



ESPE

**UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA**

**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN
Y CONTROL**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERA EN ELECTRÓNICA, AUTOMATIZACIÓN Y
CONTROL**

**TEMA: DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS CON
ROBÓTICA COOPERATIVA PARA LA CLASIFICACIÓN DE OBJETOS
DE COLOR EN UN ENTORNO ESTRUCTURADO**

AUTOR: BONILLA PERASSO, GABRIELA ELIZABETH

DIRECTOR: MSc. ERAZO SOSA, ANDRÉS SEBASTIÁN

SANGOLQUÍ

2019



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

CERTIFICACIÓN

Certifico que el trabajo de titulación: “DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS CON ROBÓTICA COOPERATIVA PARA LA CLASIFICACIÓN DE OBJETOS DE COLOR EN UN ENTORNO ESTRUCTURADO” fue realizado por la señorita GABRIELA ELIZABETH BONILLA PERASSO, el mismo que ha sido revisado en su totalidad, analizando por la herramienta de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de la Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, Enero del 2019

A handwritten signature in blue ink, appearing to read 'Andrés Erazo', is written over a horizontal line.

ING. ANDRÉS ERAZO, Msc.

C.C: 1720400082



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES
CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORÍA DE RESPONSABILIDAD

Yo, GABRIELA ELIZABETH BONILLA PERASSO, declaro que el contenido, ideas y criterios del trabajo de titulación: “DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS CON ROBÓTICA COOPERATIVA PARA LA CLASIFICACIÓN DE OBJETOS DE COLOR EN UN ENTORNO ESTRUCTURADO” es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, Enero del 2019

Gabriela Elizabeth Bonilla Perasso

C.C: 1804971636



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

Yo, GABRIELA ELIZABETH BONILLA PERASSO, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: “DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS CON ROBÓTICA COOPERATIVA PARA LA CLASIFICACIÓN DE OBJETOS DE COLOR EN UN ENTORNO ESTRUCTURADO” en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, Enero del 2019

Gabriela Elizabeth Bonilla Perasso

C.C: 1804971636

DEDICATORIA

El presente proyecto de titulación lo dedico a mi familia, que es el pilar fundamental de mi vida y el motor que me da fuerzas para seguir. Gracias a su amor, paciencia y apoyo incondicional he llegado a cumplir esta meta.

A mi mami Gladis Perasso, quien es la persona más importante de mi vida. Gracias por tu amor, por tus sacrificios, por tus regaños, gracias por ser la mejor madre del mundo y siempre estar a mi lado.

A mis tías Miriam y Paty, quienes son como mis otras mamás. Gracias a su cariño, apoyo y paciencia he llegado a cumplir este sueño.

A mis abuelitos Teresa y Rogelio, por inculcar en mí el ejemplo de esfuerzo y valentía, de no temer a las adversidades y siempre mirar al frente sin importar las circunstancias. Gracias a sus consejos, cariño y palabras de aliento hicieron de mí una mejor persona.

Finalmente quiero dedicar esta tesis a mi novio Vinicio, por apoyarme y estar junto a mí cuando más lo necesitaba, por brindarme su apoyo en momentos difíciles, por su amor, por sus palabras de aliento y por su infinita paciencia.

Gabriela E. Bonilla P.

AGRADECIMIENTO

Agradezco en primer lugar a mi mami Gladis, por apoyarme y confiar en mis decisiones. Por enseñarme valores, costumbres y principios que formaron mi personalidad. Gracias por el ejemplo de mujer que eres, y que aspiro llegar a ser. Mami me has demostrado que una mujer puede cumplir todos sus sueños con esfuerzo y dedicación, a pesar que el camino a ellos este lleno de adversidades.

A mis tías Miriam y Paty, quienes me acogieron en su casa, brindándome apoyo y cariño en estos años de estudio. Son más que mis tías, son mis segundas madres.

A mi primo Alex, por tantos momentos de risas, enojo y alegrías que compartimos, y por estar para mí en los momentos difíciles. A pesar de tener personalidades opuestas has demostrado ser un hermano para mí, y siempre tendrás mi cariño y apoyo incondicional.

A mis familiares, especialmente a mis abuelitos a quienes considero los pilares fundamentales de mi vida, gracias por haberme brindado su apoyo y creer en mí.

A mis amigos, gracias por su amistad en los buenos y malos momentos, gracias por su apoyo y palabras de aliento.

Y finalmente gracias a mi novio Vinicio, eres más que un novio, eres mi amigo y compañero. Gracias por brindarme tu apoyo en los momentos más difíciles de mi vida, por impulsarme a ser mejor, gracias por intentar comprenderme a pesar de lo difícil que puedo ser.

Gabriela E. Bonilla P.

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN	i
AUTORÍA DE RESPONSABILIDAD.....	ii
AUTORIZACIÓN.....	iii
DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE TABLAS	x
ÍNDICE DE FIGURAS.....	xi
CAPITULO I.....	1
GENERALIDADES.....	1
1.1 Antecedentes	1
1.2 Justificación e importancia.....	4
1.3 Alcance del proyecto.....	6
1.4 Objetivos	9
1.4.1 Objetivo General.....	9
1.4.2 Objetivos Específicos	9
1.5 Descripción del proyecto de investigación	10
CAPITULO II	12
ESTADO DEL ARTE.....	12
2.1 Introducción	12
2.2 Robótica Cooperativa Móvil	13
2.2.1 Arquitecturas de control en robótica cooperativa	15
2.3 Visión artificial aplicada a robótica móvil	18

2.3.1	Proceso de formación de imágenes de la cámara	19
2.3.2	Modelo de color	21
2.3.3	Procesamiento digital de imágenes	25
2.3.4	Calibración de cámara	32
2.3.5	Cámaras IP	34
2.4	Métodos de estimación de posición	36
2.4.1	Métodos de estimación basados en la percepción del entorno	37
2.4.2	Métodos de estimación explícita.....	38
2.5	Estimación de posición basándose en RSSI.....	44
2.5.1	Triangulación.....	46
2.5.2	Fingerprints o reconocimiento de patrones.....	46
2.6	Planificación de Trayectoria	47
2.6.1	Método de planificación de trayectoria RRT.....	50
CAPITULO III		53
HARDWARE Y SOFTWARE		53
3.1	Introducción	53
3.2	Hardware del sistema	53
3.2.1	Entorno estructurado.....	54
3.2.2	Robots móviles	56
3.2.3	Sensores y actuadores	60
3.2.4	Tarjetas de desarrollo.....	77
3.2.5	Batería LiPo	83
3.2.6	Wireless router.....	85
3.2.7	Computador de procesamiento	86

3.3 Software	87
3.3.1 Primera etapa	90
3.3.2 Segunda etapa	92
3.3.3 Tercera etapa.....	98
3.4 Problemas enfrentados en hardware y software	99
CAPITULO IV	100
IMPLEMENTACIÓN DE HARDWARE Y SOFTWARE	100
4.1 Introducción	100
4.2 Implementación de hardware en los robots móviles del sistema	100
4.2.1 Robot guía.....	100
4.2.2 Robots obreros	103
4.3 Desarrollo del sistema de adquisición de datos.....	105
4.3.1 Configuración de la tarjeta teensy 3.5	106
4.3.2 Configuración de la tarjeta nodeMCU.....	109
4.3.3 Configuración de la cámara IP.....	111
4.4 Desarrollo del sistema de procesamiento de datos.....	113
4.4.1 Procesamiento de datos obtenidos de los sensores	113
4.4.2 Procesamiento de datos obtenidos de la tarjeta nodeMCU.....	114
4.5 Control centralizado para el sistema multi-robot cooperativo	115
4.6 Implementación de algoritmos.....	117
4.6.1 Algoritmo de localización de objetos de color	117
4.6.2 Algoritmo de localización Fingerprints	120
4.6.3 Algoritmo RRT para generación de trayectoria.....	121
4.6.4 Algoritmo Pure Pursuit para seguimiento de trayectoria.....	122

CAPITULO V	124
PRUEBAS EXPERIMENTALES DE FUNCIONAMIENTO Y RESULTADOS	124
5.1 Introducción	124
5.2 Pruebas realizadas al algoritmo de detección de objetos de color	124
5.3 Pruebas realizadas al algoritmo de localización de robots	130
5.3.1 Método de triangulación	130
5.3.2 Método de FingerPrints	141
5.4 Pruebas realizadas a los algoritmos de planificación y seguimiento de trayectoria.....	143
5.5 Pruebas realizadas a los algoritmos de planificación y seguimiento de trayectoria.....	147
5.6 Análisis del sistema multi-robot cooperativo implementado	149
5.6.1 Análisis del tiempo de ejecución del sistema multi-robot	149
5.6.2 Análisis del subsistema de comunicación.....	152
CAPITULO VI.....	154
CONCLUSIONES Y RECOMENDACIONES	154
6.1 Conclusiones	154
6.2 Recomendaciones.....	156
REFERENCIAS BIBLIOGRÁFICAS	157

ÍNDICE DE TABLAS

Tabla 1 <i>Especificación de colores a usar en el proyecto.</i>	6
Tabla 2 <i>Análisis comparativo entre sensores de distancia.</i>	67
Tabla 3 <i>Análisis comparativo entre tarjetas de desarrollo.</i>	83
Tabla 4 <i>Análisis de consumo de energía para la selección de la batería.</i>	84
Tabla 5 <i>Hardware implementado en el robot guía.</i>	101
Tabla 6 <i>Hardware implementado en los robots obreros.</i>	103
Tabla 7 <i>Rango de valores HSV para cada color utilizado.</i>	118
Tabla 8 <i>Análisis de la variabilidad de la señal RSSI con dos AP's.</i>	132
Tabla 9 <i>Análisis de la señal RSSI implementando cuatro AP's respecto al router.</i>	133
Tabla 10 <i>Análisis de la señal RSSI implementando cuatro AP's respecto al robot guía.</i>	134
Tabla 11 <i>Análisis de la señal RSSI implementando cuatro AP's respecto a Ref1.</i>	135
Tabla 12 <i>Análisis de la señal RSSI implementando cuatro AP's respecto al Rf2.</i>	136
Tabla 13 <i>Pruebas de medición de la señal RSSI respecto al AP router.</i>	138
Tabla 14 <i>Pruebas de medición de la señal RSSI respecto al AP robot guía.</i>	139
Tabla 15 <i>Pruebas de medición de la señal RSSI respecto al AP Ref1.</i>	139
Tabla 16 <i>Pruebas de medición de la señal RSSI respecto al AP Ref2.</i>	140
Tabla 17 <i>Tiempo de ejecución del sistema.</i>	150
Tabla 18 <i>Latencia en los elementos de la red de comunicación inalámbrica.</i>	152

ÍNDICE DE FIGURAS

Figura 1. Esquema del entorno estructurado.	7
Figura 2. Formación triángulo de robots terrestres	14
Figura 3. Sistema de fútbol robótico vista de la cámara cenital	17
Figura 4. Robots en prueba de vigilancia de un perímetro.....	18
Figura 5. Cámara pin-hole.....	20
Figura 6. Componentes del modelo pin-hole	20
Figura 7. Cubo esquemático del modelo RGB	22
Figura 8. Esquemático del modelo HSI.....	23
Figura 9. Modelo HSI representado en (a) triángulos. (b) círculos.....	24
Figura 10. Ciclo del procesamiento digital de imágenes.....	26
Figura 11. Máscara de 3x3	27
Figura 12. Máscara para detección de líneas (a) horizontal, (b) vertical, (c) +45o, (d) +45o	29
Figura 13. (a) Imagen original, (b) histograma, (c) imagen resultante.....	31
Figura 14. Plantilla (a) 3D, (b) 2D para el proceso de calibración.....	34
Figura 15. Conexión en red de una cámara IP.....	35
Figura 16. Esquemático de un robot móvil diferencial	39
Figura 17. Método de trilateración	41
Figura 18. Método de triangulación	42
Figura 19. Distribución orbital de los satélites.....	42
Figura 20. Método de localización Fingerprints.....	47
Figura 21. Esquemático del algoritmo RRT.....	50
Figura 22. Diagrama de bloques del hardware del sistema.	54
Figura 23. Escenario de pruebas.....	55
Figura 24. Objetos de color.	56
Figura 25. Mini 3-layer round robot de Adafruit.	57
Figura 26. Dimensiones del chasis de los robots.....	57

Figura 27. Motorreductor DC.....	58
Figura 28. Llantas de silicona marca Pololu.	59
Figura 29. Sensor ultrasónico HC-SR04.	61
Figura 30. Rendimiento del sensor HC-SR04.	61
Figura 31. Curva de medición del sensor HC-SR04	62
Figura 32. Error de medición del sensor HC-SR04.....	63
Figura 33. Sensor láser VL53L0X.....	64
Figura 34. Curva de medición del sensor VL53L0X.	65
Figura 35. Error de medición del sensor VL53L0X.....	66
Figura 36. Comparación del error de medición entre los sensores HC-SR04 vs. VL53L0X...68	
Figura 37. Sensor MPU6050.	69
Figura 38. Mini-cámara IP.	72
Figura 39. Driver L298N.....	73
Figura 40. Servomotor SG90.....	75
Figura 41. Diseño del brazo electromecánico.	76
Figura 42. Brazo electromecánico.....	77
Figura 43. Esquema pinout de la tarjeta teensy 3.5	79
Figura 44. Esquema pinout de la tarjeta nodeMCU V1.0 V2	80
Figura 45. Batería LiPo 7.4V y 2200 mAh.....	84
Figura 46. Wireless router TP-Link modelo TL-WR840N.....	85
Figura 47. Portátil MSI modelo GE62 2QD Apache.	86
Figura 48. Diagrama de bloques del software del sistema.	88
Figura 49. Diagrama de flujo del software del sistema.	89
Figura 50. Diagrama de flujo del proceso de estimación de posición de los objetos de color.	90
Figura 51. Diagrama de flujo del proceso de envío de señales al robot desde la PC.....	92
Figura 52. Esquemático de la posición de los robots dentro del escenario.	93
Figura 53. Diagrama de flujo del proceso de estimación de posición de los robots.	95
Figura 54. Esquemático de la posición de los robots dentro del escenario.	96
Figura 55. Diagrama de flujo del proceso de estimación de posición de los robots.	98

Figura 56. Diagrama de flujo del proceso de planificación y seguimiento de trayectoria.	98
Figura 57. Diagrama de conexión del robot guía. (1) sensor HC-SR04, (2) sensor MPU6050, (3) Teensy 3.5, (4) nodeMCU V2, (5) driver L298N, (6) motor DC derecho, (7) motor DC izquierdo, (8) servomotor de cámara IP, (9) servomotor de orientación, (10) LM7805, (11) batería LiPo. .	102
Figura 58. Robot guía.	103
Figura 59. Diagrama de conexión de los robots obreros. (1) sensor MPU6050, (2) Teensy 3.5, (3) nodeMCU V2, (4) driver L298N, (5) motor DC derecho, (6) motor DC izquierdo, (7) servomotor del brazo robótico, (8) servomotor de orientación, (9) LM7805, (10) batería LiPo.	104
Figura 60. Robots obreros	105
Figura 61. Diagrama de flujo del proceso de la comunicación serial de la tarjeta teensy 3.5.	107
Figura 62. Diagrama de flujo del proceso de adquisición de datos para el robot guía.....	108
Figura 63. Diagrama de flujo del proceso de adquisición de datos para los robots obreros.	109
Figura 64. Diagrama de flujo del proceso de comunicación serial de la tarjeta nodeMCU.....	110
Figura 65. Diagrama de flujo del proceso de adquisición de datos para los robots obreros.....	111
Figura 66. Asignación de dirección IP estática a la cámara.	112
Figura 67. Conexión de la cámara IP a la red.....	112
Figura 68. Íconos correspondientes a Python y OpenCV.....	113
Figura 69. Medición del valor RSSI cuando el robot está en estado estacionario.	115
Figura 70. Control central del sistema cooperativo y asignación de IP's.	116
Figura 71. Obtención del rango de valores para el color amarillo.	118
Figura 72. Secuencia de localización de objetos de color vista aérea.	119
Figura 73. Secuencia de localización de objetos de color vista cámara IP.	119
Figura 74. Secuencia de toma de datos método Fingerprints celdas (a) 1, (b) 38, (c) 31, (d) 25	120

Figura 75. Generación aleatoria de trayectoria.....	121
Figura 76. Seguimiento de ruta con el robot y punto de mira siguiente.....	122
Figura 77. Seguimiento de trayectoria con el algoritmo Pure Pursuit para dos puntos del escenario.	123
Figura 78. Detección de los objetos de color con iluminación artificial (a) esferas y (b) cilindros	125
Figura 79. Detección de los objetos de color con iluminación (a) artificial y (b) mixta.....	126
Figura 80. Mediciones del radio del cilindro para 200 datos	127
Figura 81. Mediciones del Área del cilindro para 200 datos.....	128
Figura 82. Error porcentual de medición del radio en el algoritmo de detección de color	129
Figura 83. Error porcentual de medición del área del cilindro en el algoritmo de detección de color	129
Figura 84. Posición y orientación de los robots en pruebas de RSSI (a) 15cm, (b) 50cm, (c) 75cm y (d) 100cm.....	131
Figura 85. Medición del valor RSSI respecto al AP router	133
Figura 86. Medición del valor RSSI respecto al AP robot guía	134
Figura 87. Medición del valor RSSI respecto al AP referencia amarillo	135
Figura 88. Medición del valor RSSI respecto al AP referencia azul.....	136
Figura 89. Medición del valor RSSI respecto a todos los AP's variando la posición y orientación del robot obrero.	137
Figura 90. Análisis de errores de las pruebas de medición de la señal RSSI	141
Figura 91. Porcentaje de aciertos en la medición de un conjunto de 300 datos para cada celda.	142
Figura 92. Dispersión de localización calculada en los puntos 13, 19 y 40.	143
Figura 93. Localización de objetos de color y robots en el escenario	145
Figura 94. Generación aleatoria de trayectoria para recoger el objeto de color	145
Figura 95. Seguimiento de la trayectoria para recoger el objeto de color	146
Figura 96. Secuencia de imágenes del seguimiento de la trayectoria del robot para recoger el objeto de color	147
Figura 97. Generación aleatoria de trayectoria para transportar el	

objeto de color a su correspondiente pared.....	148
Figura 98. Seguimiento de la trayectoria para transportar el objeto de color a su correspondiente pared.....	148
Figura 99. Secuencia de imágenes del seguimiento de la trayectoria para transportar el objeto de color a su correspondiente pared.	149
Figura 100. Análisis parcial del tiempo de ejecución del sistema.....	151
Figura 101. Análisis total del tiempo de ejecución del sistema.	152

RESUMEN

En el presente trabajo de titulación se realiza la búsqueda, detección y clasificación de objetos de color en un entorno estructurado, mediante el diseño e implementación de algoritmos de control en un sistema de múltiples robots cooperativos con método de control central. El sistema consta de cinco robots en configuración diferencial, de los cuales solamente el robot guía está equipado con una cámara, la misma que proporciona información visual para la detección y localización de objetos de color en el entorno estructurado; además de un sensor ultrasónico para la corroboración de los datos de distancia obtenidos de la cámara. Los demás robots están provistos con brazos electromecánicos para la clasificación de los objetos de color, y de sensores para la geolocalización. Los robots utilizan el método RSSI (Received Signal Strength Indicator), para determinar la ubicación de los mismos en el entorno.

PALABRAS CLAVE:

- ROBOT COOPERATIVO
- RSSI

ABSTRACT

The aim of the present study is to detect and classify colored objects in an structured environment using control algorithms in a cooperative multiple robot system with a central control method. The system was composed by five robots disposed in a differential configuration. Leading robot was equipped with a camera which provided visual information for the detection and location of the colored objects in the structured environment, and an ultrasound sensor which corroborate the distance data obtained from the camera. The other robots were provided with electromechanical arms, for the classification of colored objects, and geo location sensors. Robots used the RSSI method (Received Signal Strength Indicator) to determine their location in the environment.

PALABRAS CLAVE:

- COOPERATIVE ROBOT
- RSSI

CAPITULO I

GENERALIDADES

1.1 Antecedentes

El campo de la robótica cooperativa en la actualidad ha evolucionado, debido a que varias aplicaciones tanto de investigación como militares pueden ser física y computacionalmente complejas para que las resuelva un solo robot (Murray, s. f.), (Morales, Pozo, D.; Rosero, Sandobalin, & Rodríguez, 2014). La robótica cooperativa promete robots más robustos y flexibles, a diferencia de los robots individuales que están restringidos por su tamaño, versatilidad y fuerza imposibilitando la realización de múltiples tareas (Inglett & Rodríguez-Seda, 2017; Murray, s. f.). Los sistemas multi-robot presentan un diseño y control sencillo, ofrecen mayor cobertura y se adaptan fácilmente a entornos desconocidos siendo útiles en tareas de exploración (Morales et al., 2014), vigilancia (Murray, s. f.), monitoreo de objetos en movimiento (Khan et al., 2016) y transporte de objetos (Murray, s. f.) ,(Inglett & Rodríguez-Seda, 2017).

La capacidad de percepción del entorno permite al robot navegación autónoma e identificación de objetos de interés, existen diversas técnicas y sensores para obtener información del ambiente (González Jiménez & Baturone, 2015), la utilización de cámaras junto con sistemas de navegación inercial (INS) (Etxeberria, 2010; González Jiménez & Baturone, 2015), (Efrain Ernesto, Arturo, & Juan, 2015) es una de las técnicas más utilizadas debido a que proporciona la información necesaria para detectar objetos y estimar la posición del robot.

Un sistema basado en visión artificial para la detección de objetos es afectado principalmente por la iluminación, como se muestra en (Chicala, Jacho, Atiencia, & Vintimilla, 2006) una iluminación uniforme garantiza que el proceso de reconocimiento de los objetos sea preciso.

En la actualidad existen infinidad de trabajos que presentan técnicas de detección de objetos basados en el color y/o la forma. Los siguientes trabajos (Lin, Wang, Yang, & Lai, 2015), (Chen et al., 2016) se basan en detección por forma, por otro lado (Danelljan, Khan, Felsberg, & Van De Weijer, 2014), (Possegger & Mauthner, 2015), presentan técnicas de detección basadas en el color. Además en (Efrain Ernesto et al., 2015; Etxeberria, 2010) se aborda técnicas de detección de objetos basadas tanto en forma como en color.

La clasificación y almacenamiento de sustancias químicas es de vital importancia en la industria, el manejo seguro de químicos evita incendios y/o explosiones que causan tanto lesiones al personal como daños a los equipos. Existen varios sistemas internacionales para la clasificación y almacenamiento de sustancias químicas (Carrascal Quintero, 2010), dentro de ellos destaca SAF-T-DATA de J.T. Baker que maneja un método codificado en colores el cual organiza las sustancias químicas de acuerdo a ciertas propiedades y les asigna un color, almacenando juntos los productos de similar color (Colmenares, 2014).

Por otro lado, los adelantos tecnológicos tanto en comunicaciones inalámbricas como en electrónica hacen posible el uso de redes inalámbricas de sensores (WSNs) de costo reducido y bajo consumo de potencia (Maria Garcia, 2008). La localización de personas u objetos es uno de los servicios más importantes en el campo de WSNs (Dong & Xu, 2014) tanto en espacios exteriores como interiores.

En espacios exteriores se puede conocer la posición a cada instante de tiempo mediante el uso de GPS (Global Positioning System) como se ve en (Maria Garcia, 2008; Nieto Duran, Santos Quintero, Escobar Vargas, & Salinas, 2017). Además se conoce que las tecnologías usadas para la localización en exteriores no suelen ser muy efectivas en interiores debido al margen de error que presentan (Hoyo Mondeja, 2012) . En espacios interiores se utilizan técnicas como: tiempo de llegada de la señal (ToA – Time of arrival), diferencia del tiempo de llegada (DToA – Time difference of arrival), ángulo de llegada (AoA – Angle of arrival) o fuerza de señal recibida (RSS – Received Signal Strength) (Hoyo Mondeja, 2012; Maria Garcia, 2008).

RSS o Fuerza de señal recibida es una técnica de localización que utiliza las propiedades de atenuación de la señal para obtener la distancia entre dos nodo (Hoyo Mondeja, 2012; Segura, 2010), según (Maria Garcia, 2008) esta técnica suele ser precisa en distancias cortas con un margen de error del 10% a distancias máximas de 20 metros.

Es esencial para los robots móviles que realizan tareas en grandes o medianos espacios de trabajo planificar la trayectoria de movimiento; para así llevar al robot desde una configuración inicial hasta otra final dentro de un espacio libre de colisión (Álvarez Romero & Figueroa Montenegro, 2010; López, Gómez-Bravo, Cuesta, & Ollero, 2010). Algunos de los algoritmos utilizados para realizar el control sobre la trayectoria son Roadmap, Rapidly Exploring Random Trees (RRT), Mapas probabilísticos y algoritmo de gusano, los mismos que son analizados en (Álvarez Romero & Figueroa Montenegro, 2010; Diego & Fernando, 2012; López et al., 2010) respectivamente.

Con el uso de los algoritmos antes mencionados y el continuo procesamiento de imágenes se puede generar una estimación de trayectoria aplicable al sistema de multi-robot, permitiéndole autonomía en la navegación dentro de un entorno estructurado.

Tomando en cuenta estos antecedentes, y con el objetivo de superar ciertas dificultades expuestas y los desafíos en el uso de robots cooperativos para la clasificación de objetos de color, utilizando geolocalización y seguimiento de trayectoria en un entorno estructurado de pruebas se propone este proyecto titulado “Diseño e implementación de algoritmos con robótica cooperativa para la clasificación de objetos de color en un entorno estructurado”.

1.2 Justificación e importancia

La implementación de sistemas de múltiples robots cooperativos para dar solución a problemas complejos ha sido tema de investigación durante los últimos años y hoy en día es una tecnología usada en ámbitos cotidianos de la humanidad.

La principal ventaja de la robótica cooperativa reside en la forma de resolver una tarea compleja dividiéndola en varias tareas sencillas que son realizadas por robots de diseño simple, obteniendo mayor flexibilidad de funcionamiento y tolerancia a fallos debido al número de robots utilizados en el sistema.

La idea principal de utilizar robótica cooperativa en la clasificación de objetos de color reside en la utilización de robots dispuestos a cooperar en la realización de tareas como detección y clasificación de objetos de color.

La realización de este trabajo de titulación se respalda en trabajos anteriores tales como: el transporte cooperativo de objetos (Inglett & Rodríguez-Seda, 2017), técnicas de detección de objetos basadas en forma y/o color (Lin et al., 2015) (Chen et al., 2016) (Danelljan et al., 2014) (Possegger & Mauthner, 2015), y la localización de objetos en interiores mediante el uso de RSS (Hoyo Mondeja, 2012; Maria Garcia, 2008).

El uso de robots en espacios industriales genera una gran aportación en seguridad al personal, debido a que los robots pueden realizar tareas en condiciones y ambientes hostiles. La detección, clasificación y almacenamiento de sustancias químicas es de gran importancia dentro de la industria, debido a que el manejo seguro de químicos garantiza un ambiente de trabajo libre de lesiones al personal o daños a los equipos.

Dentro de la industria química existen diversos métodos para la clasificación de sustancias químicas analizados en (Carrascal Quintero, 2010). Para el presente proyecto de titulación se trabaja con el método SAF-T-DATA de J.T. Baker mismo que maneja con un código de colores para organización de sustancias debido a ciertas propiedades almacenando juntos productos de similar color.

Para el presente proyecto de titulación se trabaja con los colores: rojo, azul, verde y amarillo. La elección de estos colores se realiza de acuerdo a ciertos requisitos como son: atraer la atención del usuario y brindarle una interpretación rápida y segura de los posibles riesgos.

Tabla 1

Especificación de colores a usar en el proyecto.

Color	Tipo de químico	Especificación
Rojo	Inflamable	Área de almacenamiento de químicos con riesgo de inflamación.
Azul	Tóxico	Área de almacenamiento de químicos con riesgo para la salud.
Verde	No peligroso	Área de almacenamiento de sustancias químicas que no ofrecen un riesgo importante.
Amarillo	Oxidante	Área de almacenamiento de químicos con riesgo de oxidación y reactividad.

Fuente: (Carrascal Quintero, 2010)

El presente proyecto de titulación se enfoca en robótica cooperativa; compartición de datos entre robots para conocer la posición de los mismos y generar trayectorias. Por estas razones, se pretende realizar un sistema multi-robot cooperativo; geolocalización y seguimiento de trayectoria en un entorno estructurado para la clasificación de objetos de color.

1.3 Alcance del proyecto

El presente trabajo de titulación tiene como objetivo el diseño e implementación de algoritmos de robótica cooperativa en un sistema multi-robot utilizando geolocalización y seguimiento de trayectoria en un entorno estructurado para la clasificación de objetos de color de acuerdo al método SAF-T-DATA de J.T. Baker (Colmenares, 2014), con el fin de emular el proceso de trasladar a un área segura sustancias químicas.

El entorno estructurado de trabajo es de aproximadamente $1.5 m^2$, donde se ubican el sistema multi-robot y los objetos de colores de manera aleatoria con ciertas limitaciones. El tamaño de los robots y de los objetos de color es acorde con el entorno como se observa en la Figura 1.

