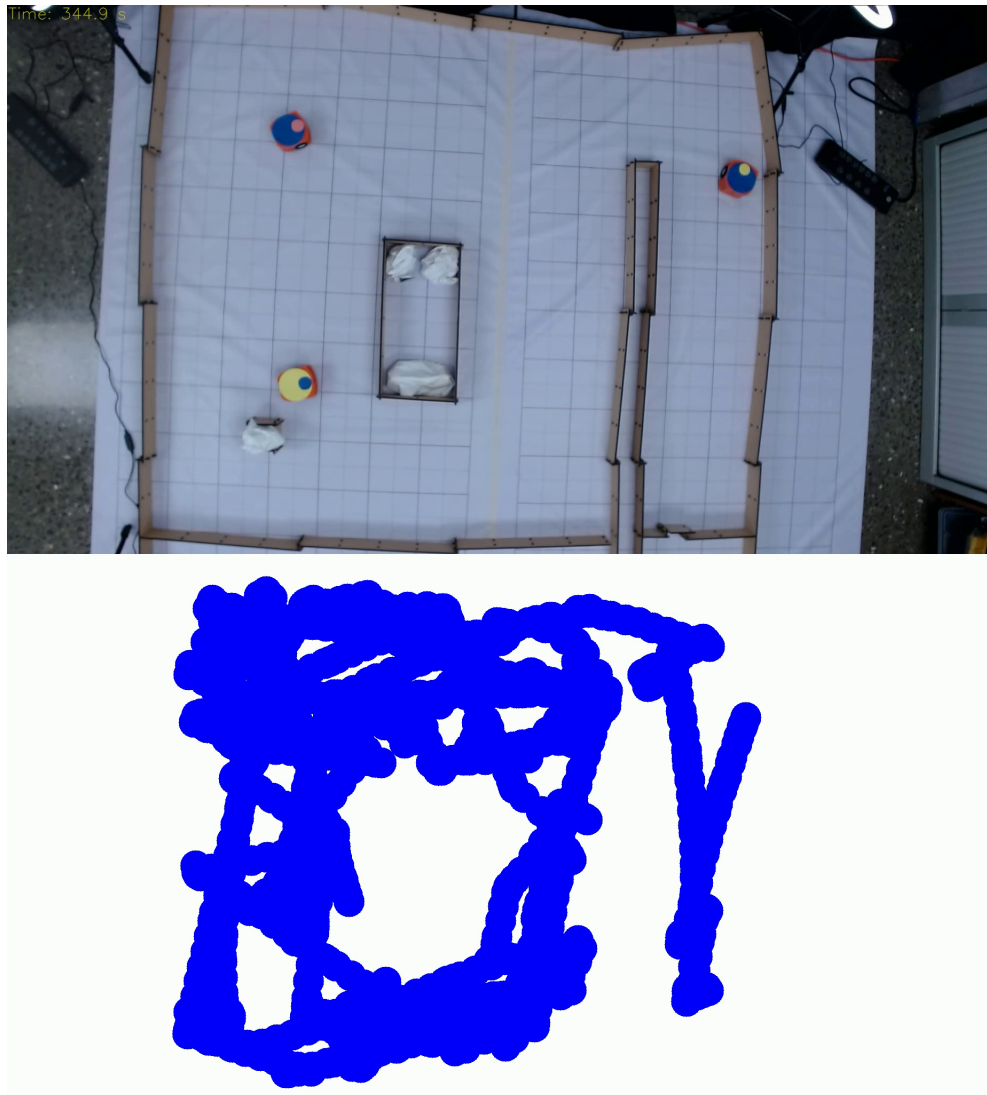


Figura 5.8: Resultados obtenidos en la segunda prueba del comportamiento de enjambre.

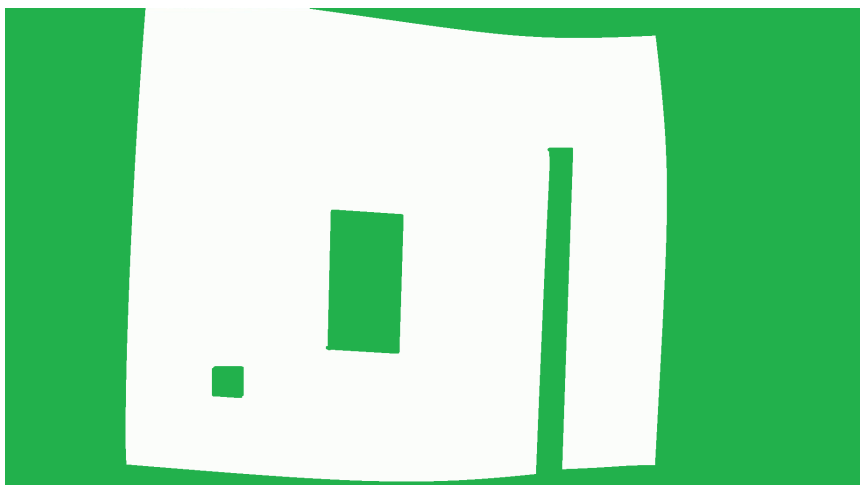


Figura 5.9: Mascara utilizada para medir de forma correcta el área explorada por los robots.

La tabla 5.11 presenta los resultados obtenidos en las diez pruebas realizadas, donde se observa que el área promedio cubierta por los tres robots en un período de 5 minutos es del 49,19%. Al comparar estos resultados con la primera versión del Atta-Bot, que alcanzaba aproximadamente un 50% de cobertura [5], se puede afirmar que la nueva versión del Atta-Bot ha logrado implementar el comportamiento de exploración aleatoria de manera efectiva y consistente, manteniendo un rendimiento similar a su versión anterior.

Tabla 5.11: Resultados del comportamiento de exploración aleatoria.

Prueba	Área explorada (%)
1	54,20
2	60,07
3	43,77
4	49,51
5	50,60
6	49,37
7	45,71
8	48,51
9	44,98
10	45,18
Promedio	49,19
Desviación estándar	4,70

Con base en los resultados obtenidos, se calculó la potencia de la muestra, obteniéndose un valor de 44.49%, considerablemente por debajo del estándar del 85% recomendado para este tipo de pruebas. Se estima que alcanzar ese nivel de potencia habría requerido aproximadamente 30 mediciones en total. Sin embargo, debido a limitaciones de tiempo, fue complicado realizar más de las 10 mediciones actuales. Aun cuando esta muestra no es del todo representativa, su propósito era únicamente observar el comportamiento general del sistema al realizar este comportamiento, sin ningún objetivo específico de precisión cuantitativa.

Por otro lado, al analizar el cumplimiento de la métrica número 9, expuesta en la tabla 3.8, el valor ideal requería la implementación de al menos dos comportamientos. En este caso, solo se logró implementar un comportamiento de manera exitosa, lo cual cumple con el valor marginal propuesto para esta métrica, pero no alcanza el ideal. Esto sugiere que, aunque el sistema está operando correctamente en cuanto a la exploración, queda margen para diversificar el rango de comportamientos que esta nueva versión del Atta-Bot es capaz de ejecutar.

5.8 Análisis económico

El análisis económico de este proyecto se centra en cuantificar y organizar los costos asociados a la adaptación realiza a el robot Atta-Bot STEM para convertirlo en un robot de enjambre, con el fin de proporcionar una visión clara y completa de los recursos empleados. A diferencia de un análisis financiero orientado a evaluar ganancias futuras, este estudio busca detallar los factores económicos involucrados en el desarrollo del proyecto, dado que en proyectos de investigación los beneficios no siempre se reflejan en ingresos monetarios. En primer lugar, se presenta una lista exhaustiva de componentes y sus respectivos precios. Posteriormente, se desglosa el salario asignado al diseñador a lo largo del proyecto, así como el costo de los productos y servicios requeridos, como software y equipos de cómputo. Esto con el propósito de establecer una base económica que facilite la planificación en etapas futuras del proyecto y permita un control preciso de los costos incurridos.

La Tabla 5.12 muestra los costos asociados a la construcción de un robot en esta nueva versión del Atta-Bot [6], con un costo estimado de aproximadamente USD 182,39 por unidad. Es importante señalar que este precio considera que todos los componentes se adquieren localmente, eliminando así gastos adicionales por envío o importación. Además, los componentes marcados con un * en la tabla son suficientes para ensamblar hasta tres robots. Por lo tanto, si se desea construir un total de tres robots, como se hizo en este proyecto, el costo total sería de aproximadamente USD 453,57; lo que implicaría un ahorro de más de USD 90 por cada tres robots ensamblados.

Para la base del sistema, se requieren principalmente una cámara y una computadora. La computadora debe ser de gama media, ya que se necesita cierto poder de cómputo para analizar las posiciones de los robots en tiempo real. Se recomienda una con un procesador Intel i5 de 12.^a generación o un equivalente, equipada con al menos 8 GB de RAM. En cuanto a la cámara, se necesita que tenga una resolución mínima de 1920x1080 para lograr un balance adecuado entre calidad de video y tiempo de procesamiento.

En la Tabla 5.13 se detallan los costos aproximados de los dispositivos utilizados en este proyecto. Como se puede observar, el costo aproximado del sistema base es de USD 670.

En cuanto al salario del investigador, se consideró una tarifa por hora de C\$2400, dedicando medio tiempo al proyecto (22.5 horas por semana) durante un total de 16 semanas, lo que da un costo de C\$864000, aproximadamente USD 1682. En cuanto a los activos, solo se requirió una licencia de SolidWorks, con un costo aproximado de USD 50.

Tabla 5.12: Coste unitario de la nueva versión del robot Atta-Bot.