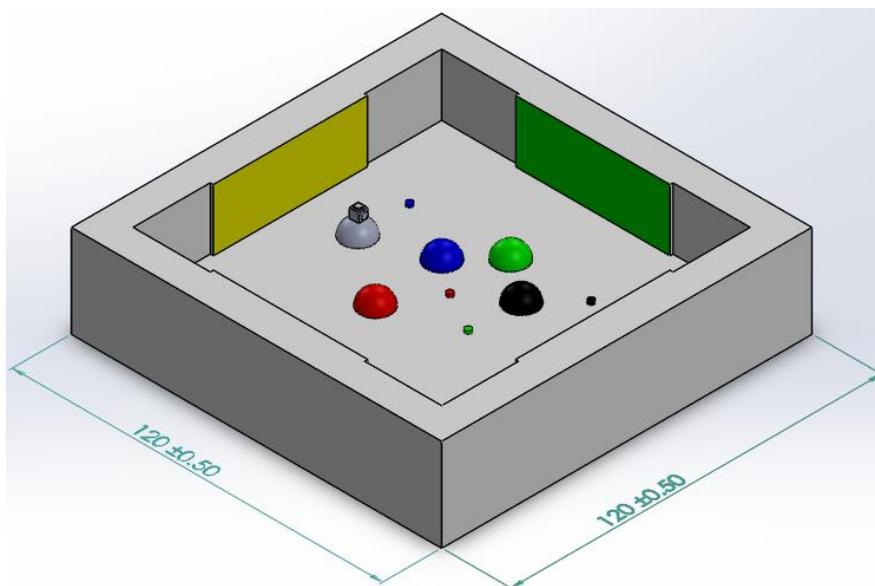


Figura 1. Esquema del entorno estructurado.

El sistema multi-robot cooperativo está basado en cinco robots terrestres de dos ruedas de configuración diferencial, de los cuales un robot consta de una cámara para la detección de objetos; siendo este el robot guía. Todos los robots tienen en general sensores de navegación, microcontroladores y módulos WiFi para la comunicación entre los robots y con un computador de procesamiento.

La posición inicial de los robots obreros dentro del entorno estructurado es aleatoria para aprovechar la movilidad de los mismos. La posición inicial del robot guía en el entorno estructurado es definida, así la navegación del robot guía en el proceso de detección de objetos es limitada, como se observa en la Figura 1.

La detección de objetos se realiza por medio de visión artificial a través de la cámara monocular del robot guía, enfocándose en objetos de los colores: rojo, azul, verde y amarillo, los cuales se detallan en la Tabla 1. Para dicha detección se capturan imágenes en tiempo real, se realiza el procesamiento continuo de las mismas en un computador y de acuerdo al color detectado se envía instrucciones para la clasificación de los objetos y el seguimiento de trayectoria de manera inalámbrica a los robots restantes.

Los cuatro robots restantes responden cada uno a un color diferente, en la clasificación de objetos de color y seguimiento de trayectoria. Estos son los encargados de llevar un objeto a una posición determinada dentro del entorno estructurado; en donde los obstáculos son los objetos de diversos colores y los mismos robots.

Los robots van a tener la capacidad de navegar por su espacio de trabajo respondiendo al movimiento de los robots, los objetos de color y las paredes con respecto al entorno en base a los datos provenientes de los sensores de navegación y el uso de algoritmos para el seguimiento de trayectorias.

Para la geolocalización de los robots se puede utilizar el método RSSI que permite saber la ubicación a cada instante de los mismos. Para el entorno estructurado en el que se trabaja se prevé que el error en posición será cerca de 10%, es decir aproximadamente 10 centímetros, en base a la documentación encontrada en (Maria Garcia, 2008).

La trayectoria a seguir por los robots es una línea recta. Cuando el robot encuentre obstáculos este debe rodearlos siguiendo un algoritmo RRT hasta retornar a la trayectoria inicial, es decir una navegación libre de mapa, donde el robot recurre a la información del entorno proporcionada por

los sensores de navegación. La transmisión y adquisición de datos se realizará mediante comunicación inalámbrica WiFi.

Finalmente se realizarán pruebas de desempeño de los algoritmos desarrollados basados en una metodología experimental en el entorno estructurado. En estas pruebas se busca medir la velocidad de respuesta del sistema ante la presencia de obstáculos, el nivel de precisión referente a la localización y distancia entre robots. Dichos resultados serán analizados y de acuerdo a su relevancia podrían ser presentados en un documento científico.

1.4 Objetivos

1.4.1 Objetivo General

Diseñar e implementar algoritmos con robótica cooperativa para la clasificación de objetos de color en un entorno estructurado.

1.4.2 Objetivos Específicos

- Establecer algoritmos de trabajo cooperativo para la clasificación de objetos de colores utilizando un sistema multi-robot.
- Determinar las características básicas de los robots terrestres para el funcionamiento cooperativo en un entorno estructurado.
- Formular algoritmos para el sistema de navegación de acuerdo a los parámetros obtenidos por medio de sensores en un entorno estructurado.

1.5 Descripción del proyecto de investigación

El proyecto de titulación da soporte al proyecto ESPE-PIC-2016-009 denominado “SmellRobSense” dirigido por la Dra. Ana Guamán y el Dr. Andrés Arcentales, en la fase de desarrollo del entorno de comunicación y localización geográfica de los robots.

El objetivo del proyecto es la detección de TNT y Pólvora de Base Doble en un entorno controlado utilizando sensores químicos y espectrometría de movilidad de iones embebidos en robots móviles utilizando robótica cooperativa.

El objetivo del proyecto de titulación consiste en la elaboración de algoritmos de robótica cooperativa para la detección y clasificación de objetos de color de acuerdo al método SAF-T-DATA de J.T. Baker, con el fin de emular el proceso de trasladar a una área segura sustancias químicas en un entorno estructurado.

El sistema de robots está compuesto de cinco robots en configuración diferencial. El robot guía está equipado con una cámara y un sensor ultrasónico proporcionando información para la detección y localización de objetos de color en el entorno. Los demás robots constan de brazos electromecánicos y de sensores para la geolocalización.

Previo al diseño de los robots es necesario realizar la caracterización de los sensores respecto a la distancia de detección y al ángulo de medición, para conocer el funcionamiento de los mismos. Se efectuó diversas pruebas dentro del escenario estructurado para determinar las curvas características de comportamiento de los sensores. En base a los datos obtenidos se desarrollan los algoritmos de detección y clasificación de objetos de color en el escenario estructurado.

Los robots se comunican entre sí mediante una red inalámbrica WIFI basada en el estándar IEEE 802.11. Para la localización de los robots en el escenario se implementa algoritmos basados en el parámetro RSSI de la señal WiFi, emita por los puntos de acceso (AP) ubicados en puntos estratégicos del escenario de pruebas.

Obtenidos los datos de localización de los objetos de color y los robots dentro del escenario estructurado, es posible generar trayectorias aleatorias mediante la implementación del algoritmo RRT (Rapidly-exploring random tree). Para el seguimiento de trayectoria se implementa el algoritmo Pure pursuit (persecución pura), debido a su sencillez y buenos resultados.

Para el desarrollo de este proyecto se definen un conjunto de actividades enmarcadas en capítulos que se muestran a continuación.

CAPITULO II

ESTADO DEL ARTE

2.1 Introducción

En años anteriores el desarrollo tecnológico estaba enfocado en la mejora del robot móvil de forma individual, motivado por la necesidad de solventar tareas complejas en el mundo real (Soria, Carelli, Kelly, & Zannatha, 2004). Como resultado se obtuvo robots individuales sofisticados, altamente costosos e incapacitados para la realización de más tareas que su propósito original (Inglett & Rodríguez-Seda, 2017).

En la actualidad la robótica cooperativa ha experimentado un gran progreso debido a que permite desarrollar aplicaciones de mayor complejidad, reduciendo tiempo de ejecución y costo de implementación, mediante el uso de varios robots de diseño sencillo que trabajan juntos para ejecutar una tarea (Apolo Peñaloza & Velasco Barrera, 2013; Lara, Urrea, & Barbosa, 2015).

De entre las principales aplicaciones de la robótica cooperativa el transporte de objetos o material peligroso (Apolo Peñaloza & Velasco Barrera, 2013) es importante, debido a las condiciones de riesgo presentes en ambientes que manejan sustancias químicas.

Para evitar complicaciones en dichos procesos, se desarrollan normas internacionales para el orden, limpieza, señalización, clasificación y almacenamiento de sustancias químicas (Gonzalo Rafael Rojo Aguirre, 2014). El sistema SAF-T-DATA de J.T. Baker destaca por manejar un método codificado en colores para organizar las sustancias químicas de acuerdo a sus propiedades, almacenando juntas sustancias de similar color (Colmenares, 2014).

Varios trabajos (Chen et al., 2016; Danelljan et al., 2014; Lin et al., 2015; Possegger & Mauthner, 2015) presentan métodos de detección de objetos basados en el color y/o forma, en los mismos la utilización de cámaras permite al robot adquirir percepción del ambiente obteniendo así información para la detección de objetos (González Jiménez & Baturone, 2015).

Las WSNs proporcionan información para la ubicación de objetos tanto en exteriores como interiores, por medio de algoritmos que usan la medida de señal recibida (RSS) para determinar la posición (Dong & Xu, 2014). La localización en base a **RSS presenta variabilidad** en la señal produciendo errores y falta de precisión en la posición estimada. Sin embargo, se sigue utilizando métodos de localización basados en RSS debido a la facilidad con la que se obtiene las mediciones RSS (Dong & Xu, 2014; Hoyo Mondeja, 2012).

Los métodos de detección y localización de objetos son implementados en robots móviles dotados con capacidad de navegación determinada dentro de un escenario estructurado.

2.2 Robótica Cooperativa Móvil

La robótica cooperativa busca diseñar sistemas multi-robot que trabajen conjuntamente para cumplir objetivos comunes (Apolo Peñaloza & Velasco Barrera, 2013; Murray, s. f.). Los sistemas multi-robot son utilizados para resolver tareas que resultan física y computacionalmente complejas para un solo robot especializado (Morales et al., 2014). Trabajar con múltiples robots brinda la posibilidad de completar tareas en menor tiempo, además los sistemas multi-robot presentan: sencillez en diseño y control. Esto los hace menos costosos que un robot especializado (Morales et al., 2014), con configuraciones flexibles y con funcionalidad diversa permitiéndoles adaptarse a diversos entornos (Inglett & Rodríguez-Seda, 2017).

La investigación en el área de robótica cooperativa creció sustancialmente con el desarrollo de sistemas inalámbricos de comunicaciones, así como nuevas tecnologías de hardware y software (Murray, s. f.). Existen numerosas aplicaciones para la robótica cooperativa que van desde transporte de objetos (Inglett & Rodríguez-Seda, 2017), formación de vuelo de UAVs (vehículo aéreo no tripulado) (How, King, & Kuwata, 2004), formación de navegación de barcos (Aranda et al., s. f.), formación de robots terrestres (Figura 2) (Robert, Toibero, Vassallo, & Carelli, 2011; Soria et al., 2004), bancos de pruebas siendo el más conocido RoboCup (Chicala et al., 2006), tareas de exploración y mapeo (Morales et al., 2014), ensamblaje industrial (Lara et al., 2015), entre otras.

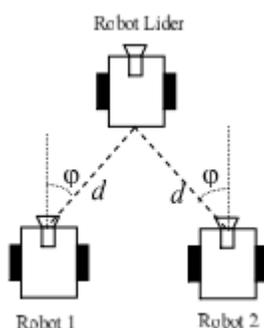


Figura 2. Formación triángulo de robots terrestres

Fuente: (Soria et al., 2004)

El sistema multi-robot cooperativo no necesariamente colabora entre sí para la realización de tareas. La robótica cooperativa para resolver una tarea se basa en el concepto de inteligencia cooperativa, es decir al existir varios robots se pueden dividir las tareas, y en caso de fallo de uno otro toma su lugar haciendo del sistema robusto y confiable (Apolo Peñaloza & Velasco Barrera, 2013; Inglett & Rodríguez-Seda, 2017). Para la formación de robots se pueden distinguir varias filosofías de control no excluyentes, siendo posible su combinación (Aranda et al., s. f.; De La

Cruz, Carelli, & Gava, 2006; Robert et al., 2011). El seguimiento al líder define una trayectoria donde los demás robots deben seguir a un robot principal, no obstante la tecnología utilizada es susceptible a perturbaciones.

- a. Control fundamentado en comportamientos de animales.
- b. Técnicas virtuales: se crea un líder virtual cuya trayectoria funciona como guía para los robots, es computacionalmente complejo.

La cooperación y coordinación entre robots aumenta las posibilidades de alcance y movilidad del sistema, donde si se aplica una o la combinación de ambas arquitecturas de control (centralizada y/o distribuida) se permite dotar de comportamiento inteligente a los robots.

2.2.1 Arquitecturas de control en robótica cooperativa

En un sistema multi-robot cooperativo todos los individuos están coordinados y comunicados en base a una arquitectura de control que puede ser: centralizada, distribuida o ambas. Dependiendo de la arquitectura de control aplicada cada individuo del sistema multi-robot tendrá inteligencia, capacidad de comunicación, coordinación y cooperación.

El comportamiento cooperativo de un sistema de múltiples robots permite resolver tareas que no se logran fácilmente con un robot. A medida que el número de individuos aumenta en el sistema multi-robot, el control del sistema se vuelve más complejo (Soria et al., 2004).

2.2.1.1 Control centralizado

El control centralizado en robótica cooperativa está constituido de una unidad central de la cual dependen los demás robots del sistema. La unidad central se encarga de las funciones de decisión

y planificación, brindando soluciones óptimas ya que posee información referente al entorno, además de mantener al sistema totalmente integrado. Los robots del sistema presentan diseños sencillos en los cuales se tiene una mínima cantidad de sensores para la adquisición de datos, un actuador para el funcionamiento del control, y medios de comunicación para el intercambio de datos con la unidad central (Apolo Peñaloza & Velasco Barrera, 2013; De La Cruz et al., 2006; Soria et al., 2004).

Al depender de una sola unidad de control central el sistema puede ser más preciso al encargarse de la resolución de problemas y la toma de decisiones, ya que almacena toda la información de los robots del sistema (Inglett & Rodríguez-Seda, 2017). La desventaja de esta arquitectura de control radica en que si existe algún fallo en la unidad de control todo el sistema se ve afectado, además que la unidad central de control requiere una gran cantidad de almacenamiento y procesamiento (De La Cruz et al., 2006; How et al., 2004).

En (Chicala et al., 2006) se aplica un control centralizado para un sistema de fútbol robótico como se visualiza en la Figura 3, en el cual utilizan una cámara cenital como único sensor, mediante el cual se realiza el reconocimiento y seguimiento de los objetos.



Figura 3. Sistema de fútbol robótico vista de la cámara cenital
Fuente: (Chicala et al., 2006)

2.2.1.2 Control distribuido o descentralizado

La arquitectura de control distribuido otorga a cada robot del sistema la capacidad de planificar y coordinar sus acciones sin la intervención de una unidad central de control. Los robots del sistema están provistos de una red de sensores para la adquisición de datos, un controlador propio que le permite resolver problemas basándose en los datos adquiridos del medio, y tecnologías de comunicación para el intercambio de datos entre los robots del sistema. Si existe una unidad central de control, esta se limita a la transmisión de información entre los robots y distribución de actividades del sistema (De La Cruz et al., 2006; Soria et al., 2004).

Al no depender de una unidad de control central cada robot del sistema es responsable de coordinar y planificar sus acciones, el sistema se vuelve más dinámico, confiable, robusto y tolerante a fallos. Si presenta errores el funcionamiento de un robot, los demás robots no se ven afectados y el sistema no se detiene.

Esta arquitectura de control presenta complejidad al desarrollar los algoritmos debido a la coordinación de las múltiples acciones a realizar por los robots sin que interfieran con los demás (Apolo Peñaloza & Velasco Barrera, 2013; Inglett & Rodríguez-Seda, 2017).

A diferencia de la arquitectura centralizada, esta no requiere de gran capacidad de almacenamiento y computo por parte de los robots, debido a la autonomía que presenta cada robot del sistema evitando los problemas presentados en la arquitectura antes mencionada. En (Acevedo, Arrue, Maza, & Ollero, 2011) se implementa un control distribuido para vigilancia de un perímetro con un sistema multi-robot cooperativo, en cual los robots intercambian información de la distancia vigilada recorrida.



Figura 4. Robots en prueba de vigilancia de un perímetro
Fuente: (Acevedo et al., 2011)

2.3 Visión artificial aplicada a robótica móvil

La robótica móvil es un área de suma importancia debido a sus diversas aplicaciones en: exploración y mapeo del entorno (Morales et al., 2014), vigilancia (Acevedo et al., 2011), trabajo en ambientes industriales (Lara et al., 2015), entre otras. El uso de visión artificial potencia las

aplicaciones de robótica móvil, ya que un solo sensor (cámara) puede proveer de información relevante del entorno al robot incrementando su autonomía. La visión artificial comprende tanto el reconocimiento como la determinación de posición de los objetos en una imagen. Un sistema de visión artificial debe realizar dos funciones principales: captar la información del entorno por medio de una imagen, y la extracción de la información para un posterior análisis (Efrain Ernesto et al., 2015; Etxeberria, 2010).

2.3.1 Proceso de formación de imágenes de la cámara

En robótica móvil es necesario que los robots perciban el entorno en el que se encuentran, siendo la visión una fuente principal de información. La implementación de una o varias cámaras en robots aporta mayor autonomía en los mismos, además las cámaras presentan independencia de dispositivos mecánicos y su costo es reducido (Etxeberria, 2010).

Existen diversos modelos para explicar tanto el funcionamiento de una cámara como la formación de imágenes, pin-hole es el modelo más simple y sirve de base para el resto de los modelos.

2.3.1.1 Modelo de cámara pin-hole

El modelo de pin-hole se basa en una cámara ideal y libre de perturbaciones, al ser un modelo simple requiere de menos parámetros para su representación. El modelo pin-hole describe la formación de una imagen, como una proyección 2D de una escena 3D.

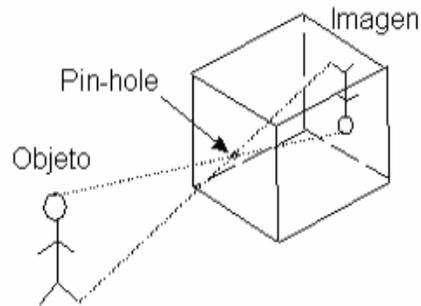


Figura 5. Cámara pin-hole
Fuente: (Ricolfe Viala, 2006)

En la Figura 5 se presenta el funcionamiento del modelo de pin-hole mismo que describe un sistema simple para formar una imagen, se representa como una caja cerrada con un orificio infinitesimalmente pequeño en uno de los lados (primera pantalla), a través de este orificio pasan los rayos de luz emitidos de la primera pantalla al lado paralelo del orificio (segunda pantalla) formando una imagen invertida (González & Cabrera Gámez, 2003; Ricolfe Viala, 2006).

El orificio central por el que pasan los rayos de luz emitidos desde el exterior de la caja se denomina foco, los rayos de luz tras haber pasado el foco llegan al plano donde se proyectara la imagen con una distancia f del orificio central como se observa en la Figura 6.

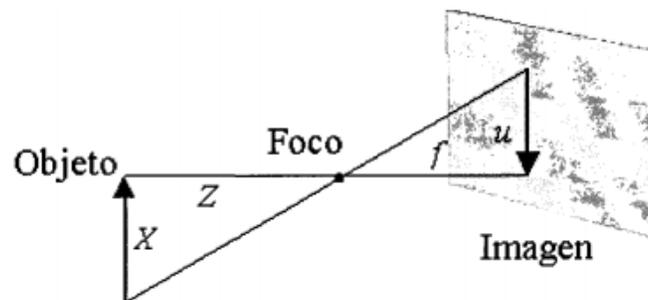


Figura 6. Componentes del modelo pin-hole
Fuente: (González & Cabrera Gámez, 2003)

2.3.2 Modelo de color

El modelo de color o espacio de color es un modelo matemático abstracto que permite representar los colores en números para uso computacional. Las características del modelo de color son usadas para seguimiento visual debido a que ofrece robustez ante cambios de escala y rápido procesamiento. El uso de las características de color presenta varios problemas como cambios en la constancia de color debido a la iluminación, y errores debido a la cámara (Etxeberria, 2010; Vezhnevets, Sazonov, & Andreeva, 2003).

La iluminación del ambiente es una consideración limitante en los algoritmos de un sistema de visión artificial. Una apropiada iluminación ayuda a obtener resultados precisos en el reconocimiento de objetos, en contraste una iluminación pobre entorpece el reconocimiento (Chicala et al., 2006).

2.3.2.1 Modelo RGB

RGB es un modelo de color basado en síntesis aditiva, es decir la representación de un color es la combinación de los colores primarios rojo, verde y azul. A pesar de ser uno de los espacios de color más utilizados para procesar y almacenar imágenes, no es una opción aplicable en los algoritmos de reconocimiento basados en color debido a la mezcla de datos de crominancia y luminancia (Etxeberria, 2010; Gonzales & Woods, 2007; Vezhnevets et al., 2003).

El modelo RGB está representado gráficamente en el sistema de coordenadas cartesianas, el subespacio de color forma un cubo como se visualiza en la Figura 7 donde los ejes principales corresponden a los colores primarios.

El color negro se encuentra en el origen y color blanco en el vértice más distante al origen, la escala de grises se extiende a lo largo de la diagonal que conecta estos dos vértices (Gonzales & Woods, 2007).

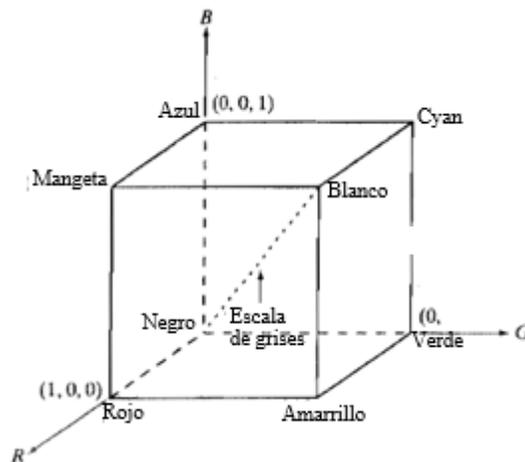


Figura 7. Cubo esquemático del modelo RGB

Fuente: (Gonzales & Woods, 2007)

A cada color primario se lo codifica con 8 bits en una escala de 0-255, donde 0 significa la ausencia del color en la combinación. La ecuación (1) representa la normalización de RGB. En superficies mate la normalización de RGB es invariante a los cambios de iluminación (Vezhnevets et al., 2003).

$$r = \frac{R}{R+G+B} \quad g = \frac{G}{R+G+B} \quad b = \frac{B}{R+G+B} \quad (1)$$

2.3.2.2 Modelo HSI (Hue, Saturation, Intensity), HSL (Hue, Saturation, Luminance), HSV (Hue, Saturation, Value)

Los modelos de color basados en características perceptuales de matiz (Hue), saturación (Saturation) y luminosidad o intensidad (Luminance) describen el espacio de color con valores intuitivos y similares a la visión humana (Vezhnevets et al., 2003). El tono o matiz especifica el color basándose en los colores primarios, la saturación mide la cantidad de color en base a la pureza y la luminosidad controla el brillo de la imagen dependiendo de la cantidad de luz (Etxeberria, 2010).

El modelo de color HSI y sus variantes derivan del modelo RGB mediante una transformación de coordenadas donde el sistema cartesiano de RGB se convierte en un sistema polar. En la Figura 8 se visualiza el modelo HSI de forma cilíndrica donde el matiz es el ángulo del sistema de coordenadas, la saturación es la distancia radial y la intensidad es la distancia en el eje perpendicular al plano (Gonzales & Woods, 2007).

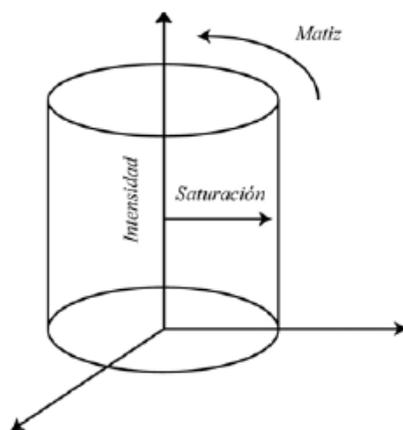


Figura 8. Esquemático del modelo HSI.
Fuente: (Gonzales & Woods, 2007)

El modelo HSI puede ser obtenido del modelo RGB mediante fórmulas geométricas, esto se deduce al rotar el cubo RGB sobre el vértice negro, teniendo en cuenta que la intensidad está representado por un eje vertical y los puntos de color son planos perpendiculares a este eje. La Figura 9 muestra el modelo HSI basado en triángulos y círculos (Gonzales & Woods, 2007; Vezhnevets et al., 2003).

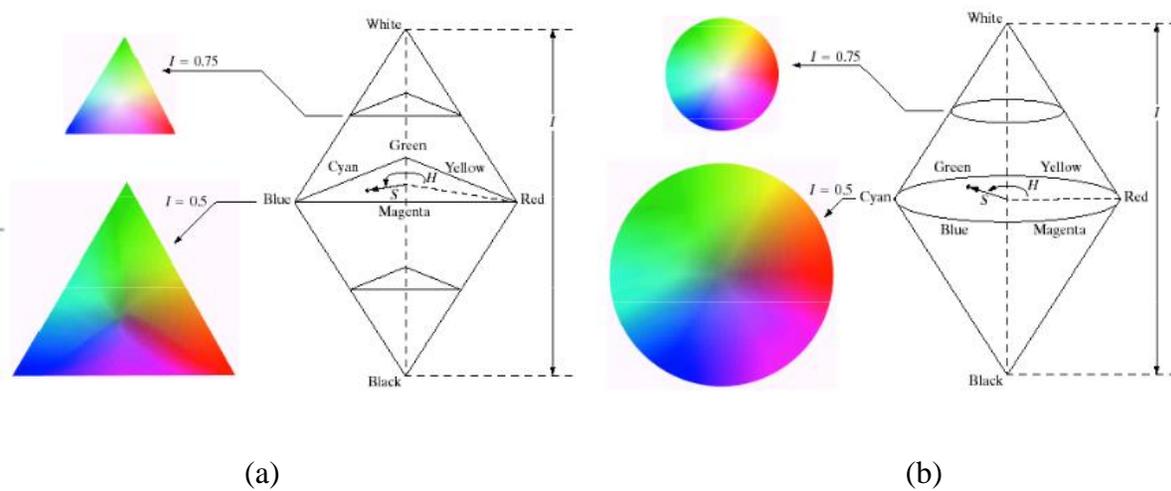


Figura 9. Modelo HSI representado en (a) triángulos. (b) círculos
Fuente: (Gonzales & Woods, 2007)

Los espacios de color HSL y HSV son variantes del espacio de color general HSI. Estos espacios de color han sido utilizados en trabajos de reconocimiento de color facial (Vezhnevets et al., 2003), con modelos que difieren por el uso de la característica de saturación. En el modelo HSL la pérdida de saturación hace a un color tender a gris, y en el modelo HSV la pérdida de saturación lleva al color tender a blanco (Etxeberria, 2010; Vezhnevets et al., 2003). La transformación del modelo RGB al modelo HSV se realiza mediante las siguientes ecuaciones (Vezhnevets et al., 2003):

$$H = \arccos \frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{((R-G)^2+(R-B)(G-B))}} \quad (2)$$

$$S = 1 - 3 \frac{\min(R,G,B)}{R+G+B} \quad (3)$$

$$V = \frac{1}{3}(R + G + B) \quad (4)$$

2.3.3 Procesamiento digital de imágenes

Una imagen puede representarse de manera bidimensional como $f(x,y)$, donde x y y son coordenadas en un plano y f es la intensidad o nivel de gris en ese punto. Al digitalizar las coordenadas y la magnitud de intensidad obtenemos la imagen digital en bits (0 y 1) (Jiménez Camacho, 2009; La Serna Palomino & Román Concha, 2009).

El procesamiento digital de imágenes hace referencia a las operaciones que se pueden efectuar en imágenes. Las etapas para realizar estas operaciones se visualizan en la Figura 10. De acuerdo al nivel de procesamiento computacional se puede hablar de tres niveles: bajo, medio y alto. En el nivel bajo o reprocesado, se encuentran los procesos para mejorar la imagen como reducción de ruido o cambio de contraste. El nivel medio está compuesto de: segmentación y descripción, que realizan la extracción de características de la imagen de entrada mediante procesos como reconocimiento de objetos. Finalmente en el nivel alto se interpreta y realiza operaciones de acuerdo a las características extraídas de la imágenes procesadas (Efrain Ernesto et al., 2015; Jiménez Camacho, 2009; La Serna Palomino & Román Concha, 2009).

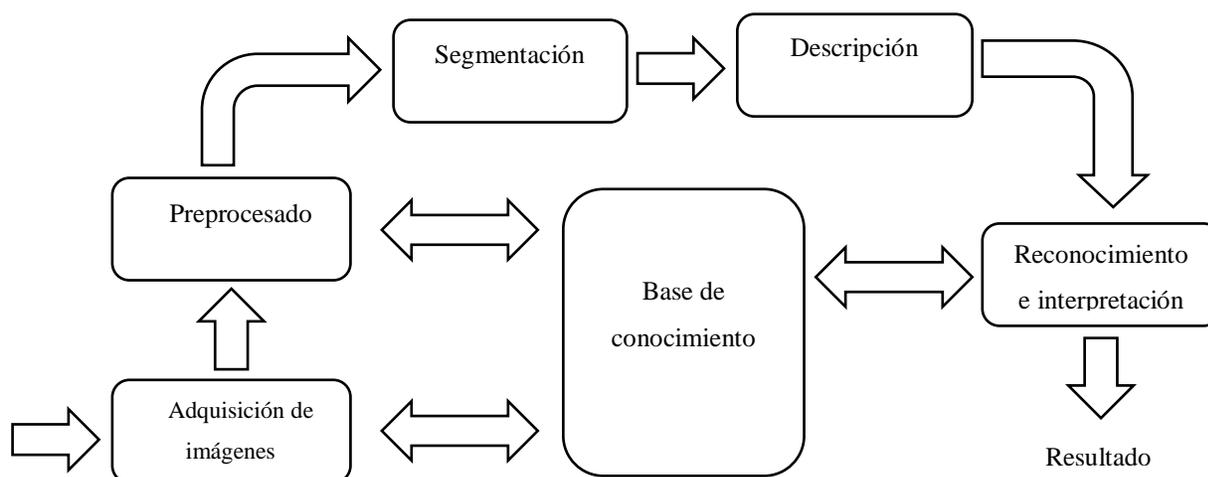


Figura 10. Ciclo del procesamiento digital de imágenes

Fuente: (La Serna Palomino & Román Concha, 2009)

2.3.3.1 Segmentación

La segmentación es una técnica utilizada para la extracción de información de una imagen, con el propósito de subdividir la imagen en partes constitutivas (Gonzales & Woods, 2007). El proceso de segmentación debe detenerse cuando todos los objetos de interés se hayan aislado, siendo una de las tareas más complicadas dentro de procesamiento de imágenes (La Serna Palomino & Román Concha, 2009). En imágenes monocromáticas la segmentación se basa en una de dos propiedades básicas de nivel de gris: discontinuidad y similitud (Gonzales & Woods, 2007).