Componente	Unidades	Precio en CR	Coste total
Motor DC con enconder y rueda N20	2	\$14,95	\$ 29,90
Suporte de motor	2	\$ 0,75	\$1,50
Batería de litio de 3.7 V y 2500mAh	1	\$17,95	\$ 17,95
PowerBoost de 5V y 1000mA	1	\$10,95	\$ 10,95
Ball Caster	1	\$ 2,75	\$2,75
IdeaBoard	1	\$39,95	\$ 39,95
Micro Servomotor MG90S	1	\$ 6,95	\$6,95
Módulo de sensor infrarrojo para evadir obstáculos	2	\$ 2,95	\$5,90
Sensor de APDS9960	1	\$11,95	\$ 11,95
Interruptor general	1	\$ 0,30	\$0,30
STEMMA QT cable	1	\$ 1,49	\$1,49
Piezas acrilico	1	\$ 6,00	\$6,00
Kit conectores*	1	\$ 8,95	\$8,95
Kit de espaciadores y tornillos*	1	\$15,95	\$ 15,95
Cable*	1	\$ 3,95	\$3,95
Juego tornillos M2*	1	\$17,95	\$ 17,95
Total			\$ 182,39

Tabla 5.13: Costos de los componentes de la base.

Dispositivo	Especificaciones	Precio en CR
Computadora	Intel i5 12. ^a generación, 8 GB RAM	\$ 600
Cámara	Resolución 1920x1080	\$ 70
Total		\$ 670

En la Tabla 5.14 se presenta el desglose del costo total del proyecto, que asciende a aproximadamente USD 2588,57. Aunque este valor puede parecer elevado en comparación con otros modelos según la tabla 2.1, no se trata del más costoso. Además, es importante considerar que los activos utilizados y el salario del investigador son gastos únicos, si se desea expandir el sistema de enjambre, solo sería necesario adquirir los componentes de los robots, ya que la parte de diseño ya está cubierta, lo cual reduciría significativamente los costos adicionales.

Tabla 5.14: Análisis económico del proyecto.

Elemento	Coste
3 Robots	\$ 453,57
Componentes base	\$ 670
Salario del investigador	\$ 1682
Activos utilizados	\$ 50
Total	\$ 2588,57

Además, al analizar el costo unitario del robot, se observa en la tabla 2.1 que esta versión del Atta-Bot se posiciona como una opción intermedia en el mercado, con un costo superior

solo al Kilobot y al Colias. Estos dos robots, aunque de gama similar, presentan limitaciones importantes en cuanto a capacidades de comunicación y expansión. Además, el robot Colias no está disponible comercialmente, lo cual restringe su acceso. En el extremo superior del espectro de precios se encuentran el E-puck y el Khepera IV, que, aunque ofrecen una mayor cantidad de sensores de detección e incluso una cámara integrada, tienen precios considerablemente más altos, alcanzando aproximadamente USD 1000 para el E-puck y USD 3200 para el Khepera IV [14].

En comparación con la versión anterior del Atta-Bot, esta nueva versión no solo puede reducir el costo unitario hasta en aproximadamente USD 65, sino que también integra mejoras significativas en tamaño, autonomía, comunicación y precisión en los movimientos. Estas optimizaciones hacen que el Atta-Bot actualizado sea altamente competitivo frente a otros robots comerciales, al ofrecer funcionalidades avanzadas a un costo accesible. Esto hace que la nueva versión sea competitiva frente a robots comerciales, al ofrecer funcionalidades avanzadas a un costo accesible, un factor importante para investigaciones de bajo presupuesto y para instituciones académicas que buscan herramientas educativas asequibles.

Desde el punto de vista investigativo, esta nueva versión del Atta-Bot abre una amplia gama de oportunidades que podrían conducir a nuevos proyectos financiados. Las mejoras en el sistema de comunicación y en la capacidad de procesamiento permiten implementar comportamientos de enjambre más sofisticados, lo que refuerza la capacidad del sistema para la exploración y el mapeo en entornos complejos. Estos avances no solo consolidan el trabajo previo en este proyecto, sino que también pueden dar lugar a nuevas investigaciones y publicaciones elegibles para financiamiento de entidades como la Vicerrectoría de Investigación y Extensión (VIE) del ITCR.

Desde 2019, la VIE ha otorgado un promedio de 2.5 millones de colones (USD 4,868) por año para proyectos dentro del grupo GIROM que han incluido en sus objetivos el diseño, construcción, validación y pruebas de prototipos similares al Atta-Bot. Esto significa que, al igual que otros proyectos, este también podría recibir financiamiento por cantidades similares, lo que podría cubrir los costos de mejora del sistema y contribuir al retorno de inversión al fortalecer la capacidad del laboratorio en robótica de enjambres. Dado que el costo total del sistema completo fue de aproximadamente USD 2588, se podría afirmar que el sistema se amortizaría en su primer año de financiamiento.