En discontinuidad el método se fundamenta en los cambios bruscos del nivel de gris, permitiendo la detección de puntos aislados, líneas y bordes en una imagen. Los métodos de segmentación basados en discontinuidad son aplicables tanto a imágenes estáticas como dinámicas. En similitud las áreas de interés de una imagen se detectan en base a características de

homogeneidad como: posición, textura, nivel de intensidad, entre otros (La Serna Palomino & Román Concha, 2009).

2.3.3.2 Métodos de segmentación

Los métodos que a continuación se describen dependen de diversas técnicas de procesamiento de imágenes como son: detección de discontinuidades y detección de similitud en imágenes.

2.3.3.3 Métodos basados en discontinuidad

Los métodos de detección basados en cambios bruscos del nivel de gris aplican técnicas de derivación mediante la aplicación de filtro espacial que engloban la detección: de puntos aislados, líneas y bordes en una imagen. La forma más común de detección de discontinuidades es aplicar una máscara a la imagen (Gonzales & Woods, 2007; Jiménez Camacho, 2009).

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Figura 11. Máscara de 3x3
Fuente: (Jiménez Camacho, 2009)

La máscara de la Figura 11 se multiplica por los niveles de gris de una región de la imagen, y la sumatoria del producto es el resultado del pixel central de la región como se observa en la ecuación

(5), donde w_i corresponde a la ponderación de la máscara y z_i es el nivel de gris del píxel (Jiménez Camacho, 2009).

$$R = \sum_{i=1}^9 w_i z_i \quad (5)$$

2.3.3.3.1 Detección de puntos aislados

Para la detección de puntos aislados en una imagen se aplica una máscara como la Figura 11, donde se sabe que se ha detectado un punto en la posición central de la máscara si,

$$|R| > T \quad (6)$$

donde T es un umbral positivo, y R es el valor obtenido de la sumatoria de productos de los coeficientes w_i y los niveles de gris z_i de la región donde se aplica la máscara (Gonzales & Woods, 2007), ver ecuación (5). La máscara más utilizada es:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (7)$$

2.3.3.3.2 Detección de líneas

Se puede definir una línea como una cadena de píxeles, donde dos píxeles sucesivos se encuentran conectados en un entorno de 3x3. Al examinar el entorno existe la posibilidad de que cada píxel puede conectarse con alguno de sus ocho píxeles vecinos, y la línea a formar siga cuatro direcciones posibles: horizontal, vertical, $+45^\circ$, y -45° . Se puede usar las siguientes máscaras en todos los píxeles de la imagen para detectar las posibles direcciones, ver Figura 12.

$$\begin{array}{cccc}
 \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} & \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} & \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} & \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \\
 \text{(a)} & \text{(b)} & \text{(c)} & \text{(d)}
 \end{array}$$

Figura 12. Máscara para detección de líneas (a) horizontal, (b) vertical, (c) +45°, (d) -45°

Fuente: (Gonzales & Woods, 2007)

Se puede determinar los píxeles donde la respuesta es máxima al aplicar las máscaras de la Figura 12 sobre todos los píxeles de la imagen. Sean R_1 , R_2 , R_3 , y R_4 las respuestas al aplicar las máscaras en el pixel central. Para interpretar el tipo de línea detectada se utiliza el criterio $|R_i| > |R_j|$ para todo $j \neq i$, donde el punto i está asociado a la dirección posible.

2.3.3.3 Detección de bordes

Los bordes son la frontera entre dos regiones de diferente nivel de gris en una imagen. En la detección de bordes se tiene en cuenta los cambios abruptos de nivel de gris de los píxeles vecinos y se suprime las regiones con valores de gris invariables (La Serna Palomino & Román Concha, 2009).

En la mayor parte de técnicas de detección de bordes el cálculo se realiza mediante un operador de derivación, el operador de primer orden gradiente o el operador de segundo orden Laplaciano. En la búsqueda de cambios abruptos de nivel de gris para la detección de bordes, un problema es la detección de ruido (Jiménez Camacho, 2009; La Serna Palomino & Román Concha, 2009).

El uso del operador **gradiente** es una técnica común para detectar bordes, el gradiente de una imagen es una función $f(x, y)$ en (x, y) definiéndose como el vector:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (8)$$

La magnitud del vector gradiente es un valor sustancial para detectar bordes, ver (9), el operador gradiente indica la dirección de máxima variación de f en el punto (x, y) .

$$\nabla f = \sqrt{G_x^2 + G_y^2} \quad (9)$$

Para conseguir un procesamiento más rápido es común aproximar el gradiente a valores absolutos, acarreando un error despreciable, ecuación (10).

$$\nabla f \approx |G_x| + |G_y| \quad (10)$$

En base a las ecuaciones (8) y (9) se concluye que el cálculo del gradiente en una imagen está basado en derivadas parciales en cada posición de pixel. Existen varias máscaras como son Roberts, Sobel y Prewit, para más detalles revisar (Gonzales & Woods, 2007).

2.3.3.4 Métodos basados en similitud

2.3.3.4.1 Umbralización

La umbralización proporciona una manera de separar las regiones de la imagen que corresponden a objetos de las del fondo mediante binarización. Además la umbralización es útil para segmentar las regiones de una imagen que se encuentren dentro de un determinado rango de nivel de gris (Gonzales & Woods, 2007; Jiménez Camacho, 2009).

Para aplicaciones simples la segmentación se establece por el umbral de intensidad, todos los píxeles de la imagen se comparan con este valor. Si la intensidad (nivel de gris) del píxel es superior al umbral, el píxel se coloca en blanco y si el píxel es inferior que el umbral se coloca en negro (o viceversa). En aplicaciones complejas se pueden determinar múltiples umbrales, de tal manera una agrupación de píxeles entre un rango de valores de intensidad se coloca en blanco y los demás en negro (La Serna Palomino & Román Concha, 2009).

Sea $f(x, y)$ la imagen original, $g(x, y)$ la imagen umbralizada y $U(0 < U < 255)$ el valor del umbral, la ecuación (11) define la operación de umbralización como:

$$\begin{aligned} g(x, y) &= 255, & f(x, y) &> U \\ g(x, y) &= 0, & f(x, y) &\leq U \end{aligned} \quad (11)$$

Para saber si una imagen puede ser segmentada mediante umbralización se analiza el histograma de intensidad de la imagen. Para ser posible la segmentación la intensidad de los píxeles de los objetos debe ser diferente de los del fondo como se observa en la Figura 13 (b), en la Figura 13 (c) se muestra el resultado del proceso de umbralización (Gonzales & Woods, 2007).

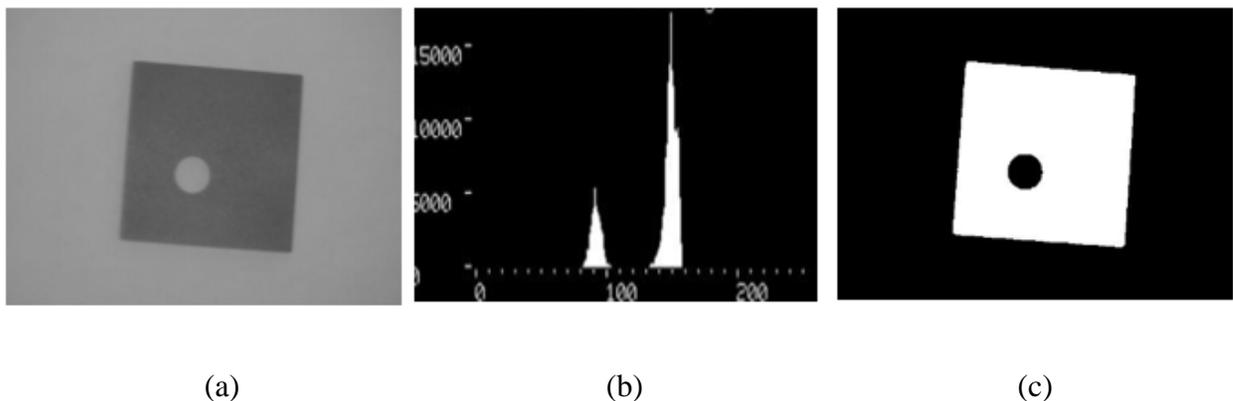


Figura 13. (a) Imagen original, (b) histograma, (c) imagen resultante

Fuente: (Gonzales & Woods, 2007)

2.3.4 Calibración de cámara

El procedimiento de calibración de la cámara es necesario para obtener información cuantitativa del entorno 3D a partir de imágenes 2D del mismo. El proceso de calibración es fundamental cuando la información obtenida del entorno se usa para medición. Una calibración precisa permite obtener distancias exactas del mundo real a partir de imágenes 2D. La obtención de parámetros intrínsecos y extrínsecos es el resultado del proceso de calibración, donde dichos parámetros no son exactos debido a imprecisiones que alteran el proceso (Ricolfe Viala, 2006; Viala, José, & Salmerón, 2008).

Los parámetros intrínsecos describen la geometría interna de una cámara y las características ópticas del sensor, los más representativos son: distancias focales, centro óptico, distorsión del lente (radial y tangencial). Los parámetros extrínsecos definen la posición y orientación respecto al mundo real, dando orientación externa a la cámara (Jiménez Camacho, 2009; Ricolfe Viala, 2006).

Existen diversos métodos para la calibración de cámaras de los cuales destacan los siguientes:

2.3.4.1 Método de resolución

Se clasifica en métodos lineales y no lineales. Los métodos lineales se basan en sistemas de ecuaciones de mínimos cuadrados donde los resultados obtenidos son aproximados, siendo estos métodos de fácil implementación y de rápida ejecución (Ricolfe Viala, 2006; Viala et al., 2008).

El resultado obtenido del proceso de calibración mediante el método lineal se usa para iniciar el cálculo no lineal de los parámetros, posteriormente se usan métodos iterativos que mejoran la

estimación de estos parámetros. Este proceso de calibración sirve para un modelo de cámara complejo que incluya las distorsiones que produce la cámara (Ricolfe Viala, 2006).

2.3.4.2 Según el resultado

Se basa en el resultado que se obtiene del proceso de calibración, considerando la existencia de la calibración explícita e implícita para una cámara. El método de calibración explícito es apropiado para la corrección de distorsiones en la imagen debido a que obtiene todos los parámetros: explícitos como intrínsecos del modelo de la cámara (Ricolfe Viala, 2006).

La calibración implícita genera matrices de transformación que contienen los parámetros siendo difícil su extracción, a pesar de no conocer el valor de dichos parámetros los resultados de este proceso pueden utilizarse para medición 3D y generación de coordenadas en una imagen (Ricolfe Viala, 2006).

2.3.4.3 Según la plantilla

Basándose en las características de la plantilla usada para el procedimiento de calibración, se tiene métodos que usan plantillas 3D, 2D, 1D o no utilizan. En el proceso de calibración con plantillas 3D, ver Figura 14 (a), una sola imagen es necesaria, dicha plantilla requiere de una elaboración compleja (Ricolfe Viala, 2006; Viala et al., 2008).

Al utilizar plantillas 2D ver Figura 14 (b), se necesita tomar diversas imágenes de la plantilla desde diferentes posiciones y orientaciones. La plantilla no requiere de una compleja elaboración

haciendo de este método versátil. Las plantillas 1D se utilizan en la calibración de sistemas con múltiples cámaras (Ricolfe Viala, 2006).

La autocalibración es una técnica que no utiliza plantillas, se basa en la correspondencia de puntos de diversas imágenes captadas de una escena estática, como mínimo tres imágenes. Los parámetros extrínsecos e intrínsecos son obtenidos mediante correspondencias y transformaciones geométricas (Jiménez Camacho, 2009).

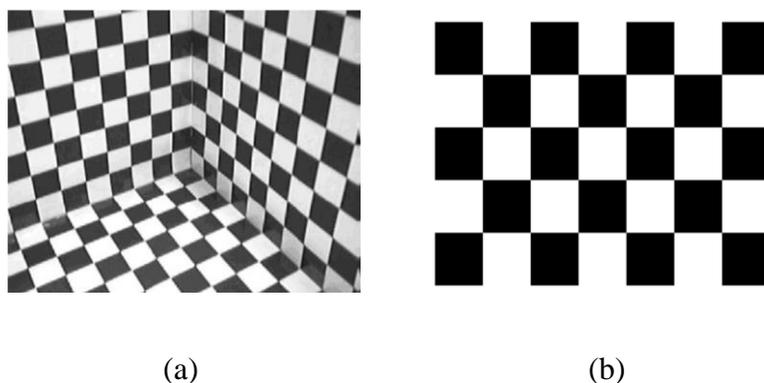


Figura 14. Plantilla (a) 3D, (b) 2D para el proceso de calibración
Fuente: (Ricolfe Viala, 2006)

2.3.5 Cámaras IP

Una cámara IP (Internet Protocol) es un dispositivo que captura y transmite señales de audio/video en tiempo real a través de una red IP sea esta alámbrica o inalámbrica. Los datos de audio/video pueden ser almacenados en ubicaciones remotas por razones de comodidad o seguridad, donde usuarios autorizados administran la información, y a su vez puede transmitirse a la red LAN o Internet (López Rodríguez Julio César, 2007; Mejía, 2015; Revista Negocios de Seguridad, 2007).

Es necesario la asignación de una dirección IP interna para la conexión a la red y para poder acceder a la información que transmiten las cámaras IP a través de un navegador, ver Figura 15.

Las cámaras IP operaran independiente de un computador para la captura y transmisión de imágenes, ya que cuentan con servidor web y sistema operativo propios (Revista Negocios de Seguridad, 2007)(López Rodríguez Julio César, 2007).

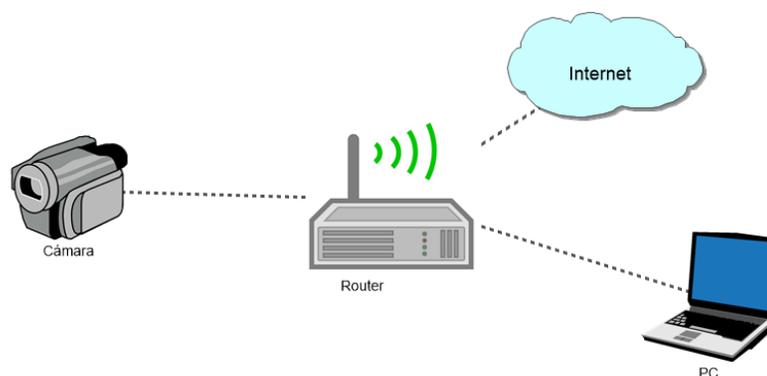


Figura 15. Conexión en red de una cámara IP

Fuente: (Revista Negocios de Seguridad, 2007)

Dentro de las características más relevantes de las cámaras IP se pueden citar (Revista Negocios de Seguridad, 2007):

- Accesibilidad remota
- Flexibilidad
- Escalabilidad
- Integración y funcionalidad actualizable

En el mercado existen diversos tipos de cámaras IP, para seleccionar la indicada se debe analizar el lugar de instalación y la aplicación que tendrá. A continuación se plantean recomendaciones al momento de elegir una cámara (Mejía, 2015)(López Rodríguez Julio César, 2007) .

- *Sensibilidad*: Intensidad de luz necesaria para el correcto funcionamiento de la cámara en ambientes de escasa iluminación. La unidad de medida es Lux, a mayor sensibilidad menor Lux.
- *Resolución*: Cantidad de píxeles horizontales y verticales que posea la cámara. A mayor número de píxeles, mejor resolución.
- *Compensación de contraluz*: Cuando la escena tiene demasiada luminosidad o muy poca, la compensación de contraluz evita que las imágenes sean excesivamente claras o muy oscuras.
- *Ajuste de blancos*: Equilibra los niveles RGB, de manera que la parte más brillante de la imagen aparezca como color blanco y la menos brillante como color negro.
- *Tamaño de la cámara*: Se debe considerar si se necesita una cámara de tamaño visible o de tamaño imperceptible, dependiendo de la aplicación.
- *Duración de la batería*: Es un aspecto importante para el rendimiento de la aplicación

2.4 Métodos de estimación de posición

En robótica móvil los sistemas de navegación autónoma son de gran importancia, porque brindan al robot la capacidad de reaccionar ante circunstancias imprevistas en el entorno (Etxeberria, 2010). Los robots móviles pueden contar con varios sistemas sensoriales que interactúan entre sí y proporcionan información del entorno al robot para realizar tareas como: estimación de posición, evasión de obstáculos y planificación de trayectoria (González Jiménez & Baturone, 2015).

Se considera la existencia de dos métodos para la estimación de posición de un robot, los estimadores basados en la percepción del entorno y los estimadores basados en medidas explícitas. (Etxeberria, 2010; García-Alegre, Ribeiro, Guinea, & Sandoval, 2014).

2.4.1 Métodos de estimación basados en la percepción del entorno

Los métodos de estimación basados en percepción del entorno utilizan un sistema sensorial para obtener información del medio y ser capaces de determinar la localización del robot.

El sistema sensorial utilizado para la navegación autónoma de un robot se divide en: sensores activos y pasivos (Etxeberria, 2010; García-Alegre et al., 2014; González Jiménez & Baturone, 2015).

Los sensores activos emiten algún tipo de energía obteniendo directamente medidas de distancia del medio, los sensores más conocidos son infrarrojos, laser y ultrasónicos. Su principio de funcionamiento es el “tiempo de vuelo”, que consiste en la medición del tiempo transcurrido entre el envío de la señal y recepción de eco de la misma. La distancia entre el objeto y el emisor se estima mediante la velocidad de propagación de la señal y la medida del tiempo de vuelo (García-Alegre et al., 2014; González Jiménez & Baturone, 2015).

Los sensores activos presentan inconvenientes cuando varios robots se encuentran cerca, debido a las interferencias entre las señales ocasionando estimaciones incorrectas. Los sensores ultrasónicos presentan simplicidad y bajo costo, pero son vulnerables al medio y a reflexiones especulares. Por otro lado los sensores laser demuestran precisión y fiabilidad en medición aunque a un elevado costo de implementación (Etxeberria, 2010).

Los sensores pasivos a diferencia de los activos captan la energía del medio, las cámaras de video pertenecen a este tipo de sensores. Las cámaras son sensores de uso frecuente en percepción debido a sus características destacables: alta resolución, gran velocidad de adquisición de datos, consumo bajo de energía, independencia de dispositivos mecánicos y bajo costo de implementación (Etxeberria, 2010; González Jiménez & Baturone, 2015). Sin embargo, presentan dependencia a las condiciones de iluminación y alta complejidad computacional, ver (Chicala et al., 2006).

Se presentan en dos categorías las técnicas para la estimación de posición de un robot mediante el uso de cámaras (Etxeberria, 2010; González Jiménez & Baturone, 2015). La primera comprende todas las técnicas basadas en la obtención de distancia de la cámara respecto a los objetos de la imagen, de este tipo es el sistema propuesto en (Jiménez Camacho, 2009).

La segunda categoría comprende la estimación a través del emparejamiento de características entre imágenes extraídas con el modelo conocido como se ve en (Possegger & Mauthner, 2015).

2.4.2 Métodos de estimación explícita

Estos métodos no necesitan capturar información del entorno para estimar la posición, se basan en diversas medidas internas que determinan la trayectoria recorrida del robot. Se subdividen en métodos de estimación basados en medidas internas y basados en estaciones de transmisión (Etxeberria, 2010; González Jiménez & Baturone, 2015).

2.4.2.1 Métodos de estimación basados en medidas internas

Este tipo de métodos trabaja con sistemas sensoriales integrados en el robot que permiten determinar su trayectoria recorrida a partir de diversas medidas internas, y así estimar la posición

del robot sin conocimiento del entorno. Dependiendo de las medidas a utilizar se distinguen dos subcategorías: sistemas odométricos y sistemas de navegación inercial (INS) (Etxeberria, 2010; González Jiménez & Baturone, 2015).

2.4.2.1.1 Sistemas odométricos

Los métodos basados en odometría estiman la posición y orientación de un robot según su desplazamiento en el entorno, mediante el cálculo del número de vueltas dadas por las ruedas. Los encoders o codificadores ópticos son los sensores que se encargan de la cuenta del número o fracción de vuelta de las ruedas, estos sensores son acoplados sobre los ejes del robot móvil (Etxeberria, 2010).

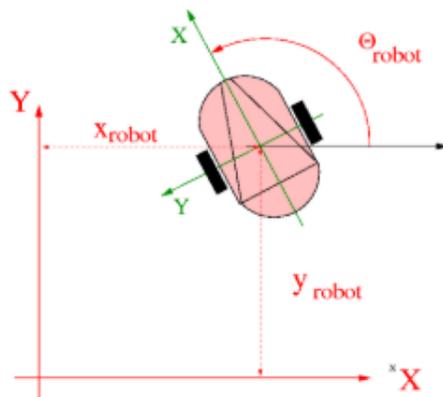


Figura 16. Esquemático de un robot móvil diferencial

Fuente: (Etxeberria, 2010)

La técnica de odometría es simple y de bajo costo, pero necesita una calibración frecuente debido al desgaste y desalineación de las ruedas. Siendo susceptible a errores que se van acumulando proporcionalmente según la distancia recorrida por el robot (García-Alegre et al., 2014; González Jiménez & Baturone, 2015).

2.4.2.1.2 Sistemas de navegación inercial (INS)

Este método estima la posición y orientación del robot basándose en medidas de aceleraciones obtenidas de acelerómetros y ángulos de orientación obtenidos de giroscopios. La velocidad se obtiene de la primera integral de las medidas de aceleración y la posición se obtiene de la segunda integración (Etxeberria, 2010; González Jiménez & Baturone, 2015).

Los sensores acelerómetro y giroscopio van acoplados al robot móvil mediante una base estabilizadora o simplemente fijos a él. Una de las características destacables de los sistemas de navegación inercial es que la interacción del robot con el piso no causa errores, obteniendo valores de estimación más fiables y precisos a diferencia del sistema odométrico (Etxeberria, 2010; González Jiménez & Baturone, 2015).

A pesar de sus ventajas sobre el sistema odométrico no se utiliza como único sistema sensorial para el robot móvil debido a la acumulación progresiva de errores de precisión, baja resolución del sensor, y factor de escala, siendo complementados con sistemas de posicionamiento absoluto (García-Alegre et al., 2014).

2.4.2.2 Métodos de estimación basados en estaciones de transmisión

Estos métodos de estimación de posición son conocidos también como sistemas de posicionamiento absoluto, debido a que proporcionan localización absoluta de un robot dentro de un área si poseer información del entorno. Están formados de un receptor y un emisor, el receptor es acoplado en el robot móvil y el transmisor o conjunto de transmisores pueden ser estáticos o móviles (Etxeberria, 2010; González Jiménez & Baturone, 2015).

Los métodos de estimación de posicionamiento que trabajan con estaciones transmisoras fijas son: trilateración y triangulación (Etxeberria, 2010).

2.4.2.2.1 Trilateración

Mediante este método se calcula la posición de un robot al medir la distancia existente entre él y varias posiciones de referencia denominadas balizas, ver Figura 17. Para determinar la posición del robot es necesario trabajar con tres balizas de distancias conocidas, la distancia entre el robot y una baliza se obtiene mediante técnicas de estimación de distancias (Etxeberria, 2010; Maria Garcia, 2008).

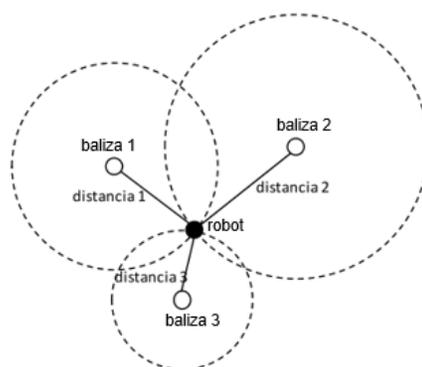


Figura 17. Método de trilateración

Fuente: (Maria Garcia, 2008)

2.4.2.2.2 Triangulación

El método de triangulación es similar al método de trilateración, debido al uso de ángulos en lugar de distancias para deducir la posición de un robot. Para estimar la posición se requiere de dos ángulos y la distancia entre dos balizas ver Figura 18, o dos distancias conocidas y un ángulo (Etxeberria, 2010; Maria Garcia, 2008).

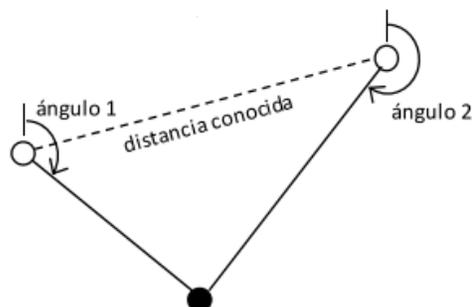


Figura 18. Método de triangulación

Fuente: (Maria Garcia, 2008)

2.4.2.2.3 GPS (Global Positioning System)

El método de estimación que trabajan con estaciones transmisoras móviles más destacado es el GPS (Global Positioning System). Funciona mediante una red de 24 satélites móviles que se desplazan alrededor de la tierra en seis órbitas distintas, ver Figura 19. Para estimar la posición el receptor capta la información de tres satélites como mínimo y calcula altitud, latitud y velocidad mediante triangulación (González Jiménez & Baturone, 2015; Maria Garcia, 2008).

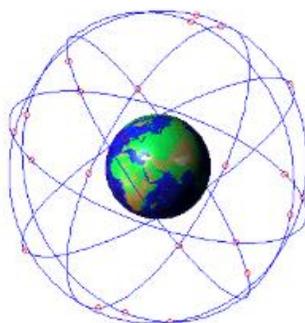


Figura 19. Distribución orbital de los satélites

Fuente: (Etxeberria, 2010)

La localización y seguimiento de objetos (robots móviles, material, etc.) o personas en interiores es de gran utilidad para diversas aplicaciones en ambientes industriales donde la seguridad de los operarios (Benasulin, Modesti, & Tanburi, s. f.) es un factor de suma importancia. De igual manera,

se lo utiliza en interiores o en edificios para determinar la posición en la que se encuentra un objeto o persona (Martin, Vinyals, Friedland, & Bajcsy, 2010).

La localización mediante sistemas de posicionamiento global (GPS) presentan una exactitud formidable en exteriores, y un bajo desempeño en interiores causado por la debilitación de la señal satelital al encontrarse con muros o estructuras en ambientes cerrados (Hoyo Mondeja, 2012). Por esta razón se busca alternativas para resolver el problema de localización en interiores, donde algunos métodos desarrollados utilizan tecnologías como WiFi (Hoyo Mondeja, 2012; Rugeles & Leon, 2010), bluetooth (Fernández, Rodas, Iglesia, & Escudero, s. f.; Rodríguez, Pece, & Escudero, 2014), y tecnologías móviles CDMA o GSM.

La tecnología WiFi basada en el estándar 802.11 es ampliamente difundida, estando presente en la mayor parte de dispositivos móviles y circuitos integrados de comunicación (Romero Macas, 2017). Además es el estándar común de comunicaciones inalámbricas en redes de área local (WLAN), por lo cual existe alta disponibilidad de dispositivos a precios reducidos, y con la ventaja que para realizar localización no necesita de hardware adicional siendo esta una de sus principales ventajas (Hoyo Mondeja, 2012; Xuesong, Wu, & Ming, 2008).

Los principales parámetros de la señal inalámbrica que pueden intervenir en la estimación de posición son: tiempo de llegada (ToA – time of arrival), diferencia de tiempos de llegada (TDoA – time difference of arrival), ángulo de llegada (AoA – angle of arrival) e intensidad de señal recibida (RSS – received signal strength); para más detalles ver (Hoyo Mondeja, 2012; Maria Garcia, 2008; Rugeles & Leon, 2010; Segura, 2010; Xuesong et al., 2008). Mediante el uso de estos parámetros se puede estimar la distancia entre emisor y receptor, y al aplicar cualquiera de

los métodos de estimación de posición basados en estaciones de transmisión se determina la localización del robot u objetos en un entorno cerrado.

2.5 Estimación de posición basándose en RSSI

Es común utilizar el parámetro RSSI (indicador de potencia de señal recibida) para localización en sistemas basados en redes inalámbricas WiFi. El método basado en RSSI necesita al menos de tres puntos de acceso (AP) para determinar la localización de un objeto/robot móvil. Utilizando la medición de la potencia de la señal recibida en la antena del receptor se estima la distancia de ubicación del transmisor y para calcular la posición del objeto en el entorno se aplica triangulación o fingerprints (huellas digitales) (Romero Macas, 2017; Segura, 2010).

La ecuación (12) de Friss para transmisión en el espacio libre determina que la potencia de la señal disminuye de manera cuadrática a medida que la distancia aumenta (Dong & Xu, 2014; Fernández et al., s. f.).

$$P_{RX} = P_{TX} G_{TX} G_{RX} \left(\frac{\lambda}{4\pi d} \right)^2 \quad (12)$$

Donde P_{RX} es la potencia del receptor, P_{TX} es la potencia transmitida, G_{TX} , G_{RX} son las ganancias de transmisor y receptor respectivamente, λ es la longitud de onda y d es la distancia entre el módulo emisor y el módulo receptor. RSSI [dBm] se calcula en base a P_{RX} y una potencia de referencia P_{ref} que generalmente toma el valor de 1mW de siguiente manera:

$$\text{RSSI} = 10 \log \frac{P_{RX}}{P_{ref}} \quad (13)$$

Se deduce de las ecuaciones (12) y (13) que a menor potencia recibida, menor será el valor de RSSI. Usando un modelo de atenuación simplificado la ecuación de la potencia de señal recibida (RSSI) puede ser:

$$\text{RSSI} = -10 n \log \frac{d}{d_0} + A_0 \quad (14)$$

Donde n es la constante de pérdida del medio ($n = 2$ espacio libre), d_0 es la distancia de referencia conocida ($d_0 = 1m$ generalmente) y A_0 es la pérdida de propagación a la distancia d_0 . De la ecuación (14) se despeja d de la siguiente manera:

$$d = 10^{\frac{\text{RSSI}-A_0}{10 n}} \quad (15)$$

Existen diversos algoritmos para determinar la posición de un objeto mediante el parámetro RSS, en los cuales se utilizan tanto métodos geométricos como estadísticos (Rugeles & Leon, 2010). En escenarios grandes de $500 m^2$ de área el error de distancia obtenido es $3 m$ (Dong & Xu, 2014) y en escenarios de $1 m^2$ de área (Rugeles & Leon, 2010) el error es de $20 cm$, se puede decir que en distancias cortas los errores son aproximadamente el 10% para una distancia máxima de $20 m$ (Maria Garcia, 2008); este porcentaje de error se debe a perturbaciones ambientales, variabilidad de la señal en función del fabricante de circuitos integrados, el nivel de carga de la batería, entre otros (Hoyo Mondeja, 2012).

Sin embargo se siguen aplicando los métodos basados en RSSI principalmente porque las mediciones se obtienen fácil y directamente de los circuitos integrados y sin necesidad de hardware adicional, además su precio es bajo al igual que el consumo de energía (Dong & Xu, 2014).

2.5.1 Triangulación

El método de triangulación funciona de manera similar al mencionado en la sección 2.4.2.2, utilizando conceptos geométricos para estimar la distancia al objeto (Etxeberria, 2010; Maria Garcia, 2008).

2.5.2 Fingerprints o reconocimiento de patrones

El método de Fingerprints (huellas digitales) es una técnica muy usada para localización en interiores, en comparación con el método de triangulación sus cálculos son reducidos (Hoyo Mondeja, 2012). Este método permite estimar la posición de un objeto comparando las características medidas de la señal respecto a una base de datos, por lo que requiere mayor tiempo de procesamiento y almacenamiento (Romero Macas, 2017).

Este método se realiza en dos fases: primera fase off-line de entrenamiento y segunda fase on-line de posición y pruebas.

2.5.2.1 Fase off-line

En esta fase el área de análisis es dividida en múltiples cuadros generando una rejilla, se realizan las mediciones del parámetro RSSI para cada punto de referencia desde uno o diversos puntos de acceso, generando una matriz con los datos de la posición del punto de referencia y el valor RSSI medido, que serán almacenados en una base de datos (Romero Macas, 2017; Rugeles & Leon, 2010). Con los datos almacenados se construye un mapa del área de análisis como se observa en la Figura 20, cuanto más particiones tenga el área mayor será la exactitud para la estimación de posición (Hoyo Mondeja, 2012).

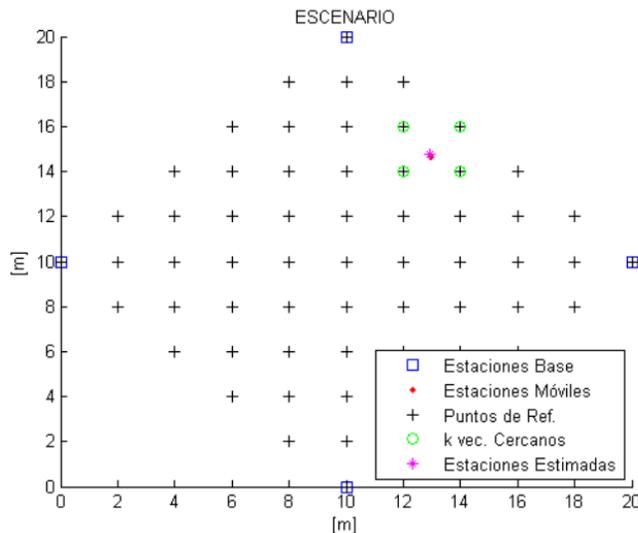


Figura 20. Método de localización Fingerprints
Fuente: (Rugeles & Leon, 2010)

2.5.2.2 Fase on-line

En la fase on-line se miden nuevos valores del parámetro RSSI en puntos desconocidos dentro del área de análisis y se infiere la localización a través de la mejor coincidencia entre los valores medidos y los registrados en la base de datos (Martin et al., 2010; Romero Macas, 2017).

El método de Fingerprints presenta un error reducido en comparación con el método de triangulación, sin embargo no permite la localización de objetos fuera del área de análisis (Hoyo Mondeja, 2012).

2.6 Planificación de Trayectoria

La robótica móvil está enfocada en navegación autónoma, es decir los robots móviles deben ser capaces de evitar colisiones con obstáculos y con otros robots mientras siguen una determinada

trayectoria desde una configuración inicial hasta otra final dentro de un entorno (Diego & Fernando, 2012; López et al., 2010).

Se puede describir tres problemas relacionados a navegación autónoma en robots móviles, primero la auto-localización donde el robot recurre a la información del sistema sensorial para estimar su posición respecto al entorno, segundo la determinación de la posición de los obstáculos y la posición a la que debe llegar en el entorno, y finalmente el tercer problema comprende la planificación de trayectoria a seguir (Álvarez Romero & Figueroa Montenegro, 2010).

Para que un robot ejecute tareas de navegación debe cumplir ciertas etapas efectuando así el control de navegación (Álvarez Romero & Figueroa Montenegro, 2010).

- *Percepción del entorno:* En esta etapa se obtiene información del entorno a través del sistema sensorial implementado en el robot para estimar la posición de obstáculos, los objetivos y del mismo robot con respecto al entorno.
- *Planificación de trayectoria:* Para esta etapa se diseña la mejor ruta a seguir para que el robot llegue al punto deseado sin colisiones.
- *Generación de trayectoria:* Se elaboran los algoritmos de planificación de trayectoria.
- *Seguimiento de trayectoria:* Finalmente en base a la trayectoria generada se ejecutan los comandos de desplazamiento en los actuadores del robot.

Según el requerimiento de información percibida del entorno se considera dos variantes de algoritmos de navegación: algoritmos fuera de línea y algoritmos en línea (Álvarez Romero & Figueroa Montenegro, 2010)..

- *Algoritmos fuera de línea:* Estos algoritmos requieren de una completa información del entorno, es decir, requieren información de la localización tanto de los obstáculos como de los demás robots del sistema si es que existen. La trayectoria a seguir se envía al robot antes que empiece a desplazarse.
- *Algoritmos en línea:* Este tipo de algoritmos requiere mínima información del entorno, tanto la planificación de trayectoria como la adquisición de información del entorno son procesamientos continuos. El sistema sensorial implementado en el robot el que proporciona continuamente información del entorno.

Algunos de los métodos utilizados para planificación de trayectoria están basados en modelos geométricos Roadmap (RM) como: grafos de visibilidad, diagramas de voronoi, descomposición de celdas, entre otros; o en modelos probabilísticos de los cuales destacan: planificador probabilístico Roadmap (PRM), campos de potencial y RRT (rapidly exploring random tree) (Choset et al., 2005; Diego & Fernando, 2012).

- *Modelos geométricos:* El espacio de trabajo se modela como un grafo, uniendo puntos iniciales y finales. Para después seleccionar el mejor camino que permita minimizar distancia, tiempo y costo computacional. Este tipo de métodos son aplicables en entornos sencillos con robots de pocos grados de libertad, para entornos complejos el costo computacional es elevado debido al número de nodos generados. Para mayor detalle revisar (Choset et al., 2005).
- *Modelos probabilísticos:* El grafo se construye aleatoriamente mediante muestreo de puntos al azar en el espacio de trabajo. Este método es comúnmente usado en espacios

de trabajo complejos, pero no puede ser usado en tiempo real. Su principal desventaja radica en no reconocer la inexistencia de ruta libre de colisiones (Choset et al., 2005; Diego & Fernando, 2012).

2.6.1 Método de planificación de trayectoria RRT

El método planificador RRT construye un árbol que crece explorando desde el punto de origen hasta el punto final, este árbol crece cubriendo todo el espacio libre de colisiones y después mediante un posprocesado se determina la mejor ruta (Diego & Fernando, 2012; López et al., 2010).

La nomenclatura común usada para entender algoritmos RRT es: \mathcal{C} conjunto de todas las configuraciones posibles, \mathcal{C}_{free} subespacio libre de colisiones, q_{init} configuración inicial, q_{fin} configuración deseada, q_{rand} configuración generada aleatoriamente, q_{near} configuración próxima a q_{rand} , q_{new} configuración nueva a ser añadida al árbol y ϵ longitud del segmento o rama a ser añadido al árbol (Choset et al., 2005; Diego & Fernando, 2012; López et al., 2010).

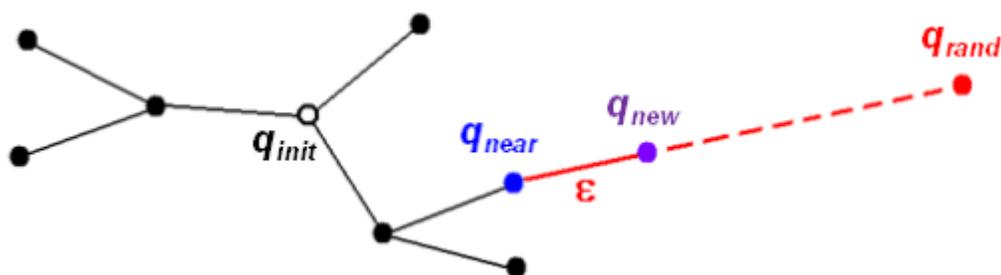


Figura 21. Esquemático del algoritmo RRT
Fuente: (Rugeles & Leon, 2010)(Diego & Fernando, 2012)

El método planificador RRT construye un árbol que crece explorando desde el punto de origen hasta el punto final, este árbol crece cubriendo todo el espacio libre de colisiones y después

mediante un posprocesado se determina la mejor ruta (Diego & Fernando, 2012; López et al., 2010).

Algoritmo 1. Algoritmo básico RRT

RRT(q_{init})

Árbol[0] = q_{init}

para $k = 1$ *k hasta* = $Kmax$

 Calcular el valor de q_{rand} a través de la función *Configuración_aleatoria*()

 Ampliar árbol usando la función *Extiende*(Árbol, q_{rand})

siguiente k

retorna Árbol

fin

El Algoritmo 1 representa el procedimiento a seguir de un algoritmo RRT, donde la función “*Extiende*” amplía la ramificación del árbol mientras no se llegue a la configuración final. El valor $Kmax$ interrumpe al algoritmo una vez que se ha corrido un determinado número de iteraciones sin haber llegado a la configuración final.

Algoritmo 2. Algoritmo RRT – función *Extiende*

Extiende(*Arbol*, q_{rand})

Calcular el valor de q_{near} de la función *Vecino_mas_cercano*(q_{rand} , *Árbol*)

si *Nueva_configuracion*(q_{rand} , q_{near} , q_{new}) es verdadero no existe colisión

Añade_vértice (*Árbol*, q_{new})

si $q_{new} = q_{rand}$

retorna “alcanzado”

caso contrario

retorna “avanzado”

caso contrario

retorna “rechazado”

fin

En el *Algoritmo 2* se describe en detalle el proceso de la función “*Extiende*” del método de planificación RRT. Al calcular q_{near} y descartar posibles colisiones con q_{rand} y la localización de los obstáculos, esta función genera nuevos vértices para extender la trayectoria como se evidencia en la Figura 21, hasta llegar al punto final donde retorna el objetivo como alcanzado.

CAPITULO III

HARDWARE Y SOFTWARE

3.1 Introducción

En este capítulo se describen los elementos seleccionados para el desarrollo de hardware y software del presente proyecto de titulación, y las razones por las cuales fueron seleccionados. Además una descripción de los problemas de hardware y software enfrentados en la realización del proyecto.

3.2 Hardware del sistema

El presente proyecto consiste en el desarrollo de un sistema multi-robot móvil cooperativo capaz de detectar y clasificar objetos de color en un entorno estructurado. El sistema está compuesto de cinco robots de configuración diferencial dotados de sensores, actuadores y microcontroladores necesarios para el control, la navegación y la comunicación del sistema. A continuación se describen los elementos de hardware utilizados por el sistema multi-robot.

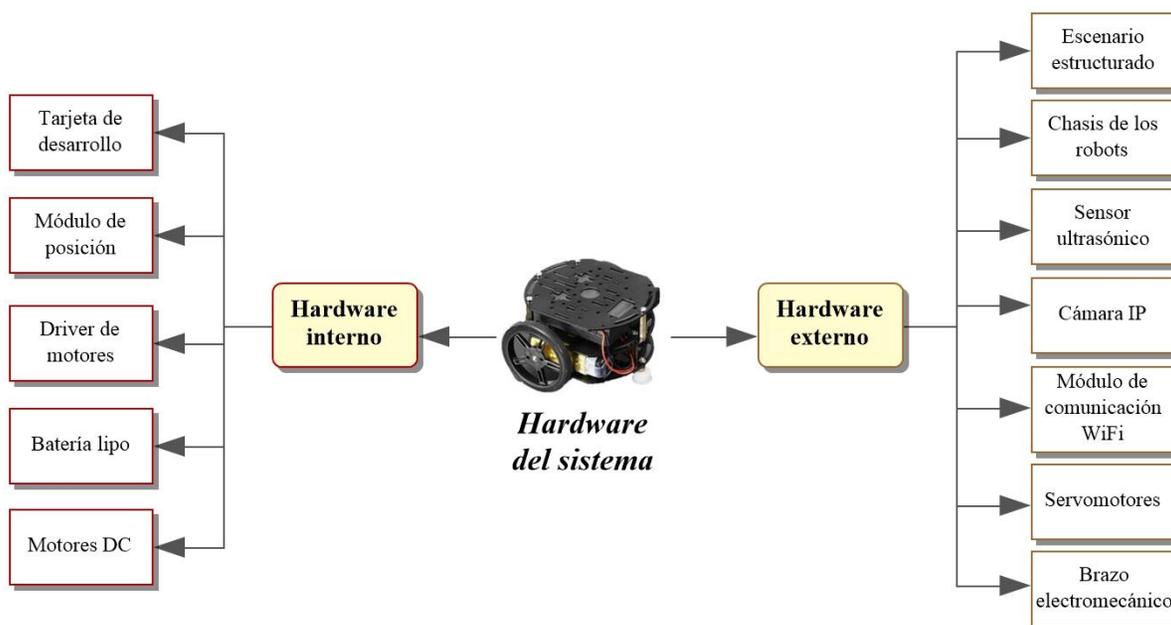


Figura 22. Diagrama de bloques del hardware del sistema.

3.2.1 Entorno estructurado

El sistema multi-robot cooperativo se implementa sobre un escenario de aproximadamente 1.5 metros cuadrados, las dimensiones de los robots móviles y los objetos de color serán de acorde con las del escenario. El diseño del escenario es un rompecabezas para facilidad de traslado y montaje, las piezas del piso tienen dimensiones de aproximadamente 75x75 cm, y las dimensiones de las paredes del escenario son de 150x40 cm aproximadamente, ver Figura 23.

El material elegido para la fabricación del escenario es madera MDF, siendo las razones principales de selección de este material: bajo costo en relación a la madera maciza, superficie lisa ideal para el desplazamiento de pequeños robots, y facilidad de corte.



Figura 23. Escenario de pruebas.

Cada pared del escenario tiene un relieve de 60 *cm* de largo por 40 *cm* de ancho y 3 *mm* de espesor localizado en el centro de cada pared y pintado de diferente color, los colores son elegidos de acuerdo al método SAF-T-DATA de J.T. Baker (Colmenares, 2014).

3.2.1.1 Objetos de color

Dentro del escenario estructurado se encuentran cuatro objetos de diferente color, de acuerdo al método SAF-T-DATA de J.T. Baker (Colmenares, 2014). El material seleccionado para fabricar los objetos de color es espuma flex cilíndrica de 8 *cm* de diámetro y 10 *cm* de altura.

Se seleccionó este material porque es ligero, es de bajo costo y puede cortarse fácilmente. El tamaño elegido para los cilindros de espuma flex permite que sean fácilmente reconocidos tanto

por la cámara como por los sensores de distancia, y para el color se forra los cilindros con fomix de color.



Figura 24. Objetos de color.

3.2.1.2 Condiciones ambientales

Las experimentaciones fueron realizadas en un entorno estructurado, en el cual se trató de controlar el nivel de luminosidad. Para mantener estable el nivel de luz ambiental se definió un rango de trabajo de 18h00 a 23h00, tiempo en el que la luz ambiental no influye en el escenario de pruebas, trabajando solamente con luz artificial blanca. En ocasiones donde será necesario la realización de pruebas en el día, se utiliza cortinas y luz artificial blanca para mantener estable el nivel de luminosidad en el escenario estructurado.

3.2.2 Robots móviles

El modelo de robot seleccionado para este proyecto es el mini 3-layer round robot de la empresa estadounidense Adafruit, este robot fue seleccionado por su tamaño, configuración, facilidad de implementación y costo accesible. El robot móvil puede ser usado para aplicaciones tanto de educación como de investigación; este modelo de robot presenta una configuración diferencial

constando de dos ruedas dotadas de motores DC y una rueda loca para mantener la estabilidad del robot. El tamaño del robot esta de acorde con las medidas del escenario.



Figura 25. Mini 3-layer round robot de Adafruit.

3.2.2.1 Chasis

El material del que está fabricado el chasis del robot es aluminio anodizado, este material posee gran resistencia ante agentes químicos, dureza y baja conductividad eléctrica. El chasis del robot está formado por tres capas de aluminio anodizado de dimensiones 10x8.5 cm unidos con tornillos de 2.5mm de diámetro como se ve en Figura 26.

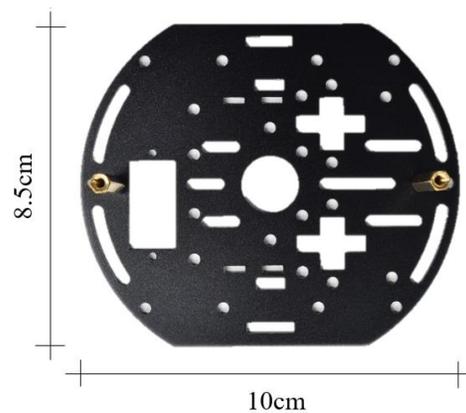


Figura 26. Dimensiones del chasis de los robots.

3.2.2.2 Motores DC

Los motores DC transforman energía eléctrica en energía mecánica, este tipo de motores permite variar la velocidad de giro sin disminuir el torque siendo utilizados en aplicaciones de control tanto en la industria como en robótica. Para aplicaciones de baja escala se utilizan motores DC con cajas reductoras denominados motorreductores DC, ver Figura 27, el torque se calcula mediante la potencia en HP y la velocidad en RPM que presente el motor.



Figura 27. Motorreductor DC.

Los motorreductores seleccionados para este proyecto presentan las siguientes características:

- Torque: 0.8Kg.cm – 1.1Kg.cm.
- Velocidad sin carga: 125RPM - 230 RPM.
- Velocidad con carga: 95RPM – 175 RPM.
- Consumo de corriente sin carga: 60mA – 100mA.
- Consumo de corriente con carga: 200mA - 670mA.
- Voltaje de alimentación: 3V – 12V DC, recomendado usar 3V – 6V DC.

- Parámetros de reducción: 1:48.
- Dimensiones: 71mm x 27mm x 23 mm aproximadamente.
- Peso: 50g.
- Ruido: < 65dB.
- Material de fabricación: plástico resistente y no tóxico.

El Motorreductor usado en este proyecto se visualiza en la Figura 27, se adquirió al igual que el chasis del robot de la empresa Adafruit.

3.2.2.3 Llantas

El material de las llantas determina el agarre del robot al desplazarse, dependiendo de la superficie en la que se posicionará el robot se elige el material de las llantas. Para este proyecto se seleccionó llantas de silicona de la marca Pololu que poseen excelente agarre en superficies de madera.



Figura 28. Llantas de silicona marca Pololu.

Las llantas miden 60mm de diámetro y 4mm de espesor, se acoplan mediante presión en el eje de los motores y adicionalmente utilizan tornillos para mejor fijación.

3.2.3 Sensores y actuadores

Los sensores y actuadores integrados al sistema multi-robot deben cumplir los siguientes criterios generales: baja sensibilidad al cambio de luz ambiental, tiempo corto de reacción y fácil calibración, funcionamiento en lugares cerrados, bajo consumo de potencia, alta integración y escalabilidad.

3.2.3.1 Sensor de distancia

Actualmente existen diversos tipos de sensores para detectar la proximidad de un objeto a diferentes distancias y con diferente sensibilidad, siendo utilizados en aplicaciones cotidianas, industriales o de investigación según sus características. Para este proyecto se analizó dos tipos de sensores de distancia basados en la medición del tiempo de vuelo de la señal que emiten.

3.2.3.1.1 Sensor de ultrasónico modelo HC-SR04

El sensor emite señales ultrasónicas inaudibles, y para determinar la distancia entre el sensor y un objeto se calcula el tiempo de vuelo de la señal a ser reflejada con el objeto (González Jiménez & Baturone, 2015). A pesar de ser un sensor de bajo costo posee una sensibilidad de $\pm 3 \text{ mm}$ en un rango teórico de medición de 2 cm a 450 cm , y un ángulo efectivo de medición de 15° por lo que es común utilizarlo en aplicaciones de robótica (ITead Studio, 2010).

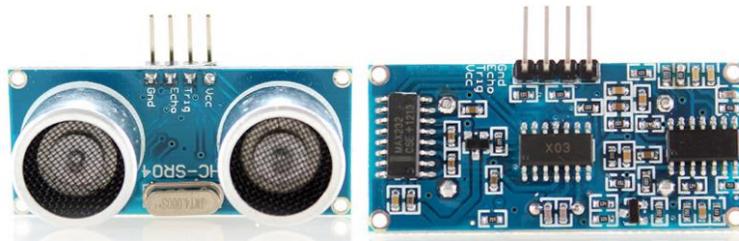


Figura 29. Sensor ultrasónico HC-SR04.

Una limitante de estos sensores es que presentan baja precisión en entornos con múltiples obstáculos, ya que el ángulo de medición del sensor, ver Figura 30, hace que la señal se refleje en las superficies de los objetos cercanos provocando falsas mediciones.

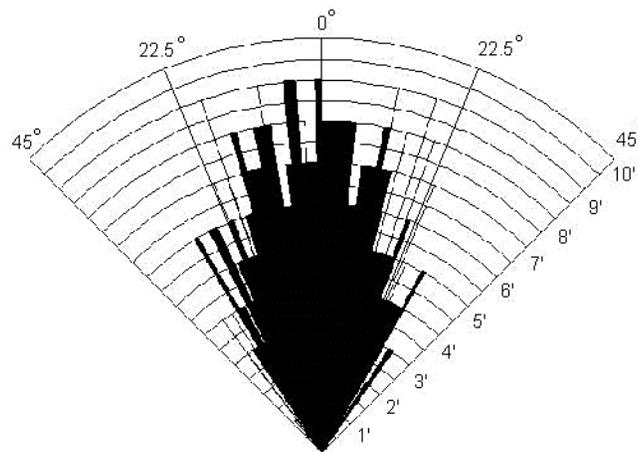


Figura 30. Rendimiento del sensor HC-SR04.

Fuente: (ITead Studio, 2010)

Las características técnicas relevantes del sensor se describen a continuación:

- Dimensiones: 43x20x17 mm.
- Voltaje de alimentación: 5 V DC.
- Frecuencia de trabajo: 40 KHz.

- Rango de medición: 2 cm – 450 cm.
- Consumo de corriente: 15 mA.
- Duración mínima del pulso de disparo: 10 μ s (TTL).
- Duración del eco de salida: 100 μ s – 25000 μ s (TTL).
- Tiempo de espera entre mediciones: 20 ms.

En la Figura 31 se visualiza las respuestas del sensor HC-SR04 obtenidas experimentalmente en el escenario estructurado con respecto a un objeto cilíndrico a diferentes distancias y ángulos. Se compara la distancia entre el sensor y el objeto cilíndrico medida manualmente con la distancia experimental adquirida del sensor utilizando una muestra de 200 datos.

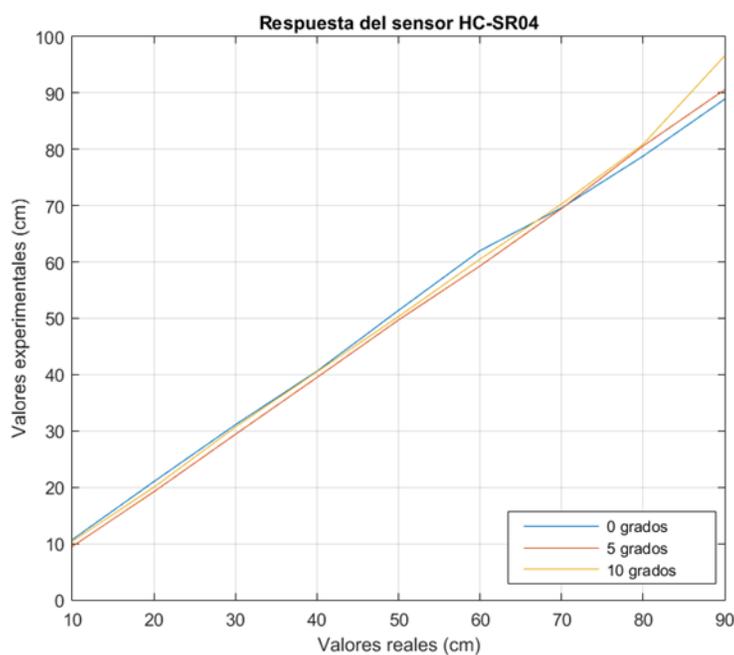


Figura 31. Curva de medición del sensor HC-SR04

Se deduce de la Figura 32 que los valores obtenidos presentan un error aceptable menor a 1 cm cuando el objeto está entre 0 y 5 grados del eje de medición del sensor tanto para distancias cortas como largas, y a partir de los 10 grados las mediciones presentan errores mayores a 2 cm.

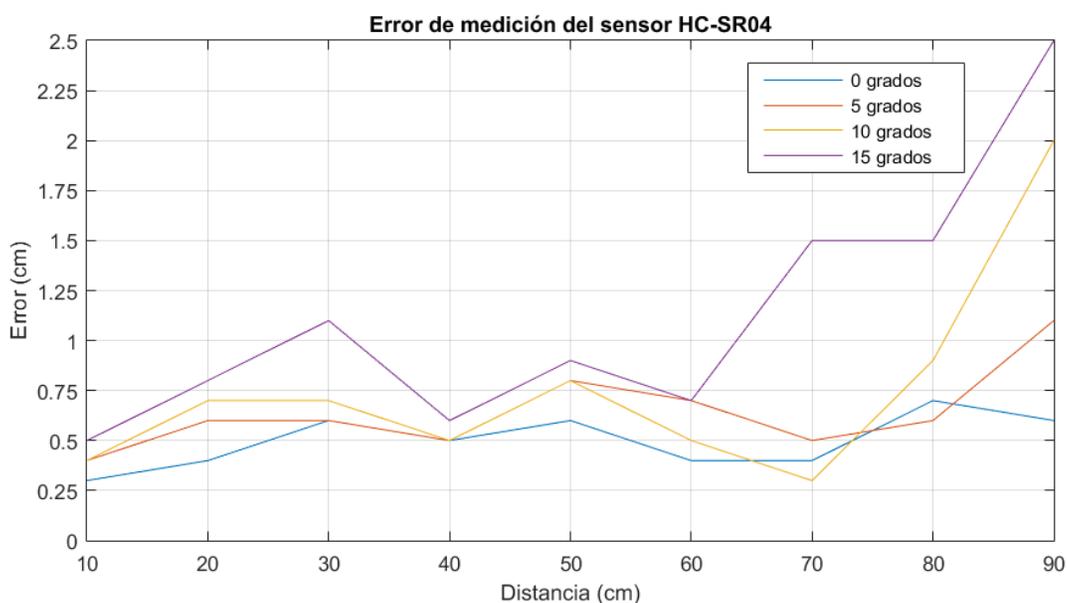


Figura 32. Error de medición del sensor HC-SR04.

No se realizaron mediciones para distancias superiores de 130 cm debido al tamaño del escenario. Además el sensor HC-SR04 presenta variaciones de 0.75 cm a 2 cm en distancias superiores a 80 cm como se puede observar en Figura 32.

El ángulo de apertura del sensor es amplio, siendo necesario separar los objetos cilíndricos de color a distancias considerables para no confundir las mediciones entre ellos, siendo este el principal inconveniente del sensor HC-SR04.

3.2.3.1.2 Sensor de laser modelo VL53LOX

El principio de funcionamiento del sensor es la medición del tiempo de vuelo de la señal, de manera similar que el sensor HC-SR04. El sensor emite pulsos de luz láser infrarroja y mide el tiempo que tardan en alcanzar al objeto más cercano y volver al detector, siendo considerado como un minúsculo sistema LIDAR (Ada, 2018).