También, el grupo de investigación está considerando el uso de este diseño de robot como una plataforma educativa en robótica en la Escuela de Mecatrónica del ITCR. Aunque esta aplicación no generaría ingresos directos, la implementación en entornos de enseñanza representa un valor significativo en términos de formación académica y el desarrollo de habilidades técnicas en los estudiantes. Esto añade valor a la inversión realizada, ya que fomenta el conocimiento, la innovación y la capacitación de los estudiantes. En conclusión, el proyecto no solo amplía el alcance investigativo del laboratorio, sino que también refuerza el compromiso de la institución con la educación de calidad y el desarrollo tecnológico, logrando un impacto positivo en la comunidad educativa y científica.

Capítulo 6

Conclusiones y Recomendaciones

6.1 Conclusiones

Al finalizar este proyecto, se logró adaptar la arquitectura de hardware del Atta-Bot STEM para obtener un sistema de enjambre preciso y eficaz en comparación con su versión anterior. También se implementó un nodo base que amplía las capacidades de este sistema, cumpliendo así con el objetivo principal del proyecto y permitiendo en un futuro la implementación de nuevos comportamientos de enjambre que extenderán el alcance de investigación del laboratorio. Las conclusiones alcanzadas son:

- A partir del diagnóstico realizado a la primera versión del Atta-Bot, se identificaron los principales problemas y áreas de mejora para esta nueva versión, estableciendo además los valores objetivo para el sistema de enjambre actualizado.
- Se implementó un sistema de comunicación que facilita la interacción entre robots y con la base, mejorando significativamente la tasa de éxito en la transmisión de mensajes. En promedio, la comunicación entre robots presenta una tasa de éxito del 99,24%, mientras que la comunicación con la base alcanza un 99,92%.
- Se desarrolló un sistema de detección y clasificación robusto frente a interferencias provocadas por otros robots o por el entorno, logrando una tasa de éxito en detección y clasificación superior al 97%.
- Se cumplieron las nueve métricas establecidas para este proyecto, de las cuales ocho se alcanzaron en su valor ideal y una en su valor marginal. Además, se validó la implementación del sistema de enjambre mediante pruebas de exploración aleatoria, logrando un área explorada promedio de 49,19% en 5 minutos, comparable al 50% alcanzado por la versión anterior en condiciones similares.

6.2 Recomendaciones

Una vez concluido este proyecto, se ofrece una serie de recomendaciones que pueden facilitar y maximizar el aprovechamiento del sistema de enjambre Atta-Bot en futuras investigaciones. Estas son:

- Explorar el uso de tecnologías similares al BLE para optimizar la clasificación de objetos y reducir la carga de comunicación entre los robots, lo cual permitiría mejorar la eficiencia del sistema y ampliar su capacidad de procesamiento.
- Modificar el método de medición de velocidad para que permita un control más fluido y menos escalonado, sin necesidad de aumentar el tiempo de muestreo. Esto facilitaría la implementación de movimientos más complejos en el robot.
- Configurar adecuadamente los sensores HW-488 para ajustar la sensibilidad de detección, ya que esta afecta directamente el comportamiento de los robots. Si la distancia de detección es demasiado amplia, los robots pueden volverse excesivamente cautelosos, limitando su exploración. La configuración óptima de estos sensores permitirá un equilibrio adecuado entre exploración y seguridad.
- Realizar mantenimiento periódico de los motores, ya que durante las pruebas, se observó que los motores tendían a atascarse, requiriendo lubricación de engranajes para restaurar su funcionamiento adecuado. Se recomienda establecer un plan de mantenimiento regular para asegurar que los motores operen de manera óptima y evitar interrupciones durante las pruebas.
- Desarrollar una interfaz de usuario (UI) que facilite la operación y configuración del sistema de enjambre, especialmente si se planea agregar comportamientos o movimientos en el futuro. Esto simplificaría el uso del sistema, haciéndolo más accesible y eficiente para los usuarios.

Bibliografía

- [1] J. Colomer Barbera. Estudio de los sensores para la detección de obstáculos aplicables a robots móviles. (2018)
- [2] N. Laković, M. Brkić, B. Batinić, J. Bajić, V. Rajs, and N. Kulundžić, “Application of low-cost vl53l0x tof sensor for robot environment detection,” in *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, DOI 10.1109/INFO-TEH.2019.8717779, pp. 1–4, 2019.
- [3] O. Á. Arturo, M. E. Oscar, F. de Vega, and O. C. G. Francisco, “Sistema embebido de detección de obstáculos con freertos para un robot móvil.”
- [4] K. Ulrich and S. Eppinger, *Product Design and Development*, 6th ed. McGraw-Hill Education, 2016.
- [5] C. Calderón-Arce, J. C. Brenes-Torres, and R. Solís-Ortega, “Atta-bot: A do-it-yourself approach for swarm robotics,” 2024.
- [6] L. C. Villalobos-Quesada, “Rediseño del robot móvil atta-bot para actividades de educación en colegios públicos,” Master’s thesis, Instituto Tecnológico de Costa Rica, Cartago, Costa Rica, 2024.
- [7] B. Ryerson. Ir sensor for obstacle avoidance ky-032 (ad-032). <http://irsensor.wizocode.com/>. (2023)
- [8] C. Calderón Arce, J. C. Brenes Torres, R. Solís Ortega, K. Morales Paz, and A. Jiménez Mora, “Tracking the trajectory of a swarm of mobile robots with a computer vision system,” *Tecnología en marcha*, vol. 37, no. 3, pp. 44–49, 2024.
- [9] A. A. Espitia Cubillos, E. Rodríguez Carmona, and R. Jiménez-Moreno, “Usability measurement model of the human-robot interactive communication interface for human assistance tasks,” 2024.
- [10] M. Dorigo, G. Theraulaz, and V. Trianni, “Swarm robotics: Past, present, and future [point of view],” *Proc. IEEE*, vol. 109, no. 7, pp. 1152–1165, 2021.
- [11] I. Navarro and F. Matía, “An introduction to swarm robotics,” *International Scholarly Research Notices*, vol. 2013, no. 1, p. 608164, 2013.

- [12] M. Dorigo and M. Birattari, “Swarm intelligence,” *Scholarpedia*, vol. 2, DOI 10.4249/scholarpedia.1462, no. 9, p. 1462, 2007, revision #138640.
- [13] C. Calderón-Arce, J. C. Brenes-Torres, and R. Solis-Ortega, “Swarm robotics: Simulators, platforms and applications review,” *Computation*, vol. 10, no. 6, 2022. [En línea]. Disponible: <https://www.mdpi.com/2079-3197/10/6/80>
- [14] P. G. F. Dias, M. C. Silva, G. P. Rocha Filho, P. A. Vargas, L. P. Cota, and G. Pessin, “Swarm robotics: A perspective on the latest reviewed concepts and applications,” *Sensors*, vol. 21, no. 6, p. 2062, 2021.
- [15] G. R. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. P. Costa, and B. Walke, “The iee 802.11 universe,” *IEEE communications magazine*, vol. 48, no. 1, pp. 62–70, 2010.
- [16] S. VADYM, “Bluetooth low energy compared to zigbee and bluetooth classic,” 2010.
- [17] A. J. Pérez, F. A. Stramana, A. Cejas, A. Chardon, and A. J. Rubiales, “Algoritmos avanzados de posicionamiento en interiores utilizando la combinación de distintos tipos de sensores,” *Mecánica Computacional*, vol. 37, no. 47, pp. 1881–1890, 2019.
- [18] A. Ozer and E. John, “Improving the accuracy of bluetooth low energy indoor positioning system using kalman filtering,” in *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 180–185. IEEE, 2016.
- [19] L. S. Mallari and F. B. Santos Jr, “Internet of things-based remotely simulated educational modular robotics trainer utilizing esp-now protocol.”
- [20] J. C. Brenes-Torres. Proyecto proe: Implementación de un prototipo de enjambre de robots para la digitalización de escenarios estáticos y planificación de rutas óptimas. <https://github.com/jcbrenes/PROE>. (2021)
- [21] C. A. Barrantes-Jiménez, “Diseño e implementación de rutinas de navegación y comunicación para un robot móvil utilizado en sistemas multirobot de enjambre,” Master’s thesis, Instituto Tecnológico de Costa Rica, Cartago, Costa Rica, 2019.
- [22] C. J. Porras-Aguilar, “Implementación de un sistema de comunicación para un sistema multirobot de enjambre.” Master’s thesis, Instituto Tecnológico de Costa Rica, Cartago, Costa Rica, 2022.
- [23] M. A. Afzal, D. He, Z. Zhu, and Y. Yang, “Performance evaluation of wi-fi bluetooth low energy li-fi technology in indoor positioning,” in *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, pp. 1–5. IEEE, 2018.
- [24] J. C. Brenes-Torres. Proyecto atta-bot stem del tecnológico de costa rica. <https://github.com/jcbrenes/Atta-Bot-STEM>. (2024)
- [25] cplusplus. Double ended queue. <https://cplusplus.com/reference/deque/deque/>. (2023)

-
- [26] I. O. for Standardisation, “Ergonomics of human-system interaction—part 11: Usability: Definitions and concepts (iso 9241-11: 2018),” 2018.
- [27] J. Schmidtler, K. Bengler, F. Dimeas, and A. Campeau-Lecours, “A questionnaire for the evaluation of physical assistive devices (quead): Testing usability and acceptance in physical human-robot interaction,” in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, DOI [10.1109/SMC.2017.8122720](https://doi.org/10.1109/SMC.2017.8122720), pp. 876–881, 2017.