El sensor posee una sensibilidad de $\pm 1 \text{ mm}$ en un rango teórico de medición de 2 cm a 200 cm , y un ángulo total de medición de 25° . No se ve afectado por el efecto rebote y las condiciones de luz ambiental afectan mínimamente la medición, el ángulo de medición estrecho permite estimar la distancia de un objeto frente al sensor con mayor precisión (Ada, 2018; ST, 2018).



Figura 33. Sensor láser VL53LOX.

Las características técnicas del sensor se describen a continuación (Ada, 2018; ST, 2018):

- Dimensiones: 21x18x2.8 mm.
- Voltaje de alimentación: 3.5 V – 5 V.
- Consumo de corriente: 18 mA.
- Rango de medición: 2cm – 200 cm
- Emisor láser: 940 nm VCSEL (Vertical Cavity Surface-Emitting Laser).

- Detector SPAD (Single Photon Avalanche Diodes).
- Ángulo total de medición: 25° (área de 44 cm de diámetro a 1 m de distancia).
- Interfaz de comunicación I2C.

En la Figura 34 se muestra las respuestas del sensor VL53L0X obtenidas experimentalmente a diferentes distancias y ángulos en el escenario estructurado con respecto a un objeto cilíndrico.

La gráfica se realizó en base a la toma de 200 datos adquiridos del sensor con respecto a una medición manual. En la Figura 35 se observa el margen de error de las mediciones realizadas con el sensor VL53L0X. Cuando el objeto a medir se encuentra en el eje del sensor se obtienen errores máximos de 7.5 cm a distancias de 90 cm, y al desplazar el objeto 10 grados del eje del sensor los errores de medición aumentan significativamente como se observa en la figura.

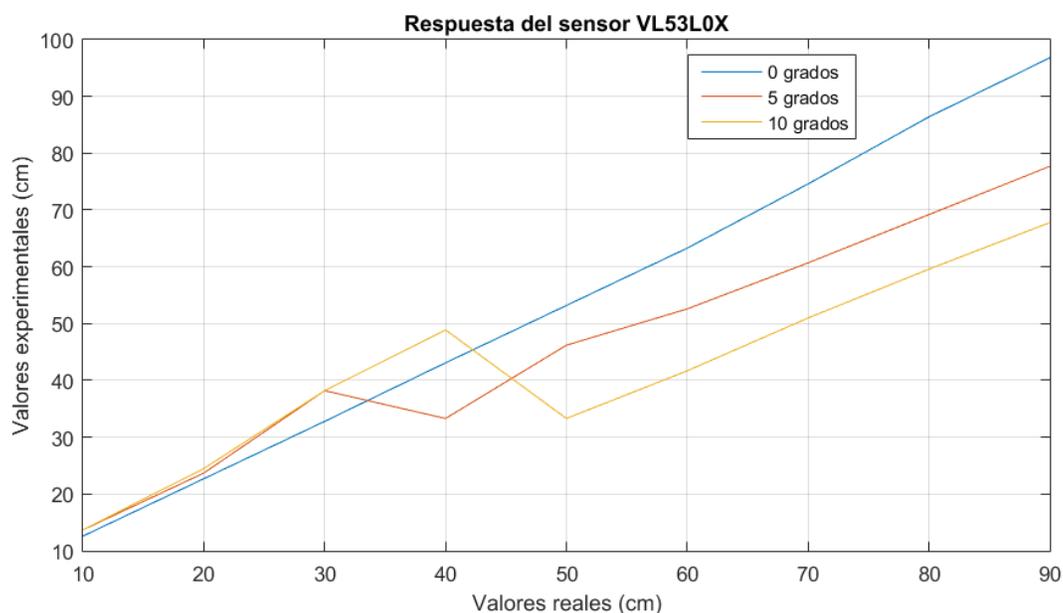


Figura 34. Curva de medición del sensor VL53L0X.

De la Figura 35 se deduce que apertura del sensor VL53L0X es reducida, debido a la variación de los datos al desplazar el objeto cilíndrico del eje de medición 5 y 10 grados. Al presentar un ángulo de apertura reducido la distancia de separación entre los objetos del escenario estructurado para evitar confusiones de medición es pequeña.



Figura 35. Error de medición del sensor VL53L0X.

3.2.3.1.3 Análisis comparativo

Para el proyecto de titulación se requiere que el sensor pueda medir distancias en un rango de 15 cm a 150 cm con alta precisión en un ambiente cerrado, sin que influya en las mediciones las condiciones de iluminación pre-existentes. Además se requiere que el sensor sea de minúsculas dimensiones, ya que será implementado en los robots de la sección 3.2.2.

Para la selección del sensor de distancia a utilizar se realiza un análisis comparativo de las especificaciones técnicas entre los sensores HC-SR04 y VL53L0X como se muestra en la Tabla 2

y un análisis comparativo en base a los datos adquiridos de los sensores como se visualiza en la Figura 36.

Tabla 2

Análisis comparativo entre sensores de distancia.

	HC-SR04	VL53L0X
Dimensiones	43x20x17 mm	21x18x2.5 mm
Distancia de medición mínima teórica	2 cm	2 cm
Distancia de medición máxima teórica	450 cm	200 cm
Distancia de medición mínima experimental	4 cm	20 cm
Distancia de medición máxima experimental	110 cm	110 cm
Ángulo total de medición	30°	25°
Sensibilidad	±3 mm	±1 mm
Interfaz de comunicación	14	I2C
Voltaje de alimentación	3.5 – 5 V	3.5 - 5 V
Consumo de corriente	15 mA	18 mA
Costo	4.50	16.50

Fuente: (ITead Studio, 2010; ST, 2018)

Observando la Tabla 2 se deduce que sensores VL53L0X son más pequeños que los sensores ultrasónicos HC-SR04, su ángulo de apertura es menor siendo ideales para medir objetos en línea recta y con poca separación, además son más costosos que los sensores ultrasónicos.

En la Figura 36 se observa la comparación de los errores de medición entre los sensores HC-SR04 y VL53L0X, donde el sensor láser VL53L0X presenta un mayor margen de error en las mediciones en comparación con el sensor ultrasónico HC-SR04.

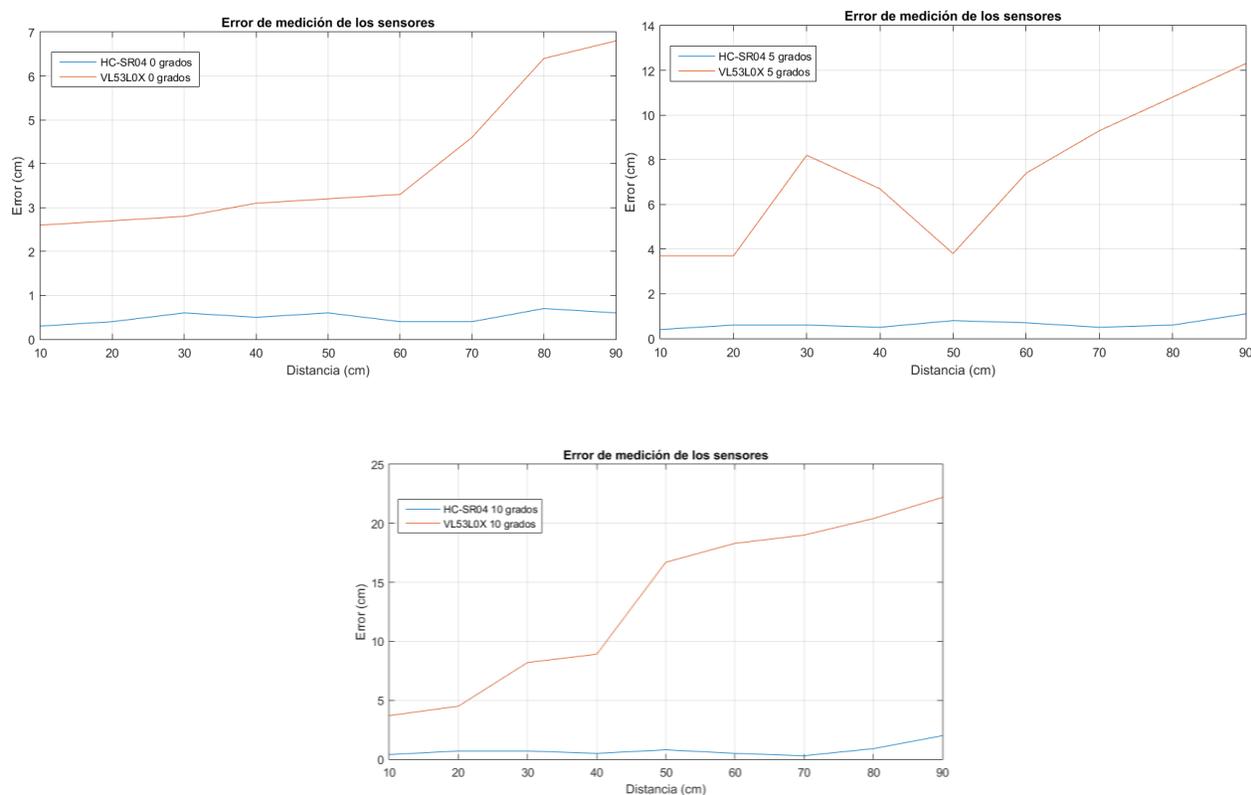


Figura 36. Comparación del error de medición entre los sensores HC-SR04 vs. VL53L0X.

A pesar que el sensor VL53L0X presenta un menor ángulo de medición y mayor sensibilidad, los errores presentados en la adquisición de datos no son aceptables para este proyecto, siendo seleccionado el sensor HC-SR04 debido a su mínima variación en la adquisición de datos a diferentes distancias pese a que su ángulo de medición es amplio.

3.2.3.2 Sensor de medición inercial

Para este proyecto se necesita saber la orientación y el desplazamiento angular del robot guía dentro del escenario para un posterior proceso de localización, actualmente esta tarea puede ser realizada por diversos sensores como GPS, giroscopios, acelerómetros, brújulas, entre otros.

Se eligió el módulo MPU6050 (Figura 37) como sensor de orientación debido a sus prestaciones computacionales, tamaño y bajo costo. Este módulo es una IMU (unidad de medición inercial) fabricado por Invensense de seis grados de libertad que integra un giroscopio y un acelerómetro cada uno de 3 ejes (grados), además de un procesador digital de movimiento (DMP).

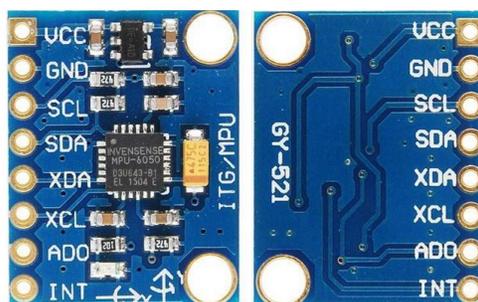


Figura 37. Sensor MPU6050.

Para este módulo existen librerías que combinan los datos obtenidos del acelerómetro y el giroscopio embebidos, permitiendo procesar fácilmente complejos algoritmos de movimiento. A pesar de la precisión teórica del módulo MPU6050 el ruido causa interferencias en las medidas del acelerómetro provocando un margen de error, además los cálculos realizados con los datos del giroscopio acumulan un pequeño error en función del tiempo (InvenSense Inc., 2013).

Para eliminar los errores en las mediciones del acelerómetro y giroscopio se aplica un filtro complementario a las mediciones, es decir un filtro pasa-alto en los datos del giroscopio y un filtro

pasa-bajo en los datos del acelerómetro (InvenSense Inc., 2013). Las principales especificaciones técnicas del sensor se describen a continuación:

- Dimensiones: 20x16x3 mm.
- Voltaje de alimentación: 3.3 V – 5 V.
- Consumo de corriente: 3.6 mA.
- Grados de libertad: 6.
- Rango del acelerómetro: 2g/4g/8g/16g.
- Rango del giroscopio: 250°/seg, 500°/seg, 1000°/seg, 2000°/seg.
- Conversor ADC: 16 bits.
- Interfaz de comunicación: I2C.

De las características presentadas: tanto la alimentación compatible con la tecnología TTL, interfaz de comunicación de uso frecuente como los es I2C y rango de medición suficientes para un giro a 360°, facilitan en gran medida la incorporación del sensor como medio de adquisición de valores inerciales para la aplicación.

3.2.3.3 Sensor de visión – cámara IP

Una cámara IP es un dispositivo que captura y envía señales de audio/video en tiempo real mediante una red IP (López Rodríguez Julio César, 2007; Revista Negocios de Seguridad, 2007).

Para seleccionar el tipo de cámara IP se identificó las características requeridas por el proyecto:

- La cámara solo se usará en interiores, en un ambiente de pruebas estructurado por lo que no necesita de protección adicional.
- El tamaño de la cámara debe ser reducido para poder implementarla sobre un robot de la sección 3.2.2 con dimensiones 10x8.5 *cm*.
- La cámara será usada para reconocimiento de colores y estimación de la posición de los objetos dentro del escenario estructurado, por lo que la resolución de la cámara puede ser de 0.8 (1024x768) – 2.3 (1920x1200) megapíxeles.
- La cámara trabajará en condiciones de iluminación: natural y artificial preestablecidas en la sección 3.2.1.2, por lo que no es necesario que posea visión nocturna.
- No es necesario que posea memoria para grabación, ya que el video se transmitirá de forma inalámbrica en tiempo real.
- El tiempo de autonomía de la batería de la cámara debe ser mayor de 30 minutos, además la batería debe ser recargable.

Con base en el análisis de las características requeridas por el proyecto, se optó por una mini-cámara IP inalámbrica como se muestra en la Figura 38. Esta cámara cumple con los requerimientos necesarios para la detección y estimación de la posición de objetos de color dentro del escenario estructurado.



Figura 38. Mini-cámara IP.

A continuación se describen las especificaciones técnicas la mini-cámara IP seleccionada:

- Dimensiones: cuerpo 19.5 cm y lente 0.7 cm.
- Formato de video: AVI.
- Resolución: 1280x720 megapíxeles, 30fps.
- Capacidad de la batería recargable: 480 mAh.
- Tiempo de grabación: 70 minutos.
- Tiempo de carga: 2.5 horas.

3.2.3.4 Driver L298N para motores DC

Los robots deben seguir una determinada trayectoria dentro del escenario estructurado para realizar las tareas de reconocimiento y clasificación de objetos de color. El centro de control emite señales de la trayectoria que deben seguir los robots, estas señales deben ser interpretadas por los motores para lograr desplazar los robots con una trayectoria determinada dentro del escenario.

El módulo L298N es un dispositivo de potencia que interpreta las señales permitiendo controlar la velocidad y el sentido de giro de hasta dos motores DC, soportando voltajes de alimentación de hasta 35 V y corrientes de máximo 2 A por canal (STMicroelectronics, 2000).

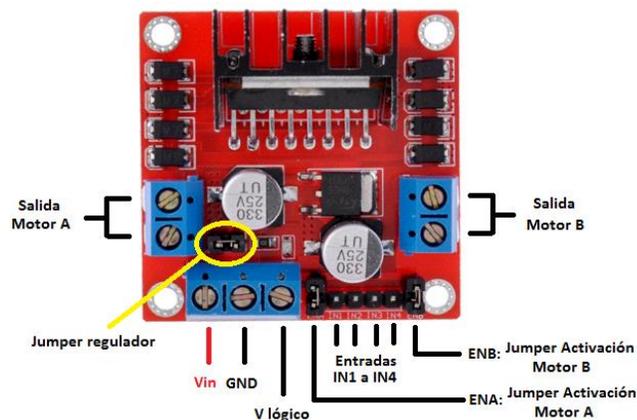


Figura 39. Driver L298N.

Para controlar los giros y velocidades de cada uno de los robots del sistema se emplea en cada el driver L298N para el control de los motores DC montados en los robots. Sus especificaciones técnicas se describen a continuación:

- Dimensiones: 55x60x30 mm
- Voltaje de alimentación: 4-35 V.
- Corriente máxima soportada: 4 A.
- Voltaje de alimentación lógica: 5 V.
- Entradas PWM: 2.

3.2.3.5 Servomotor SG90

El servomotor es un motor DC que permite el control del desplazamiento angular, es decir tiene la capacidad de posicionar su eje en un ángulo determinado. Internamente está compuesto de una caja reductora que disminuye la velocidad para aumentar el torque, un potenciómetro y un circuito electrónico para controlar el ángulo.

Los servomotores pueden desplazarse angularmente en un rango de 0° a 180° , y debido a las limitaciones internas no son capaces de completar una vuelta. Usualmente son utilizados en aplicaciones de robótica en las que se requiere controlar la posición de giro como: en torretas, brazos electromecánicos, orientación de sensores, entre otras.

La principal diferencia entre los modelos de servomotores existentes es el torque, los fabricantes recomiendan elegir un modelo de servo con torque superior al requerido para no estropearlo. Los servomotores poseen tres cables de conexión, dos para alimentación y uno para la señal de control PWM.

Para este proyecto se necesitan en total diez servomotores distribuidos de la siguiente manera: cuatro para mover brazos electromecánicos montados en los robots, cinco para la orientación de los módulos de comunicación WiFi, y por último uno para mover la cámara IP.



Figura 40. Servomotor SG90.

El modelo seleccionado de servomotor es el SG90 por las siguientes razones: tamaño reducido, alto torque, bajo consumo y precio accesible; siendo comúnmente usados en proyectos de robótica. Estos servos trabajan con una señal PWM entre 1 ms y 2ms, con un periodo de 20 ms (50 Hz). Las especificaciones técnicas relevantes se describen a continuación:

- Dimensiones: 22x11.5x27 mm.
- Voltaje de alimentación: 4-7.2 V, ideal 6 V.
- Velocidad de operación a 4.8 V: 0.12 s/60°.
- Torque a 4.8 V: 1.8 Kg/cm.
- Consumo de corriente a 4.8 V: 327 mA.
- Peso: 11 gramos.

3.2.3.6 Brazo electromecánico

Un brazo electromecánico o brazo robótico, está compuesto en esencia de una estructura mecánica y un accionamiento electrónico normalmente programable. En la industria también se los

conoce como manipuladores e imitan las funciones de un brazo humano, razón por la que son utilizados en actividades como: soldar, manipular cargas y ensamblar piezas.

El número de articulaciones móviles establece el número de grados de libertad (DOF) del brazo robótico, y se determinan en función de la necesidad de la aplicación. Para realizar el proceso de clasificación de los objetos de color, es necesario desplazarlos de su posición inicial a una determinada dentro del escenario estructurado. Siendo necesario el transporte de estos objetos de color mediante un brazo robótico de 1DOF montado en los robots móviles, el mismo que muestra en la Figura 41.

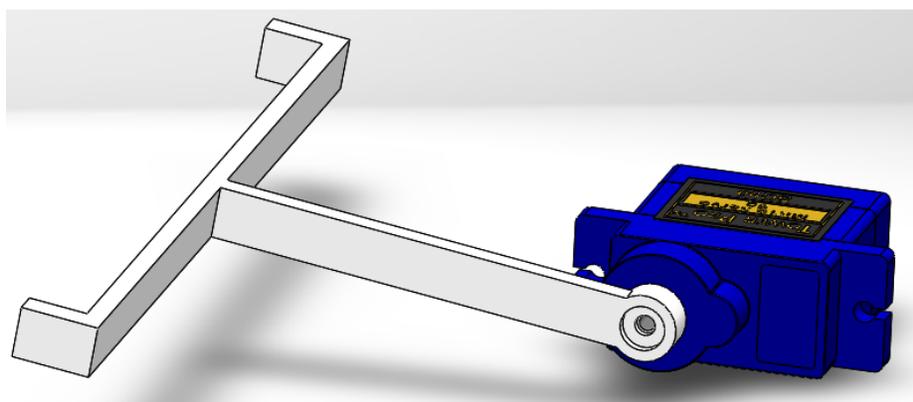


Figura 41. Diseño del brazo electromecánico.

La estructura del brazo robótico está diseñada en un software CAD y posteriormente elaborada mediante impresión 3D. Las órdenes para controlar el movimiento del brazo serán enviadas de la tarjeta de desarrollo y ejecutadas por un servomotor SG90.

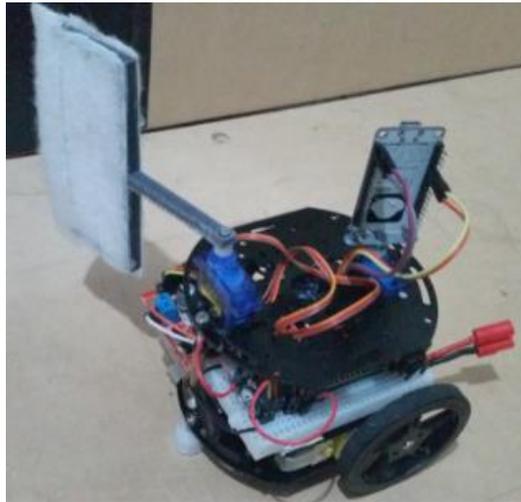


Figura 42. Brazo electromecánico.

3.2.4 Tarjetas de desarrollo

Las tarjetas de desarrollo son dispositivos electrónicos que incorporan un microcontrolador, memoria y módulos de comunicación. Estas tarjetas no necesitan de dispositivos externos como grabadores para subir programación al microcontrolador, la programación se realiza mediante el puerto USB. Actualmente el mercado tecnológico ofrece diversas tarjetas de desarrollo para aplicaciones de control, procesamiento de datos y comunicación en sistemas robóticos, las más utilizadas en sistemas de baja escala son:

- *Arduino*: Esta tarjeta es fácil de configurar, es compatible con varios módulos de hardware y además tiene buena relación costo/beneficio. Los microcontroladores implementados en las tarjetas son de la serie ATmega creados por Atmel. Actualmente existen varios modelos de tarjetas Arduino que difieren entre sí por la capacidad computacional y la memoria.
- *NodeMCU*: Está basada en el módulo de comunicación WiFi ESP8266.

- *Teensy*: Esta diseñada para ser montada en protoboards o en placas PCB, es muy potente debido a su procesador ARM Cortex de 32 bits, y es compatible con las librerías de Arduino.

Para el proyecto se seleccionó las tarjetas de desarrollo teensy 3.5 y nodeMCU v1.0 V2. La tarjeta Teensy 3.5 se encarga de los sensores y actuadores del sistema multi-robot y la tarjeta NodeMCU realiza la comunicación inalámbrica con la PC.

3.2.4.1 Teensy 3.5

Teensy es una tarjeta de desarrollo basada en un procesador ARM Cortex-M4 de 32 bits y está diseñada para usarse en protoboards o en placas PCB. En comparación con tarjetas de desarrollo de similares características, teensy ofrece mejores prestaciones en costo y funcionalidad. Para programar la tarjeta no es necesario de dispositivos externos, ya que viene con un gestor de arranque o bootloader que permite programarlo usando conexión USB. Se puede programar teensy en un editor de programas basado en C o mediante el IDE de Arduino para lo cual se requiere el complemento teensyduino (DigiKey, 2017; PJRC, s. f.).

Actualmente la compañía PJRC dispone de seis modelos de tarjetas teensy que varían en modelo de procesador, tamaño de memoria, número de entradas y salidas, y tamaño de las placas.

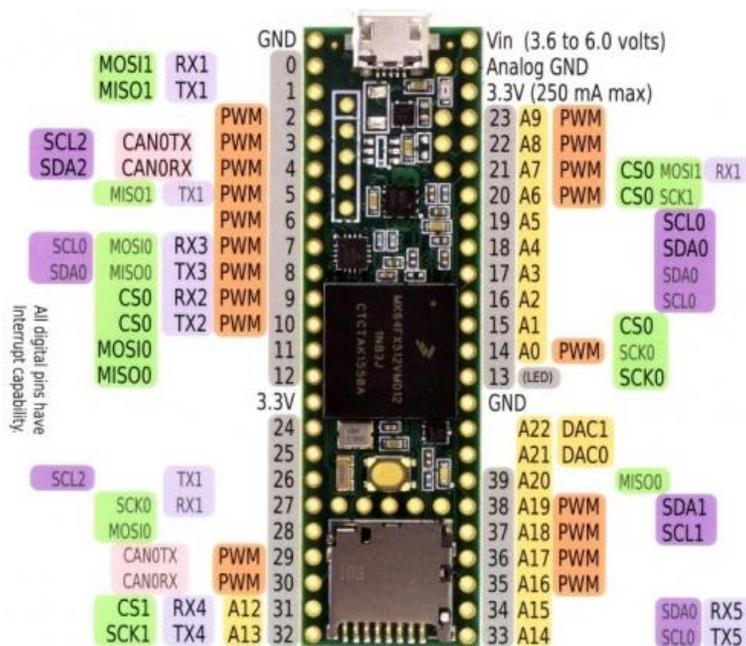


Figura 43. Esquema pinout de la tarjeta teensy 3.5

Fuente: (González Jiménez & Baturone, 2015)

El tamaño de la placa de la tarjeta teensy 3.5 es de 62.3mm x 18mm, siendo ideal para montaje en robots móviles pequeños. Además que la tarjeta cuenta con un procesador ARM Cortex-M4 de 120MHz con múltiples canales de acceso a memoria, siendo la potencia de procesamiento, tamaño y costo, las características fundamentales para ser seleccionada en aplicaciones de robótica móvil.

A continuación especificaciones técnicas de la tarjeta (PJRC, s. f.; SparkFun, s. f.):

- Procesador ARM Cortex-M4 de 120 MHz.
- Microcontrolador MK64FX512VMD12.
- Voltaje de alimentación 3.3V DC a 6V DC.
- Voltaje de operación: 3.3 V.

- 58 pines digitales E/S, cada pin puede ser de interrupción y son tolerantes a 5V.
- 20 salidas PWM.
- 3 puertos I2C.
- 1 puerto USB tipo B de velocidad 12 Mbit/s.

3.2.4.2 NodeMCU v1.0 V2

La tarjeta de desarrollo NodeMCU está orientada al IoT (internet de las cosas), y se basa en el módulo WiFi ESP8266 fabricado por la compañía Espressif. ESP8266 es un SoC (system on chip) integrado principalmente por un procesador de 32 bits que funciona a una máxima velocidad de 160 MHz y un chip de conectividad WiFi (nodeMCU, 2014).

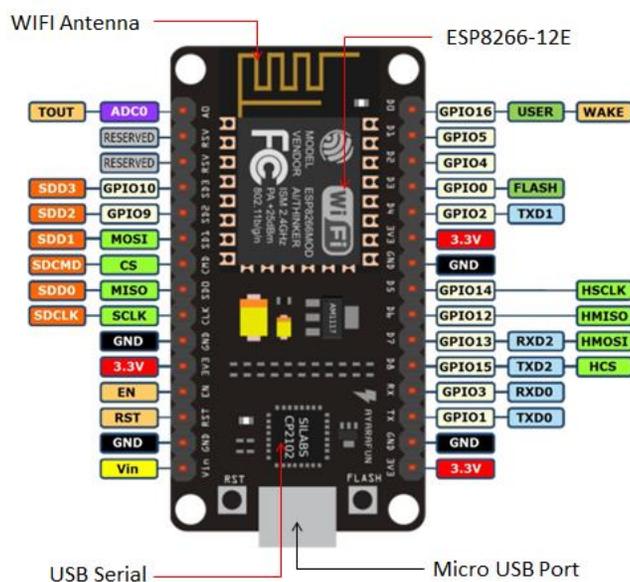


Figura 44. Esquema pinout de la tarjeta nodeMCU V1.0 V2
Fuente: (nodeMCU, 2014)

NodeMCU puede ser programado en el IDE de arduino o mediante el lenguaje interpretado LUA (LUA, s. f.), e incorporan un conversor USB a serie para programar directamente la tarjeta. Actualmente existen tres versiones de la tarjeta nodeMCU que varían en memoria y número de pines, siendo elegida la versión dos para este proyecto de titulación (Owner, 2017). A continuación especificaciones técnicas de la tarjeta:

- SoC: ESP8266 (ESP-12).
- Procesador: Tensilica L106 32bits, velocidad 80MHz a 180MHz.
- Memoria RAM: 128KB, FLASH: 512KB.
- 11 GPIO, cada GPIO puede ser PWM.
- 1 entrada ADC, resolución 10 bits.
- Soporta el estándar 802.11b.
- Voltaje de alimentación externa 3.3V – 10V DC.
- Voltaje de GPIO: 3.3 V.
- Antena incorporada en la placa PCB.

La tarjeta NodeMCU al estar basada en el módulo ESP8266 fusiona la tecnología wifi con la rapidez de procesamiento de la placa y sus funcionalidades del GPIO. Otras tecnologías como ZigBee con módulos ZigBee y Xbee que trabajan en el estándar 802.15.4 también son una opción de comunicación inalámbrica, pero por su costo exceden en gran medida a la tarjeta NodeMCU que de igual manera permite comunicaciones inalámbricas para aplicaciones puntuales.

3.2.4.3 Análisis comparativo entre tarjetas arduino, teensy y nodeMCU

Se realizó un análisis comparativo para la selección de las tarjetas se entre arduino, teensy y nodeMCU como se indica en la Tabla 3.

Para el proyecto de titulación se requiere como máximo 12 pines GPIO para conexión de sensores y actuadores, pines para la comunicación I2C con el módulo de orientación y pines Tx, Rx para comunicación serial con el módulo WiFi.

Observando la Tabla 3 se deduce que la tarjeta teensy 3.5 ofrece mayor rendimiento computacional, dispone de más pines GPIO en comparación con las otras tarjetas y además cada pin GPIO puede usarse como pin de interrupción, también posee mayor capacidad de memoria lo que ayuda en la implementación de algoritmos complejos y el voltaje de operación de la tarjeta es de 3.3V lo suficiente para activar a los sensores y actuadores del sistema. Teensy 3.5 puede ser programada mediante el IDE de arduino a través de la instalación del componente teensyduino, y es compatible con la mayor parte de librerías del IDE. Estas son las razones para seleccionar la tarjeta de desarrollo teensy para este proyecto.