Apéndice A

Gráficas de funcionamiento del control PID

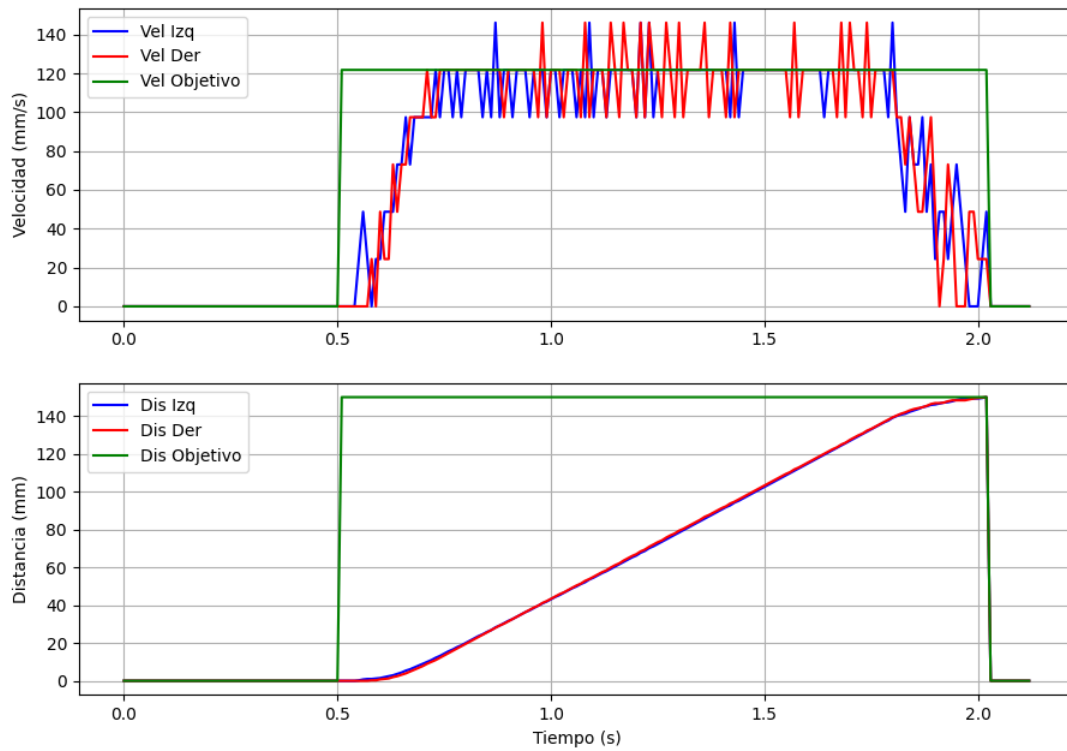


Figura A.1: Gráfica del control PID en funcionamiento para desplazamiento de 150mm.