La tarjeta nodeMCU fue seleccionada sobre los módulos WiFi ESP8266 por ser programable, poseer mayor velocidad de cómputo y memoria en comparación con la tarjeta arduino, además de tener un tamaño reducido y un costo similar con los módulos ESP8266.

Tabla 3
Análisis comparativo entre tarjetas de desarrollo.

	Arduino Mega	Teensy 3.2	Teensy 3.5	NodeMCU
Microcontrolador	ATmega2560	MK20DX256VLH7	MK64FX512VMD12	Tensilica L106
Velocidad del procesador	16 MHz	72 MHz	120 MHz	80 MHz
Memoria Flash	256 KB	256 KB	512 KB	512 KB
Memoria RAM	8 KB	64 KB	192 KB	128 KB
GPIO	54	34	58	11
ADC	16	21	27	1
DAC	-	1	2	-
PWM	14	12	20	11
Conector	USB B	USB micro-B	USB micro-B	USB micro-B
Voltaje de operación	5.0 V	3.3 V	3.3 V	3.3 V
Voltaje de alimentación	7.0 – 12.0 V	5.0 V	5.0 V	5.0 V
Serial	4	3	6	2
SPI	2	1	3	2
I2C	1	2	3	1
Tarjeta Sd	-	-	1	-

Fuente: (nodeMCU, 2014; SparkFun, s. f.)

3.2.5 Batería LiPo

Para alimentar a los robots del sistema es necesario utilizar una batería LiPo (litio y polímero) recargable que sea ligera y de tamaño reducido. El modelo de la batería fue seleccionado tras un análisis de consumo de voltaje y corriente de cada elemento implementado en los robots, considerando los máximos valores de consumo de corriente, ver Tabla 4.

Tabla 4

Análisis de consumo de energía para la selección de la batería.

Cantidad	Hardware	Voltaje de operación	Consumo de corriente individual	Consumo de corriente total
1	Sensor de distancia	3.3 – 5 V	18 mA	18 mA
1	Sensores MPU6050	3.3 – 5 V	3.6 mA	3.6 mA
1	Driver L298N	5 - 35 V	450 mA	450 mA
2	Motorreductores	3 - 6 V	250 mA	500 mA
2	Servomotor SG90	3.3 – 5 V	327 mA	654 mA
1	Teensy 3.5	3.3 – 5 V	80 mA	80 mA
1	NodeMCU V2	3.3 – 5 V	80 mA	80 mA

Analizando la Tabla 4 se deduce que el consumo de corriente de cada robot del sistema es aproximadamente 1785 mA, por lo que se eligió una batería de dos celdas de 2200 mAh y 7.4V de capacidad del fabricante Tenenergy.



Figura 45. Batería LiPo 7.4V y 2200 mAh.

Utilizando la ecuación (16) se estima el tiempo de autonomía de la batería, con el objetivo de determinar el tiempo de operación del sistema y así planificar las tareas de reconocimiento y clasificación de objetos de color.

$$\text{Tiempo de autonomía} = \frac{\text{Capacidad batería}}{\text{Consumo sistema}} \quad (16)$$

Reemplazando los datos obtenidos anteriormente se tiene:

$$\text{Tiempo de autonomía} = \frac{2200 \text{ mAh}}{1785 \text{ mA}} = 1.23 \text{ h} = 73 \text{ min} \quad (17)$$

En base al resultado de la ecuación (17) cada robot del sistema puede operar por aproximadamente por 73 minutos, correspondiendo al consumo de la Tabla 4.

3.2.6 Wireless router

Un wireless router o enrutador inalámbrico es un dispositivo que administra el tráfico de información de una red, y combina las funciones de un punto de acceso (AP) y un router. Es decir puede conectar la red local a internet, y además conectar dispositivos inalámbricos a la red local.



Figura 46. Wireless router TP-Link modelo TL-WR840N.

En este proyecto se utiliza el router inalámbrico de la Figura 46 de marca China TP-Link como AP, sin conexión a Internet. Este modelo de router está diseñado para redes pequeñas, es ideal para streaming, y su tamaño está acorde con el escenario estructurado.

Además es compatible con el estándar IEEE 802.11b/g/n, posee un rendimiento inalámbrico de 300Mbps y es de fácil configuración (TP-Link, 2018).

Este proyecto requiere de una red local para comunicar el sistema multi-robot y el computador de procesamiento, la comunicación inalámbrica se realiza por medio de la tarjeta de desarrollo nodeMCU que incorpora el módulo WiFi ESP8266.

3.2.7 Computador de procesamiento

El computador de procesamiento es un computador portátil que es capaz de conectarse mediante WiFi al sistema multi-robot, y enviar órdenes para cumplir las tareas necesarias para completar el proceso de reconocimiento y clasificación de objetos de color. Además debe contar con recursos computacionales suficientes para soportar herramientas de software y algoritmos complejos.



Figura 47. Portátil MSI modelo GE62 2QD Apache.

Para este proyecto se usará la portátil gamer de la Figura 47 cuyas especificaciones técnicas se describen a continuación:

- Fabricante: MSI Inc.
- Modelo: Ge62 2QD Apache Pro.
- Pantalla: 15.6 pulgadas.
- Resolución de pantalla: 1920 x 1080 píxeles.
- Procesador: Intel® Core™ i7 5700HQ quinta generación @3.5GHz.
- Memoria RAM: 16GB.
- Memoria gráfica: GeForce GTX 960M de 2GB
- Tipo de sistema: 64 bits.
- Sistema operativo: Windows 10 pro.
- Batería: 6 celdas, 120 minutos de autonomía.

La computadora usada como elemento central de procesamiento permite un procesamiento de imagen, conexión inalámbrica y transmisión rápidos y eficientes debido a sus características de procesador de quinta generación, memoria RAM y tarjeta gráfica denotando un elemento apto para controlar un conjunto de robots independientes.

3.3 Software

El software del sistema multi-robot cooperativo para clasificación de objetos de color está implementado en tarjetas de desarrollo (teensy 3.5 y nodeMCU) y en un computador portátil, realizando los procesos visualizados en la Figura 48.

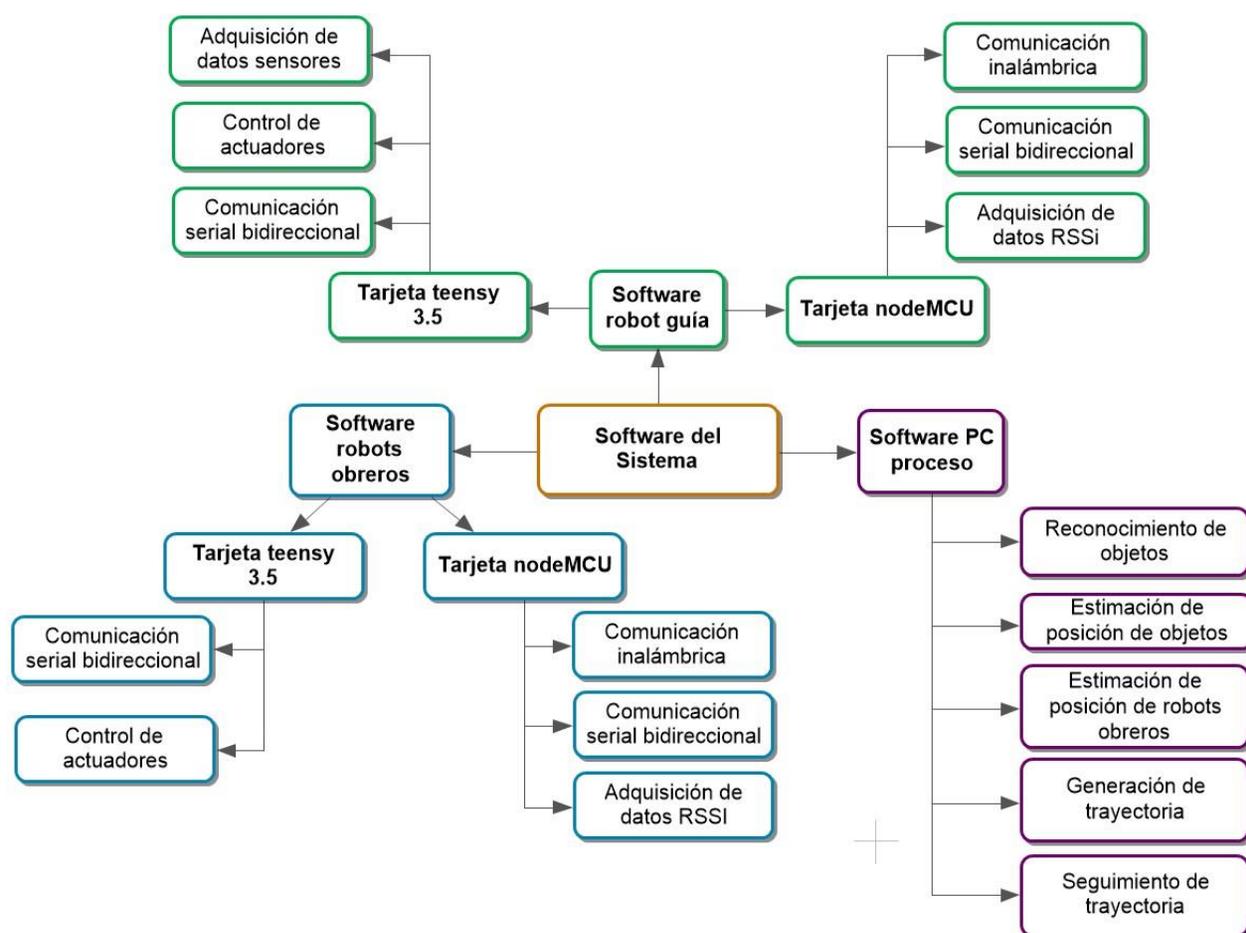


Figura 48. Diagrama de bloques del software del sistema.

Los algoritmos implementados en las tarjetas de desarrollo Teensy3.5 y nodeMCU fueron desarrollados en la interfaz de arduino IDE, y los algoritmos implementados en la PC portátil fueron realizados en Python.

El sistema cuenta con tres etapas de software: la primera etapa se encarga de estimar la posición de cada objeto de color en el escenario estructurado, la segunda etapa tiene como objetivo estimar la posición de cada robot del sistema dentro del escenario y finalmente la tercera etapa realiza la

planificación de trayectoria a seguir por los robots junto con el proceso de recolección de los objetos de color, ver Figura 49.

El proceso de conexión entre el PC, la cámara IP y la tarjeta de desarrollo nodeMCU indicado en la Figura 49, se realiza mediante comunicación inalámbrica bidireccional WiFi permitiendo el envío y recepción de datos entre los dispositivos del sistema.

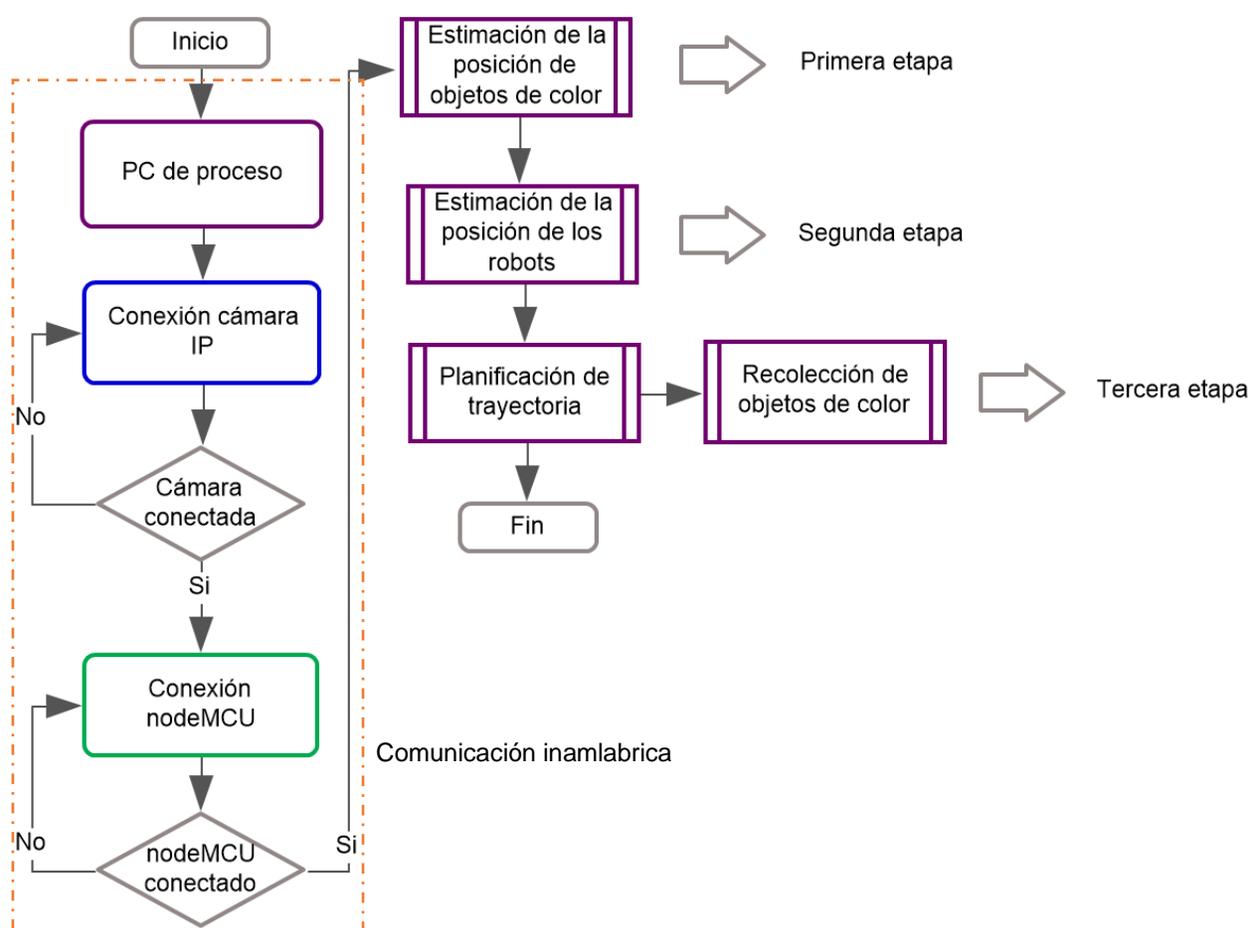


Figura 49. Diagrama de flujo del software del sistema.

3.3.1 Primera etapa

El algoritmo correspondiente a esta primera etapa realiza el proceso de estimación de posición de los objetos de color dentro del escenario estructurado, ver Figura 50. Inicialmente se realiza la búsqueda de objetos de color con el robot en estado estacionario, a los objetos detectados se les asigna prioridad dependiendo de su color; los objetos no detectados tienen menor prioridad.

A continuación se procede a centrar la imagen de los objetos en la pantalla enviando señales de control a los actuadores (motores) del robot guía. Una vez centrado el objeto de color se procede a la adquisición de datos de los sensores (distancia y ángulo), y se repite el procedimiento para los demás objetos de color. Finalmente con toda la información adquirida se estima la posición de cada objeto dentro del escenario estructurado.

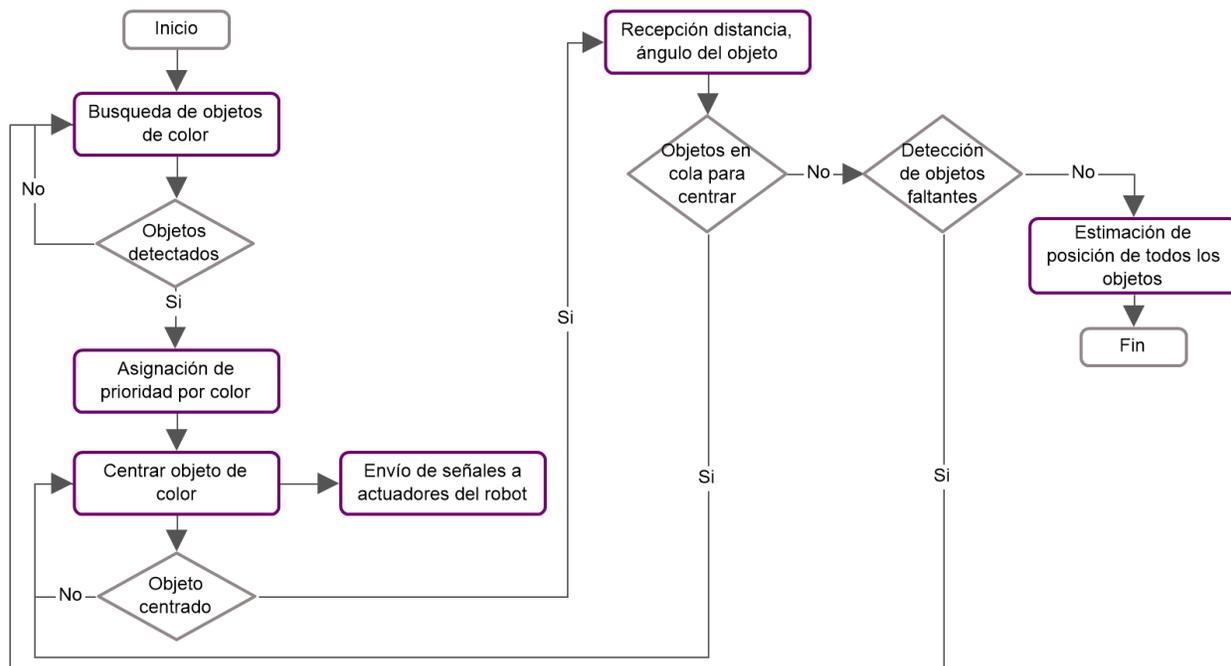


Figura 50. Diagrama de flujo del proceso de estimación de posición de los objetos de color.

El subproceso **“Envío de señales a actuadores del robot”** de la Figura 50, está desglosado en la Figura 51. Este subproceso se encarga de la transmisión bidireccional de datos entre la PC y el robot guía, con el objetivo de mover al robot hasta centrar la imagen capturada por la cámara IP en la PC y adquirir los datos de distancia y ángulo medidos por los sensores con referencia a un objeto de color. Para realizar la transmisión se comprueba que la conexión inalámbrica WiFi está activada, a continuación la PC envía señales de control a través de conexión WiFi a la tarjeta de desarrollo nodeMCU, que a su vez envía las señales de control mediante comunicación serial a la tarjeta teensy 3.5 quien se encarga de procesar estas señales. Son tres las señales de control que pueden ser transmitidas desde la PC al robot guía: “I” para que el robot se desplace a la izquierda, “D” indica que debe desplazarse a la derecha y “C” cuando la imagen está centrada, además la señal “C” activa la transmisión de datos de distancia y ángulo desde el robot guía hacia la PC.



Figura 51. Diagrama de flujo del proceso de envío de señales al robot desde la PC.

3.3.2 Segunda etapa

En esta segunda etapa se realiza el proceso de estimación de la posición de los robots dentro del escenario estructurado. Se implementa dos métodos de estimación basados en el valor RSSI, y se

selecciona el método con menor error. En la Figura 52 se observa un esquemático de la posible distribución de los robots en el escenario para el método de triangulación.

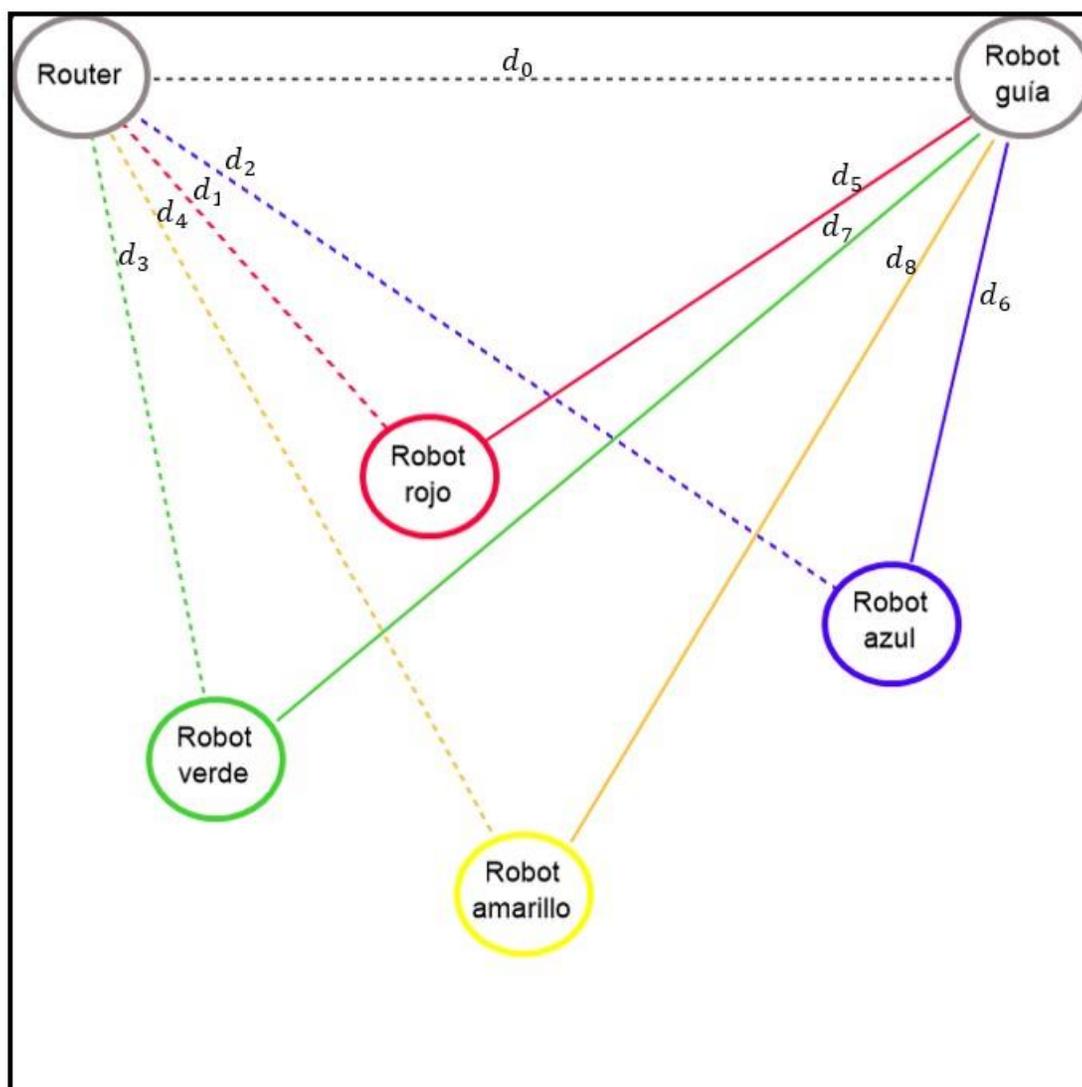


Figura 52. Esquemático de la posición de los robots dentro del escenario.

El algoritmo correspondiente a esta método realiza la estimación de posición de los robots mediante cálculos geométricos, para lo cual es necesario calcular las distancias (d) indicadas en la Figura 52.

El algoritmo de la Figura 53 primeramente verifica si está disponible la conexión WiFi, luego verifica que cada tarjeta nodeMCU se comuniquen mediante conexión WiFi, una vez verificado las conexiones la PC portátil procede a la adquisición de valores RSSI de las tarjetas nodeMCU respecto al router, seguido la PC adquiere los valores RSSI de las tarjetas nodeMCU respecto al robot guía. Con todos los valores RSSI adquiridos se estima distancias y mediante cálculos geométricos se determina la posición de los robots dentro del escenario.

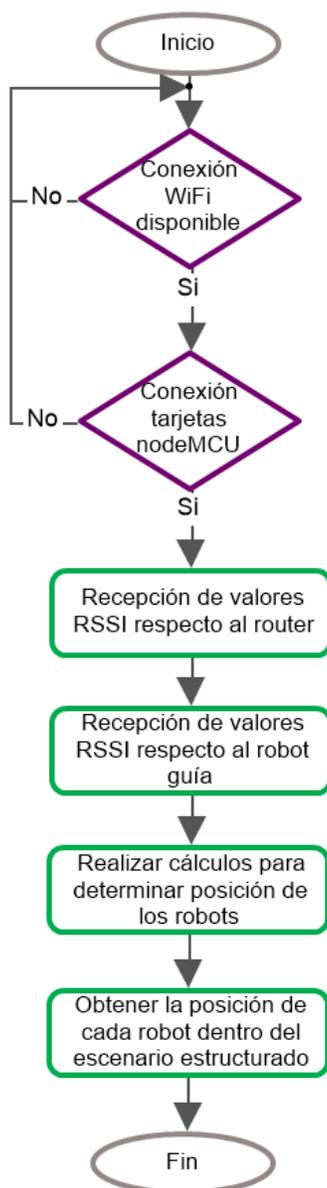


Figura 53. Diagrama de flujo del proceso de estimación de posición de los robots.

En la Figura 54 se observa un esquemático de la posible distribución de los robots en el escenario para el método de Fingerprints o huella digital.

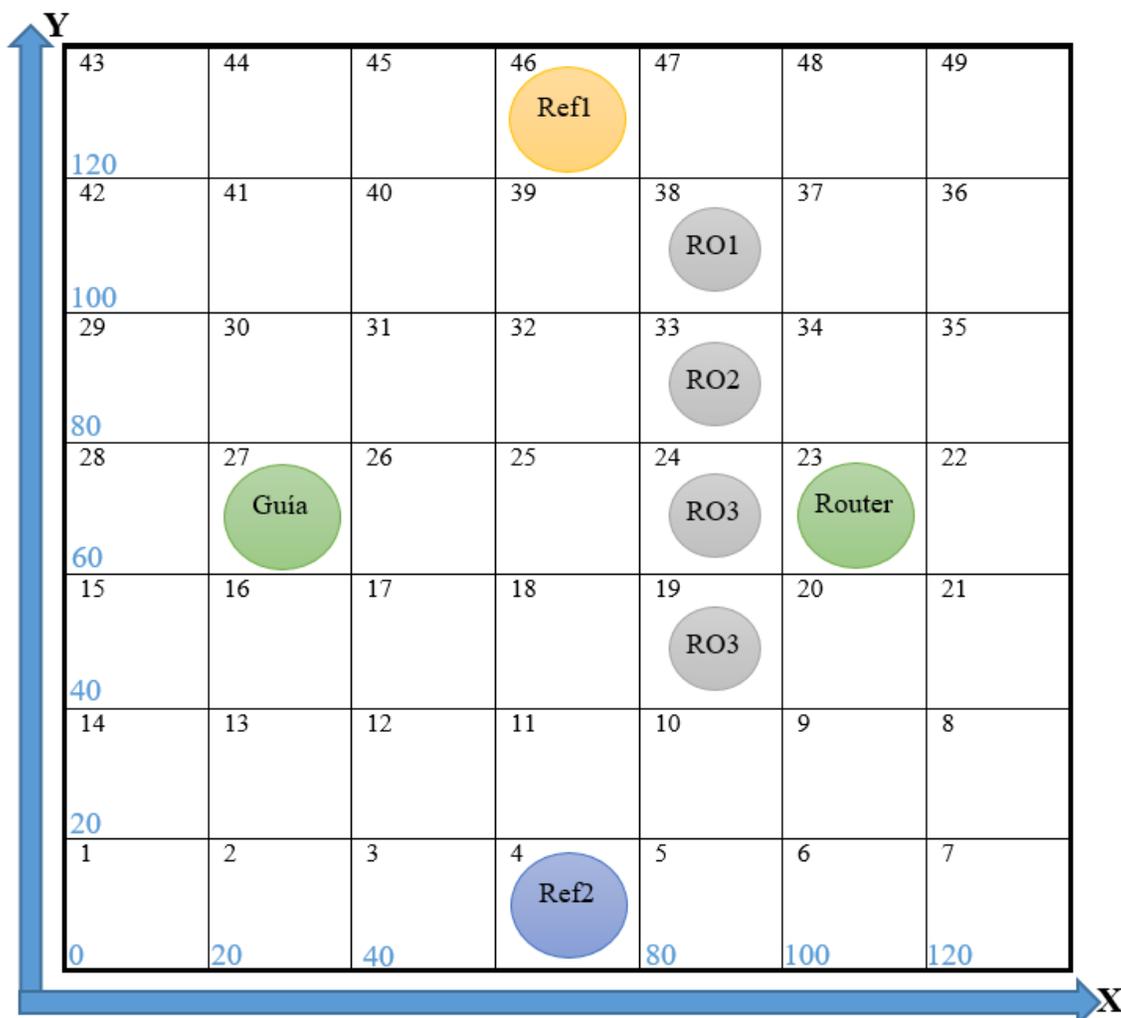


Figura 54. Esquemático de la posición de los robots dentro del escenario.

El algoritmo correspondiente a este método realiza la estimación de posición de los robots mediante comparación del valor RSSI de la señal actual respecto a una base de datos, para lo cual es necesario realizar mediciones previas en cada cuadrícula. El algoritmo de la Figura 55 es similar al algoritmo de la Figura 53.

Primeramente verifica si está disponible la conexión WiFi, luego verifica que cada tarjeta nodeMCU se comunique mediante conexión WiFi, una vez verificado las conexiones la PC portátil

procede a la adquisición de valores RSSI de las tarjetas nodeMCU respecto al router, seguido adquiere los valores respecto al robot guía, a continuación adquiere los valores respecto a la referencia de la pared amarilla y finalmente adquiere los valores respecto a la referencia de la pared amarilla. Con todos los valores RSSI adquiridos se estima distancias mediante cálculos geométricos simples y se determina la posición de los robots dentro del escenario.