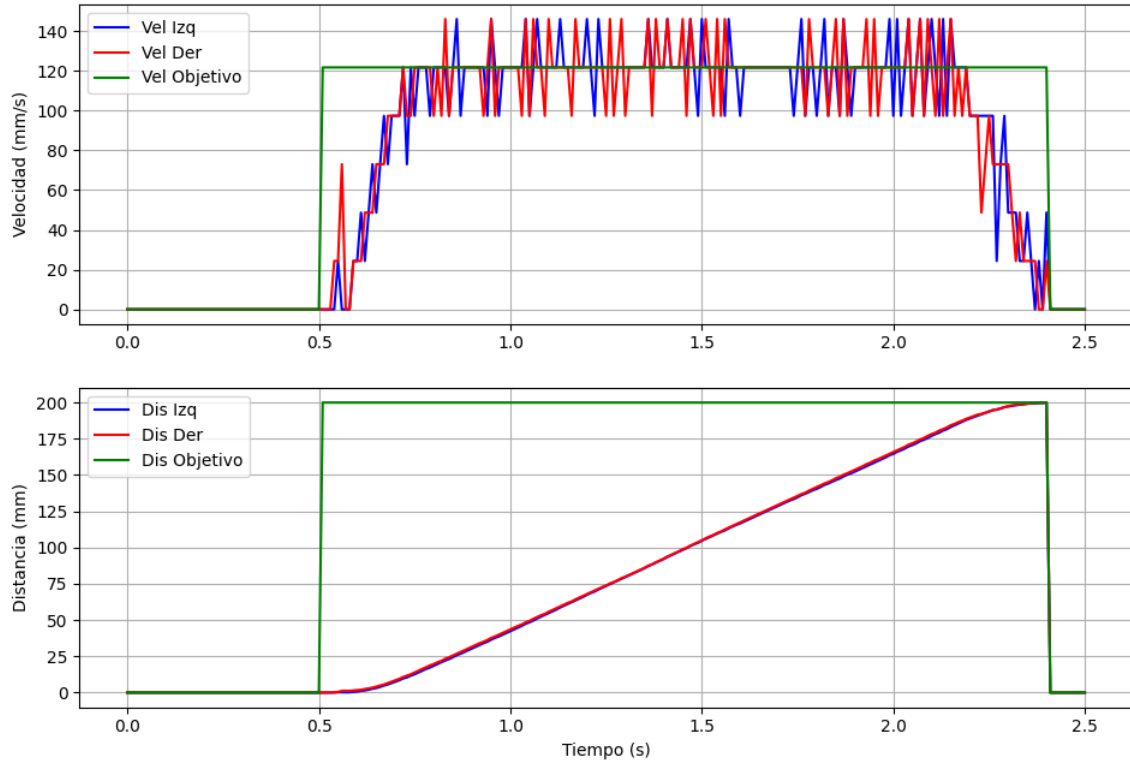


Figura A.2: Gráfica del control PID en funcionamiento para desplazamiento de 200mm.

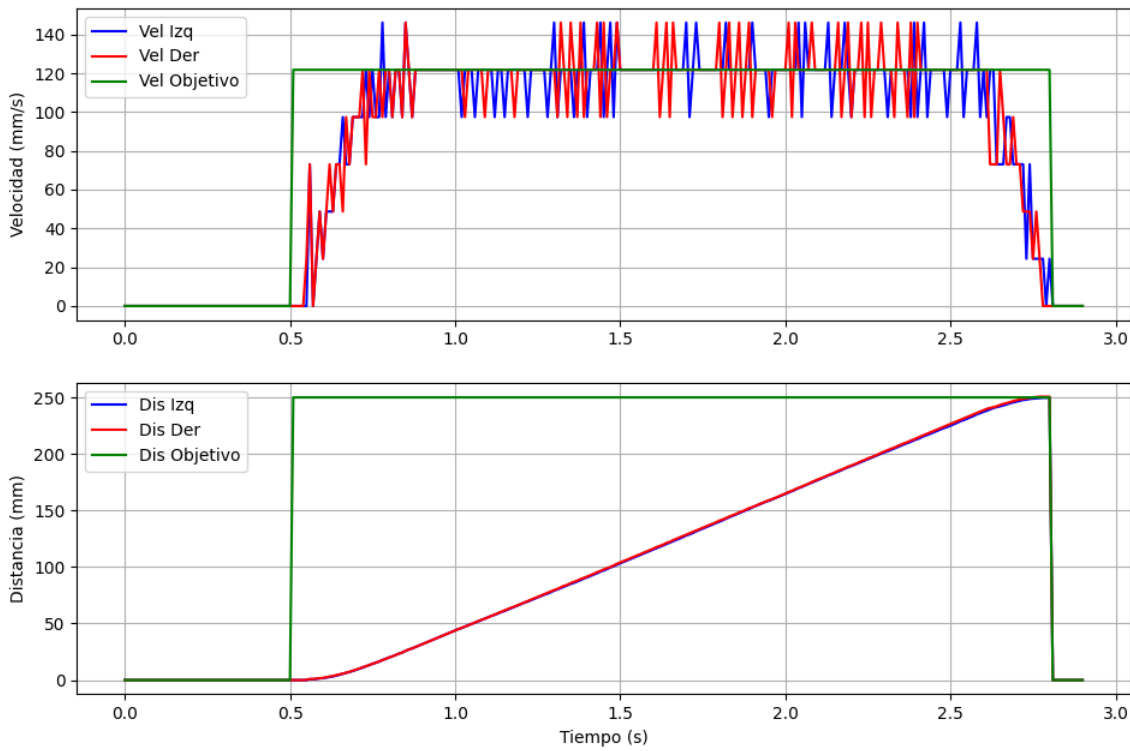


Figura A.3: Gráfica del control PID en funcionamiento para desplazamiento de 250mm.