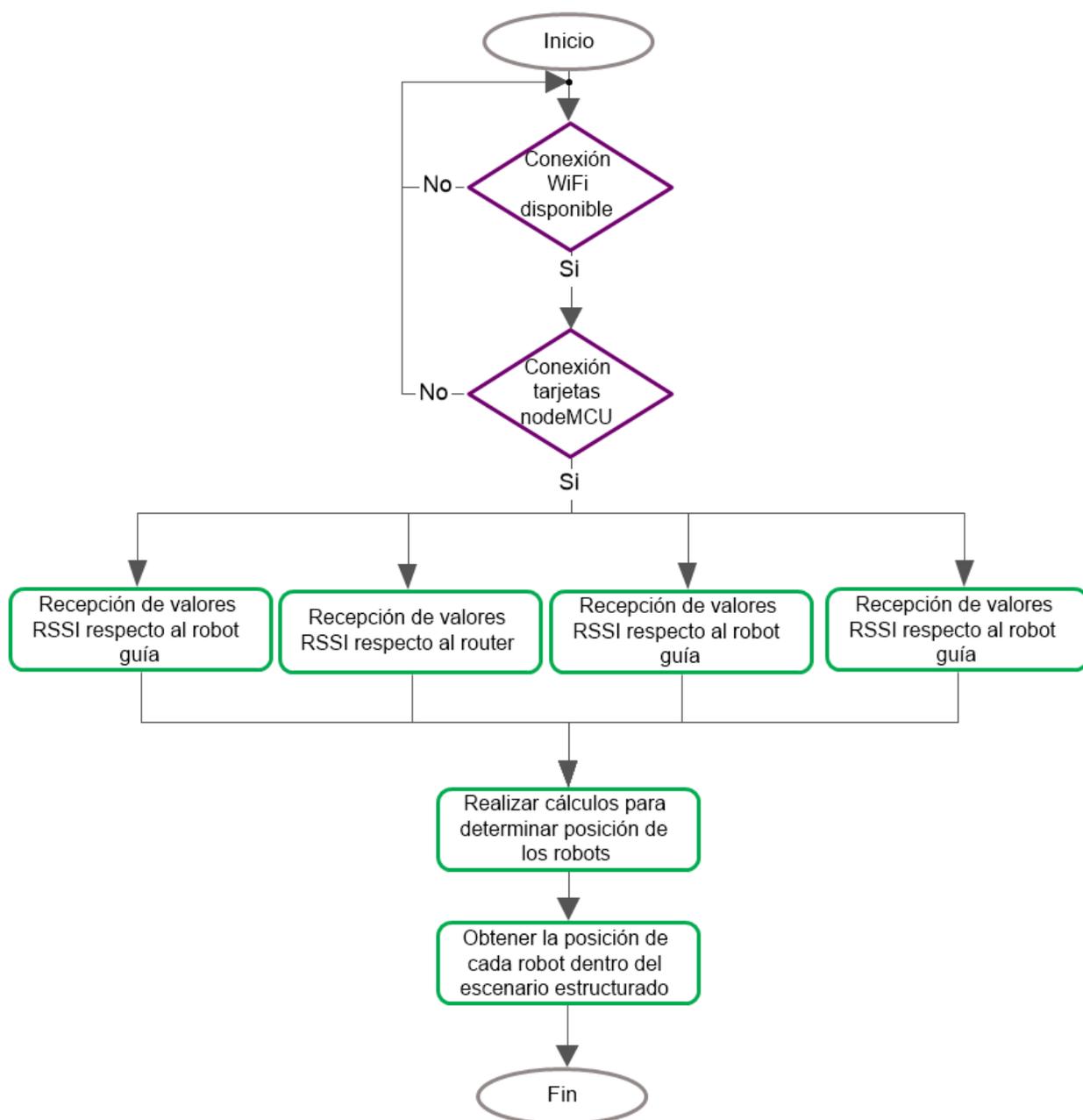


Figura 55. Diagrama de flujo del proceso de estimación de posición de los robots.

3.3.3 Tercera etapa

En la etapa final se realiza la planificación de trayectoria de los robots y la recolección de objetos de color dentro del escenario estructurado, el algoritmo a implementar se describe en Figura 56. De las etapas anteriores se obtiene la posición tanto de los objetos de color como de los robots dentro del escenario. Conocida esta información se procede a generar la trayectoria “A”, que parte de la posición inicial de los robots hasta la posición conocida de los objetos de color. El PC portátil envía señales de control a los actuadores de cada robot del sistema hasta que se complete la trayectoria planificada.

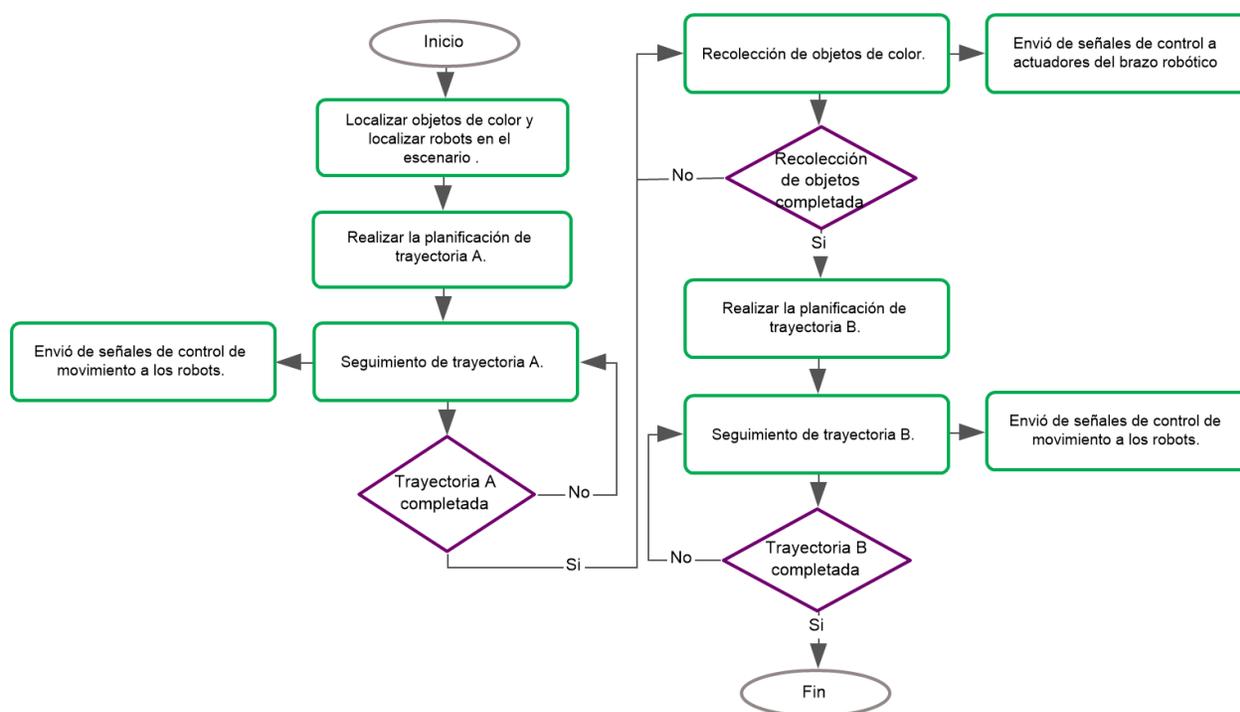


Figura 56. Diagrama de flujo del proceso de planificación y seguimiento de trayectoria.

Una vez culminada la primera trayectoria los robots proceden a recolectar los objetos de color, para lo cual el PC envía señales de control al actuador de cada brazo robótico del sistema multi-robot. En este subproceso, el brazo robótico rodea al objeto de color para un posterior desplazamiento.

A continuación, se genera la segunda trayectoria “B”, que parte de la posición de recolección de los objetos hasta la posición de almacenamiento de los mismos, es decir hasta las paredes de color. De forma similar a la generación de trayectoria anterior el PC envía las señales de control a los actuadores de cada robot del sistema hasta completar la trayectoria.

3.4 Problemas enfrentados en hardware y software

- *Chasis de los robots:* La altura del chasis no era la indicada para el montaje de todos los módulos del proyecto, por lo que se cambió los tornillos originales por otros de mayor largo.
- *Servomotores para orientación:* Se consideró usar servomotores para orientar la tarjeta nodeMCU debido a que la antena es direccional, es decir los valores RSSI varían dependiendo de la orientación de la antena.
- *Objetos de color:* Inicialmente se trabajó con esferas de colores, las mismas que fueron sustituidas por cilindros de color debido a su respuesta de medición ante el sensor de distancia HC-SR04.

CAPITULO IV

IMPLEMENTACIÓN DE HARDWARE Y SOFTWARE

4.1 Introducción

En este capítulo se desarrolla la implementación de algoritmos cooperativos para clasificación de objetos de color dentro de un escenario estructurado. Siendo implementados algoritmos de reconocimiento de objetos de color, estimación de posición de objetos y robots dentro del escenario estructurado, y algoritmos de planificación y seguimiento de trayectoria para clasificación de los objetos de color. Adicionalmente se presenta el diseño y construcción de los robots móviles del sistema, tanto del robot guía como de los robots obreros.

4.2 Implementación de hardware en los robots móviles del sistema

Los componentes de hardware seleccionados en el capítulo anterior son integrados en cada robot del sistema, cabe destacar que algunos componentes electrónicos y mecánicos varían para el robot guía y los robots obreros.

4.2.1 Robot guía

En la Tabla 5 se especifican los elementos implementados en el robot guía, los mismos son agrupados de acuerdo a su aplicación: adquisición, control y potencia. El montaje de estos elementos se realiza por etapas con el objetivo de evitar fallos por conexión.

Tabla 5
Hardware implementado en el robot guía.

Aplicación	Elementos	Tarea
Adquisición	Sensor HC-SR04	Detección de distancia de objetos de color en el escenario
	Sensor inercial MPU6050	Detección de desplazamiento angular del robot
	Cámara IP	Reconocimiento de objetos de color
Control y procesamiento	Teensy 3.5	Tarjeta dedicada a adquisición de datos (distancia y ángulo) y el control de actuadores
	NodeMCU V2	Tarjeta de comunicación inalámbrica WiFi. Generación del parámetro RSSI.
Potencia	Driver L298N	Control de motores DC
	Motores DC	Desplazamiento del robot
	Servomotores SG90	Control de orientación de la tarjeta nodeMCU y control del brazo robótico
	Regulador de Voltaje LM7805	Regulación de voltaje a 5(V)
	Baterías LIPO	Fuente de voltaje DC

En la primera etapa se realiza la conexión de los sensores con la tarjeta de desarrollo Teensy 3.5, en la segunda etapa se realiza la conexión serial entre las tarjetas Teensy 3.5 y nodeMCU, y finalmente se conecta la etapa de potencia: el regulador de voltaje LM7805, el driver L298N, los motores DC y la batería LiPo con las etapas anteriores. En la Figura 57 se presenta el diagrama de conexión electrónico de los elementos acoplados en el robot guía, todos los elementos empleados fueron descritos en la sección de Hardware.

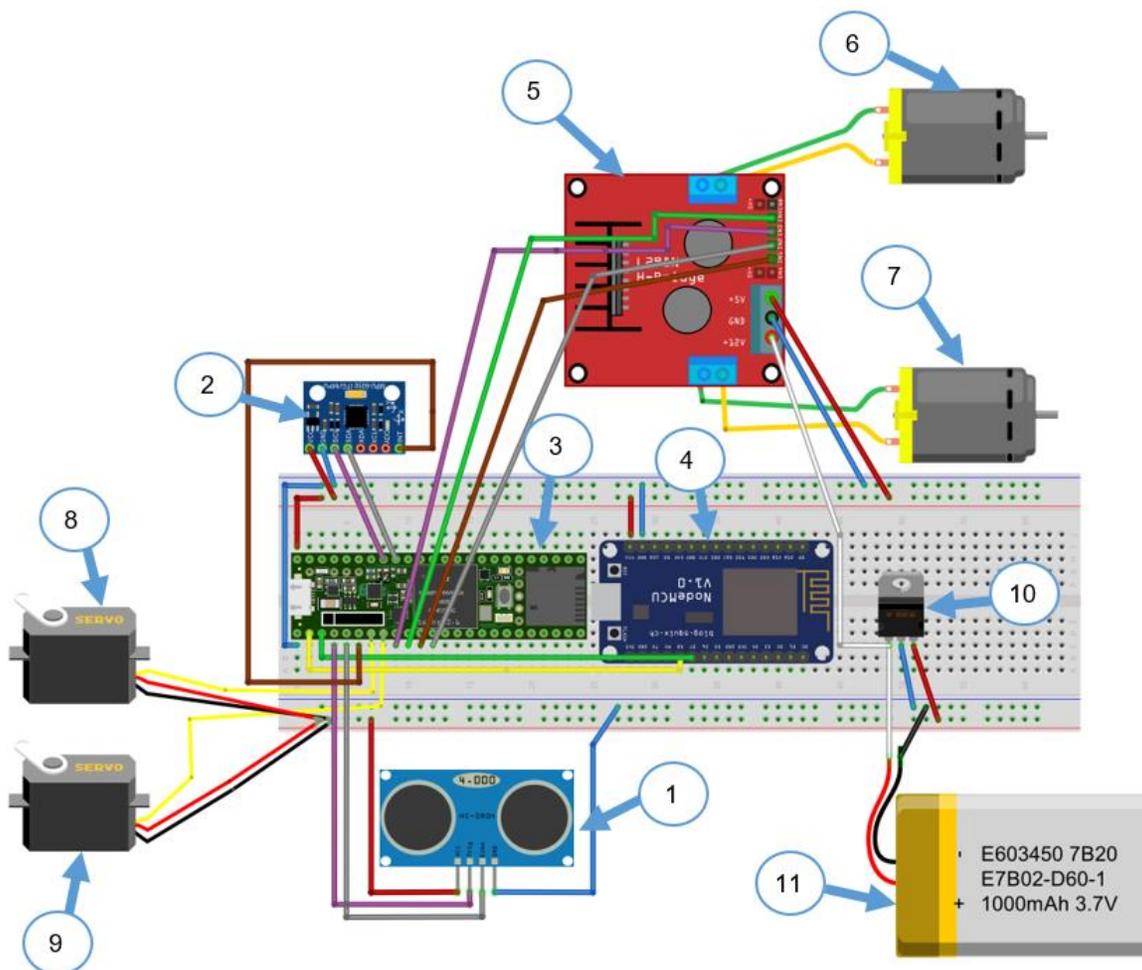


Figura 57. Diagrama de conexión del robot guía. (1) sensor HC-SR04, (2) sensor MPU6050, (3) Teensy 3.5, (4) nodeMCU V2, (5) driver L298N, (6) motor DC derecho, (7) motor DC izquierdo, (8) servomotor de cámara IP, (9) servomotor de orientación, (10) LM7805, (11) batería LiPo.

En la Figura 58 se observa la disposición final de los elementos acoplados en el chasis del robot guía, obteniendo una estructura que realice las tareas de detección de objetos de color dentro del escenario estructurado sin riesgo de desconexión o fallos.

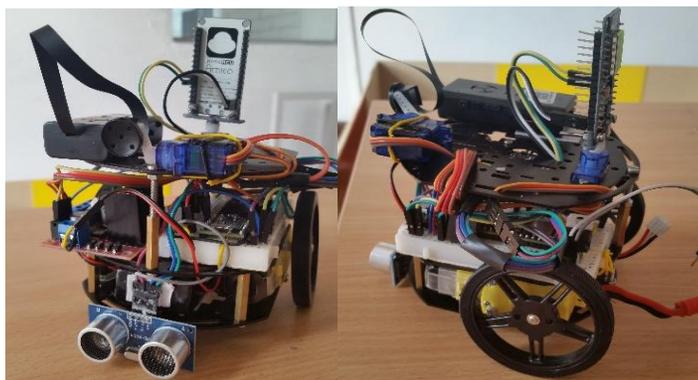


Figura 58. Robot guía.

4.2.2 Robots obreros

De forma similar al acoplamiento del robot guía, los elementos descritos en la Tabla 6 son implementados en el chasis de los robots obreros por etapas según su aplicación.

Tabla 6

Hardware implementado en los robots obreros.

Aplicación	Elementos	Tarea
Adquisición	Sensor inercial MPU6050	Detección de desplazamiento angular del robot
Control y procesamiento	Teensy 3.5	Tarjeta dedicada a adquisición de datos (distancia y ángulo) y el control de actuadores
	NodeMCU V2	Tarjeta de comunicación inalámbrica WiFi. Generación del parámetro RSSI.
Potencia	Driver L298N	Control de motores DC
	Motores DC	Desplazamiento del robot
	Servomotores SG90	Control de orientación de la tarjeta nodeMCU y control del brazo robótico
	Regulador de Voltaje LM7805	Regulación de voltaje a 5(V)
	Baterías LIPO	Fuente de voltaje DC

En la primera etapa se realiza la conexión del sensor inercial MPU6050 con la tarjeta de desarrollo Teensy 3.5, a continuación se realiza la conexión para comunicar por puerto serial las tarjetas nodeMCU y Teensy. Y finalmente se conecta la etapa de potencia, es decir se realiza las conexiones del driver L298N, los motores DC, los servomotores, el regulador LM7805 y la batería LiPo con las etapas anteriores. En la Figura 59 se visualiza el diagrama de conexión electrónico de los elementos acoplados en un robot obrero, el mismo modelo se utiliza para los demás robots obreros.

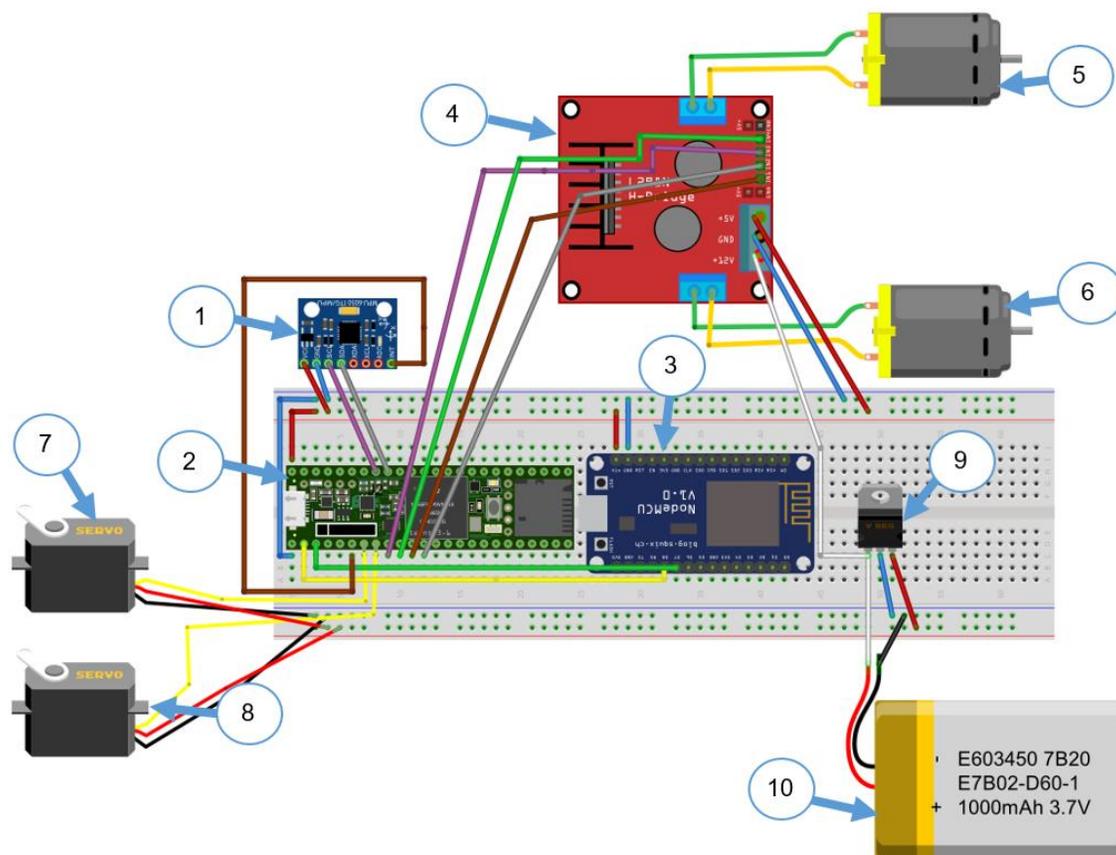


Figura 59. Diagrama de conexión de los robots obreros. (1) sensor MPU6050, (2) Teensy 3.5, (3) nodeMCU V2, (4) driver L298N, (5) motor DC derecho, (6) motor DC izquierdo, (7) servomotor del brazo robótico, (8) servomotor de orientación, (9) LM7805, (10) batería LiPo.

Finalmente en la Figura 60 se presenta la estructura de los robots obreros, la misma difiere del robot guía por la ausencia del sensor de distancia HC-SR04 y la cámara IP, además acopla un brazo robótico para desplazar los objetos de color dentro del escenario.

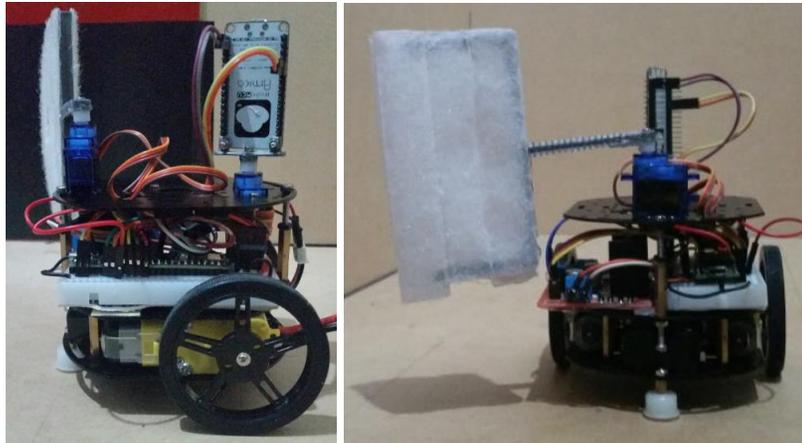


Figura 60. *Robots obreros*

4.3 Desarrollo del sistema de adquisición de datos

La tarjeta de desarrollo Teensy 3.5 es utilizada para la adquisición de datos de los sensores de distancia y desplazamiento angular. Los datos adquiridos son registrados y transmitidos a la tarjeta de desarrollo nodeMCU mediante comunicación serial y a su vez enviados al PC de procesamiento a través de comunicación inalámbrica WiFi, previa solicitud de envío y transmisión de datos del PC. La adquisición de datos de la cámara IP se realiza mediante la PC por comunicación inalámbrica, la información es registrada y almacenada para posteriores procesamientos.

A continuación, se detalla las configuraciones necesarias para establecer la comunicación entre los elementos del sistema multi-robot, además del control de actuadores.

4.3.1 Configuración de la tarjeta teensy 3.5

La tarjeta Teensy 3.5 es programada en el IDE de arduino versión 1.8.4, por lo que se instala el complemento teensyduino versión 1.42 necesario para trabajar con la tarjeta desde el IDE de arduino. La tarjeta dispone de seis puertos de comunicación serial, siendo necesario usar solo uno para establecer la comunicación serial con la tarjeta nodeMCU.

La comunicación serial permite la transmisión y recepción de datos entre las tarjetas **Teensy** encargada de adquisición de datos y control de actuadores, y **nodeMCU** encargada de establecer la comunicación inalámbrica en el sistema, a una velocidad de transmisión de 115200 bps. En el diagrama de flujo de la Figura 61 se detalla la comunicación serie realizada por la tarjeta Teensy 3.5.

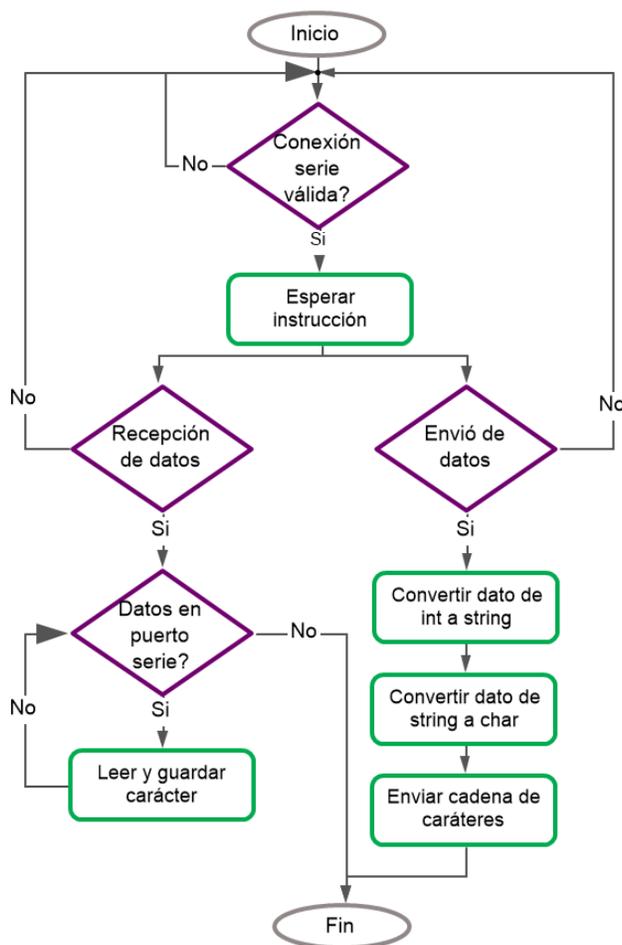


Figura 61. Diagrama de flujo del proceso de la comunicación serial de la tarjeta teensy 3.5.

La tarjeta Teensy se encarga de la adquisición y almacenamiento de datos de los sensores: HC-SR04 y MPU6050. Mediante una petición emitida por la PC la tarjeta Teensy realiza la lectura de los sensores, almacena los datos obtenidos y los transmite a la PC para su posterior procesamiento. Además controla el accionamiento de los motores (derecho e izquierdo) del robot mediante señales PWM enviadas al driver L298N.

En la Figura 22 se detalla el procedimiento de adquisición y transmisión de datos de los sensores implementados en el robot guía, así como el control de los actuadores.

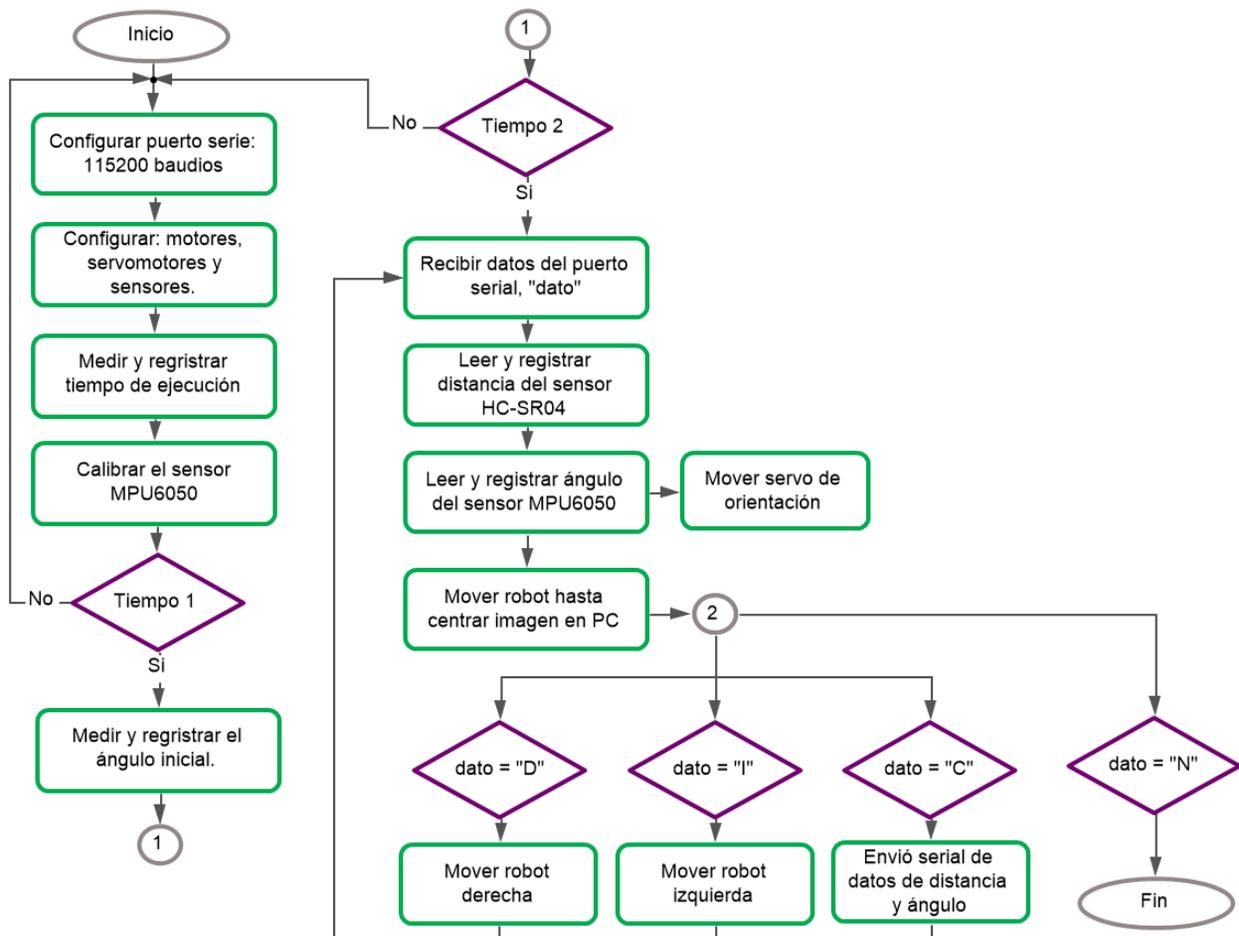


Figura 62. Diagrama de flujo del proceso de adquisición de datos para el robot guía.