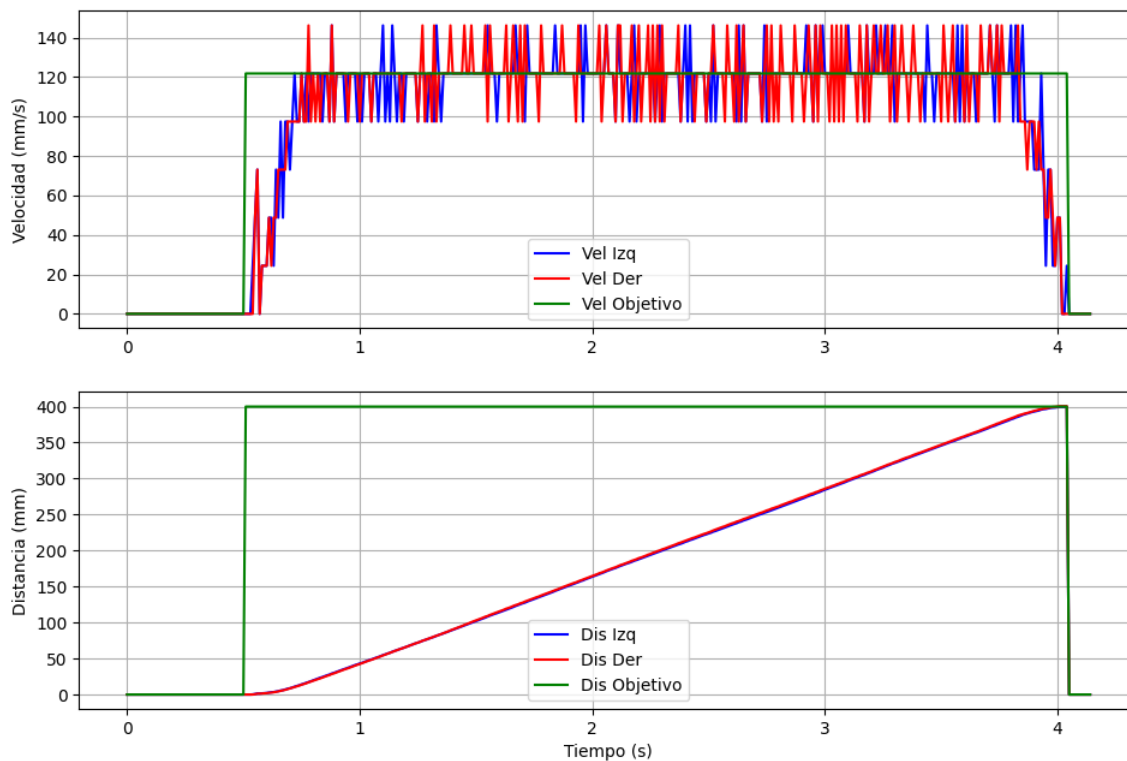


Figura A.4: Gráfica del control PID en funcionamiento para desplazamiento de 400mm.

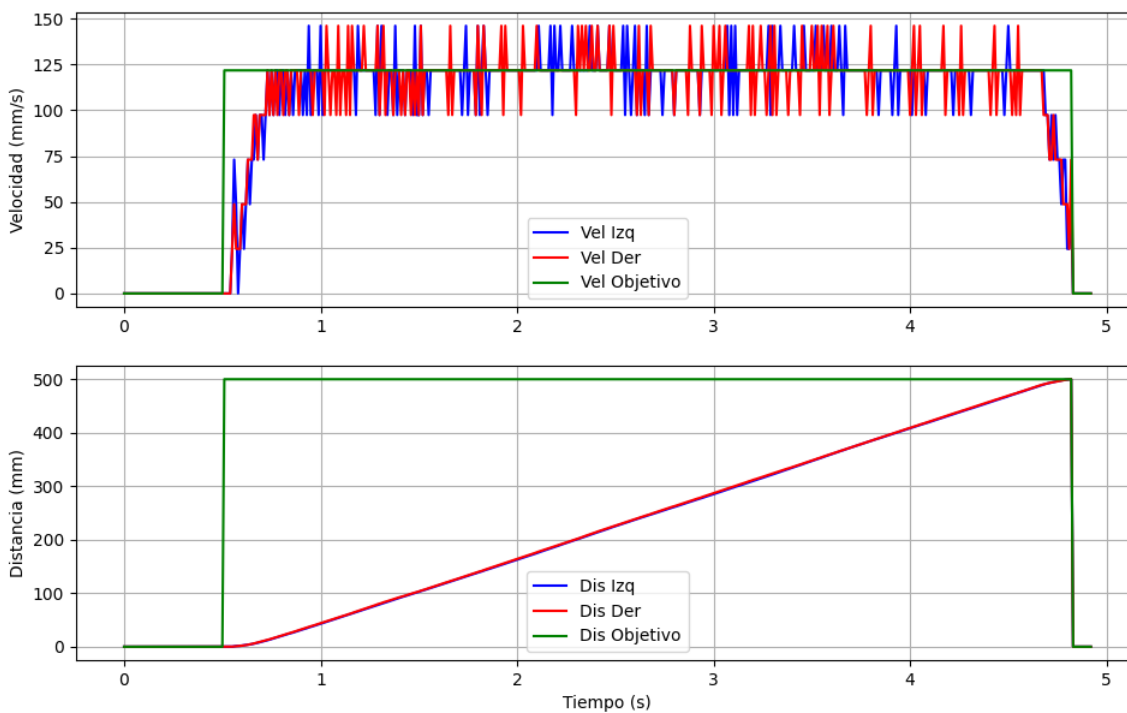


Figura A.5: Gráfica del control PID en funcionamiento para desplazamiento de 500mm.

Apéndice B

Código de los Robots y la Base

El código fuente del proyecto está disponible en GitHub en el siguiente enlace:

<https://github.com/jcbrenes/Atta-Bot/tree/7e792553dbe18d4255da5b39db2ca0d90793e46d>

Este repositorio contiene el código completo tanto para los robots como para la base central.