En base al proceso de adquisición y transmisión de datos del robot guía se realiza el procedimiento a seguir para los robots obreros especificado en la Figura 63, así como el control de los actuadores.

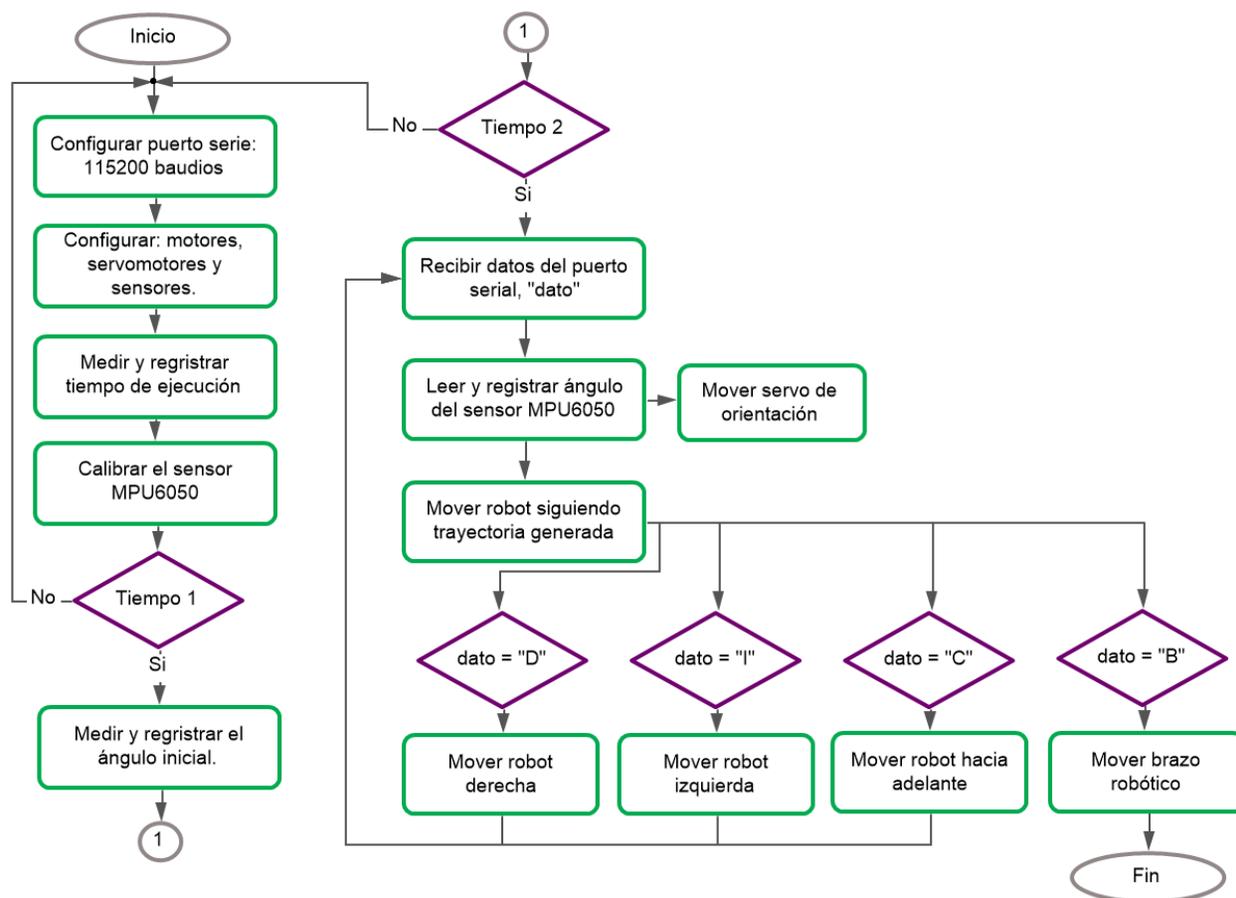


Figura 63. Diagrama de flujo del proceso de adquisición de datos para los robots obreros.

4.3.2 Configuración de la tarjeta nodeMCU

La tarjeta nodeMCU es programada en el IDE de arduino, y está conectada a la tarjeta Teensy por comunicación serial y a la PC por conexión inalámbrica. La velocidad de transmisión serial es la misma de la tarjeta Teensy, es decir 115200 bps. En la Figura 64 se describe la recepción y transmisión serial realizada por la tarjeta nodeMCU con respecto a la tarjeta Teensy.

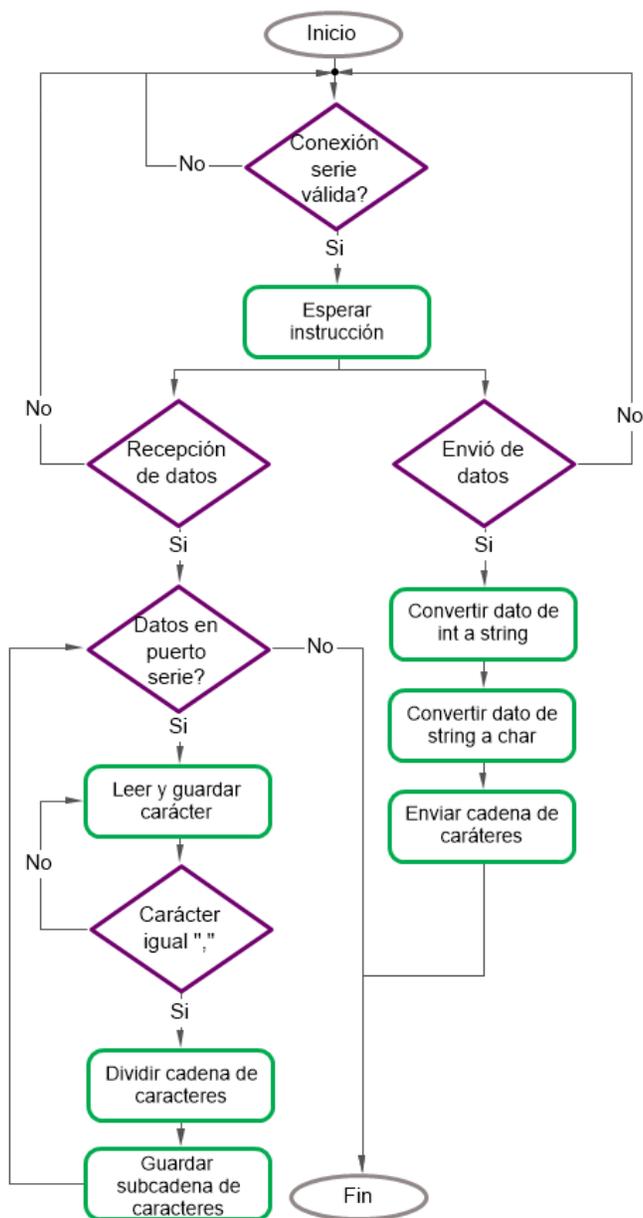


Figura 64. Diagrama de flujo del proceso de comunicación serial de la tarjeta nodeMCU.

Además la tarjeta nodeMCU se encarga de comunicar inalámbricamente los robots con la PC, el procedimiento está detallado en la Figura 65. Cabe destacar que de esta tarjeta se adquiere el valor RSSI, el mismo que es transmitido inalámbricamente a la PC y utilizado para estimar la posición de los robots en el escenario.

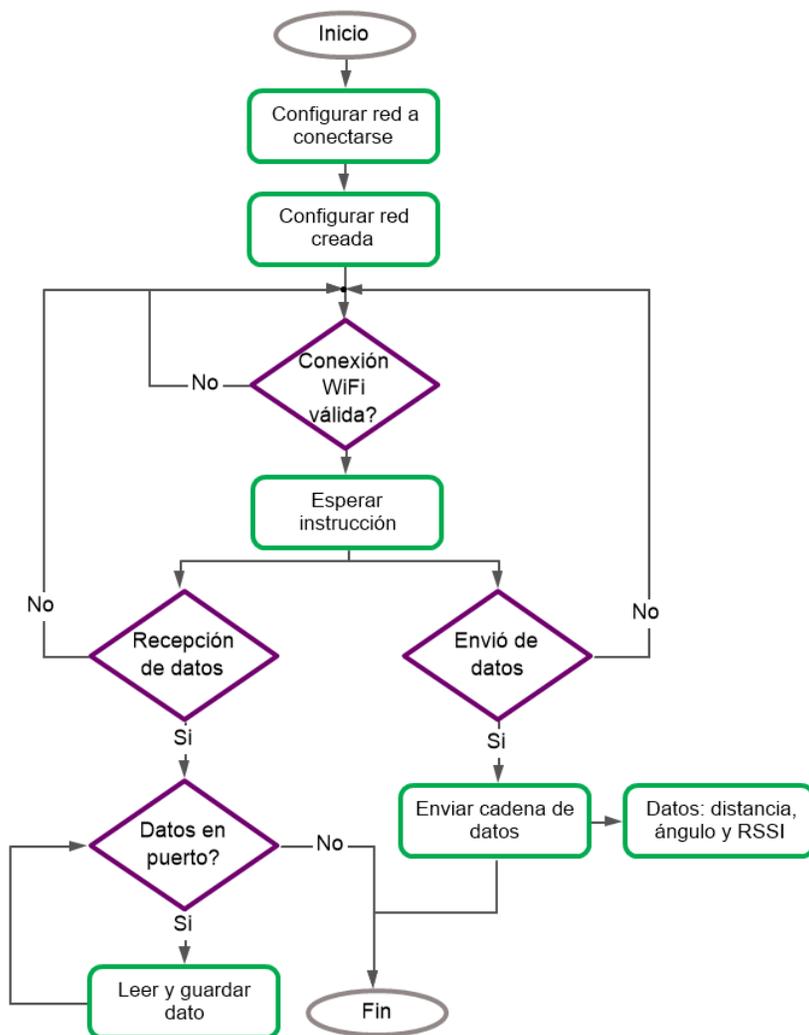


Figura 65. Diagrama de flujo del proceso de adquisición de datos para los robots obreros.

4.3.3 Configuración de la cámara IP

Inicialmente la cámara IP crea una red ad-hoc que permite la conexión con un dispositivo sea este una PC o un teléfono inteligente, se accede a la cámara mediante la dirección IP por defecto “192.168.10.1” con usuario “admin” y contraseña en blanco.

Para configurar la cámara IP es necesario instalar el software Search Tool, que permite asignar una dirección IP estática a la cámara, ver Figura 66. A continuación se realiza la conexión de la cámara IP con la red “TP-Link_Tesis” creada por el router.

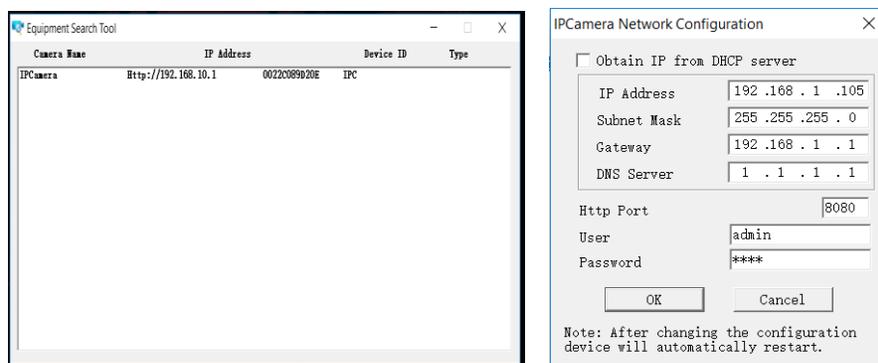


Figura 66. Asignación de dirección IP estática a la cámara.

Para verificar si se realizó correctamente la asignación de la dirección IP estática en la cámara, se teclea en el navegador la dirección establecida: “192.168.1.105:8080” como se observa en la Figura 67. La adquisición de frames de la cámara se realiza mediante programación en Python en el IDE PyCharm.

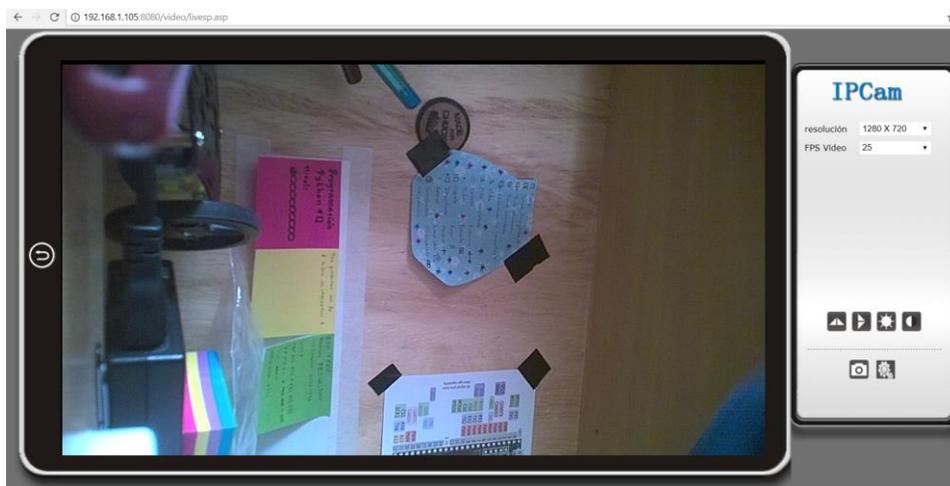


Figura 67. Conexión de la cámara IP a la red.

4.4 Desarrollo del sistema de procesamiento de datos

La PC se encarga del procesamiento de los datos adquiridos de los sensores (HC-SR04, MPU6050, y cámara IP) acoplados en los robots del sistema, así como de los valores RSSI obtenidos de las tarjetas nodeMCU, e implementa algoritmos de detección y reconocimiento de objetos de color, algoritmos de localización de objetos de color y robots en el escenario estructurado, y algoritmos de planificación y seguimiento de trayectoria. Además la PC se encarga de enviar caracteres a la tarjeta de control Teensy 3.5 para el control de los actuadores.

En la PC de proceso se utiliza el lenguaje de programación Python versión 2.7.12 con el IDE Pycharm debido a que posee una sintaxis sencilla, y trabaja tanto Linux como en Windows, además se instala la librería de visión artificial OpenCV versión 3.1.0 y la librería Socket para realizar la comunicación inalámbrica.



Figura 68. Íconos correspondientes a Python y OpenCV.

4.4.1 Procesamiento de datos obtenidos de los sensores

La PC de proceso obtiene los valores de los sensores mediante comunicación inalámbrica WiFi establecida entre la tarjeta de adquisición nodeMCU y la PC de proceso. Los valores obtenidos de los sensores presentan variabilidad por lo que se optó en realizar un filtro de moda de una n cantidad

de datos. Al inicial el sistema se estableció un tiempo de calibración de 20 segundos para los sensores, en esencial para el sensor de medición inercial MPU6050. Se observa que transcurrido el tiempo de calibración los valores del sensor MPU6050 se estabilizan.

Con los valores obtenidos de los sensores HC-SR04, MPU6050 y cámara IP, se implementa el algoritmo de detección y localización de objetos de color en el escenario estructurado. De esta primera etapa de implementación se obtienen las posiciones respecto a (x, y) de cada objeto de color en el escenario. La tarjeta de control Teensy 3.5 acoplada en el robot guía recibe un carácter desde la PC de proceso para el control de los motores DC, siendo los caracteres de control “D” (derecha), “I” (izquierda) y “C” (centro).

En los robots obreros los valores del sensor MPU6050 controlan el movimiento del servo de orientación para la tarjeta nodeMCU. De forma similar al robot guía reciben caracteres enviados desde la PC para el control de los actuadores, siendo “T” el carácter dispuesto para que el robot avance de frente, “U” para el avance hacia la derecha, y “V” para que avance hacia la izquierda. Adicionalmente reciben el carácter “Z” para el control del servo del brazo robótico.

4.4.2 Procesamiento de datos obtenidos de la tarjeta nodeMCU

Los valores RSSI adquiridos de la tarjeta nodeMCU mediante comunicación WiFi presentan variabilidad aun si el robot está en estado estacionario, como se observa en la Figura 69. Para disminuir el error de medición se aplica filtro de moda a los datos obtenidos.

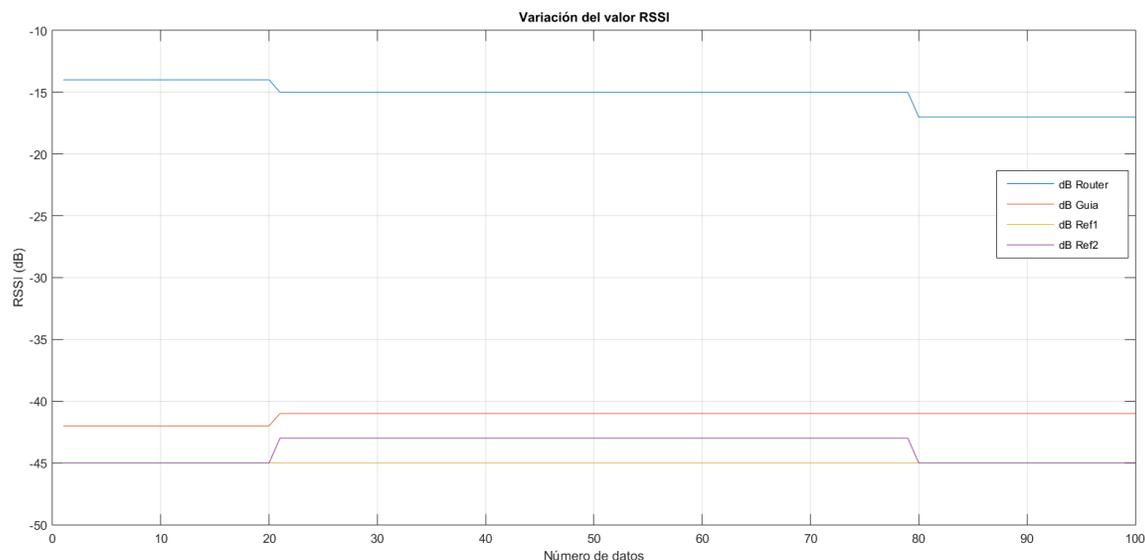


Figura 69. Medición del valor RSSI cuando el robot está en estado estacionario.

Con los valores RSSI filtrados se realizan los cálculos de estimación de posición de los robots dentro del escenario utilizando los dos métodos de estimación descritos en el capítulo tres, en la PC de proceso.

4.5 Control centralizado para el sistema multi-robot cooperativo

El control centralizado del sistema multi-robot está desarrollado en base a una arquitectura de red estrella, en la que todos los elementos del sistema están conectados mediante tecnología WiFi. Los algoritmos de localización de objetos de color y localización de robots en el escenario estructurado, y los algoritmos de planificación y seguimiento de trayectoria se ejecutan en un solo PC de procesamiento, es decir el sistema cooperativo implementa un control central, como se observa en la Figura 70.

Las tareas cooperativas entre los cinco robots del sistema están enfocadas en conocer la localización de todos los robots del sistema dentro del escenario estructurado, con el fin de clasificar los objetos de color según la distancia a las que se encuentren de los mismos. Es decir los robots se desplazan hasta el objeto más cercano y luego lo transportan a la pared de color correspondiente.

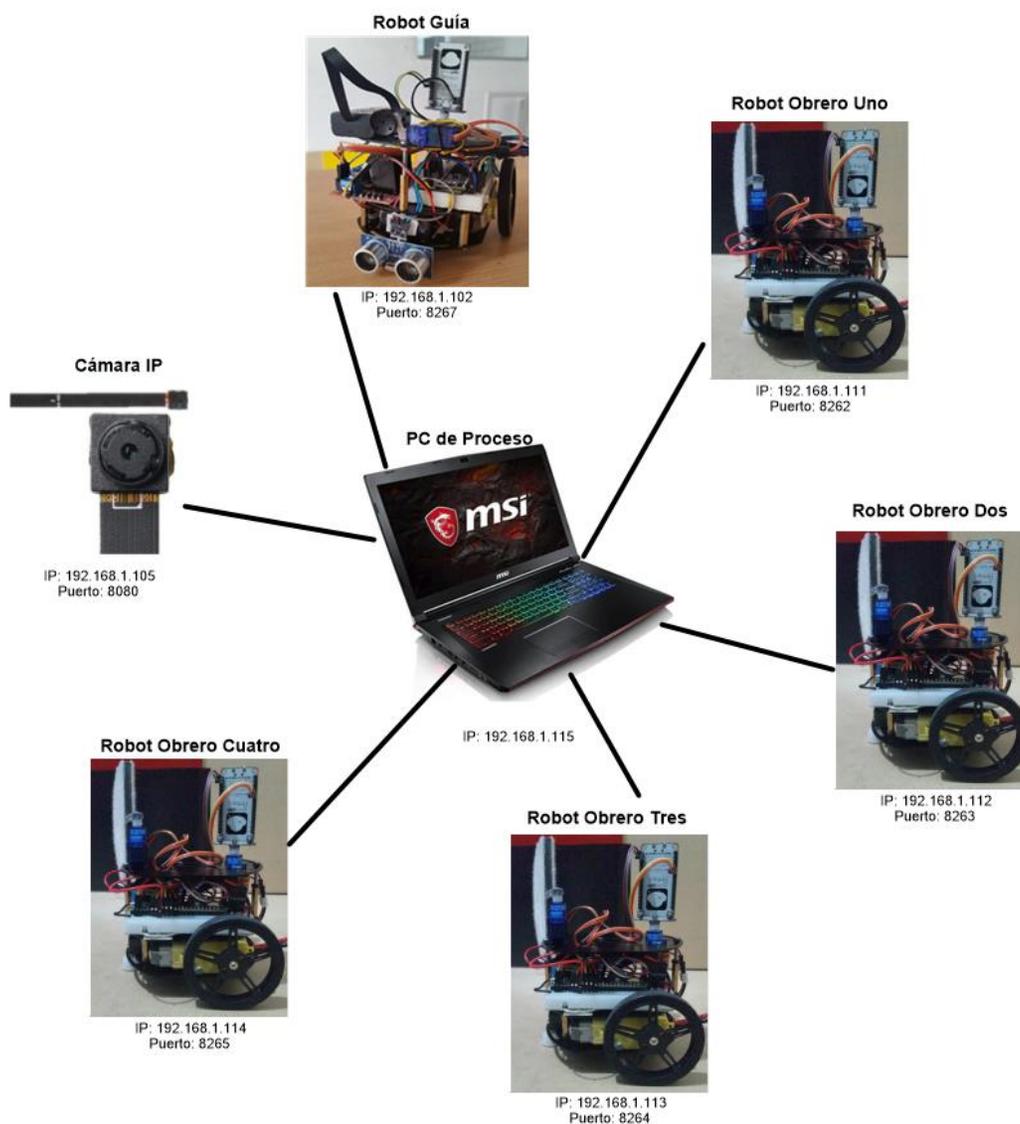


Figura 70. Control central del sistema cooperativo y asignación de IP's.

La comunicación del sistema multi-robot cooperativo se realiza mediante una red inalámbrica WiFi, donde los elementos del sistema tienen asignadas IP estáticas como se visualiza en la Figura 70.

4.6 Implementación de algoritmos

Los algoritmos implementados en el PC control central del sistema multi-robot móvil fueron los siguientes: algoritmo de detección de objetos de color mediante visión artificial para la localización de objetos en el escenario, algoritmo Fingerprints o huella digital para la localización de robots en el escenario, algoritmo RRT para generar trayectorias y una versión simplificada del algoritmo Pure Pursuit o persecución pura para seguimiento de trayectoria. En los robots móviles se implementó algoritmos para el control de los motores DC, algoritmos para el control de los servomotores, y algoritmos para comunicación serial y WiFi.

4.6.1 Algoritmo de localización de objetos de color

Para encontrar la posición de los objetos de color en el escenario estructurado, se utilizó visión artificial, sensado de distancia y sensado de la posición angular. Para detectar un objeto en base a su color es necesario extraer el rango de valores de dicho color; el modelo de color HSV permite establecer fácilmente el rango.

La cámara IP proporciona imágenes en el espacio de color RGB, por lo que inicialmente el algoritmo debe transformar las imágenes al espacio HSV. Para determinar el rango de valor correspondiente a cada color se utilizó un algoritmo de filtrado basado principalmente en la función **inRange()**. En la **Tabla 7** se observa el rango de valores superior e inferior de cada color, el cual fue obtenido experimentalmente como se visualiza en la **Figura 71** Además en la se observa el

resultado de filtrar un color, en este caso el amarillo. Cabe mencionar que obtener los rangos en HSV para el color rojo es difícil, por lo su detección no es muy buena en comparación con los demás colores.

Tabla 7

Rango de valores HSV para cada color utilizado.

Color	Rango inferior	Rango superior
Rojo	0, 136, 128	226, 175, 255
Azul	93, 67, 66	160, 255, 255
Verde	46, 88, 40	86, 255, 255
Amarillo	11, 105, 97	42, 250, 255

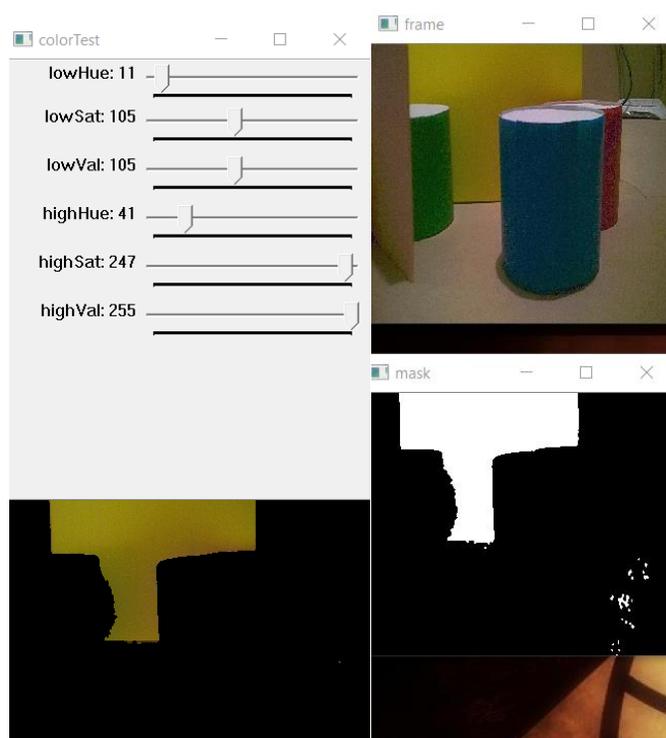


Figura 71. Obtención del rango de valores para el color amarillo.

Una vez obtenidos los rangos de valores HSV para todos los colores, se asigna prioridades de acuerdo al estado del arte. Los objetos de color son detectados uno a la vez en base a su prioridad

y luego son centrados en pantalla, cuando el objeto de color es detectado se envía una señal desde el PC de proceso a la tarjeta Teensy para el control de los actuadores.

Los sensores HC-SR04 y MPU6050 transmiten sus valores a la PC solo cuando la imagen del objeto de color es centrada. En la Figura 72 y la Figura 73 se observa el proceso de detección y localización de los objetos de color en el escenario estructurado. Adicionalmente se puede revisar la Figura 62 de la sección 4.3 y la Figura 50 de la sección 3.3.1

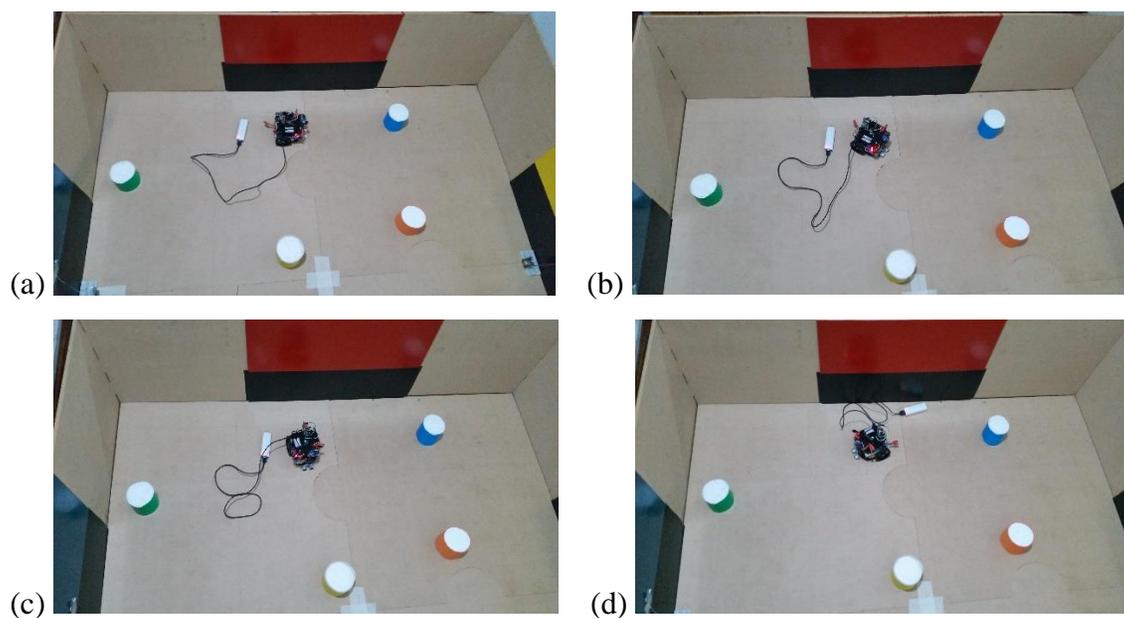
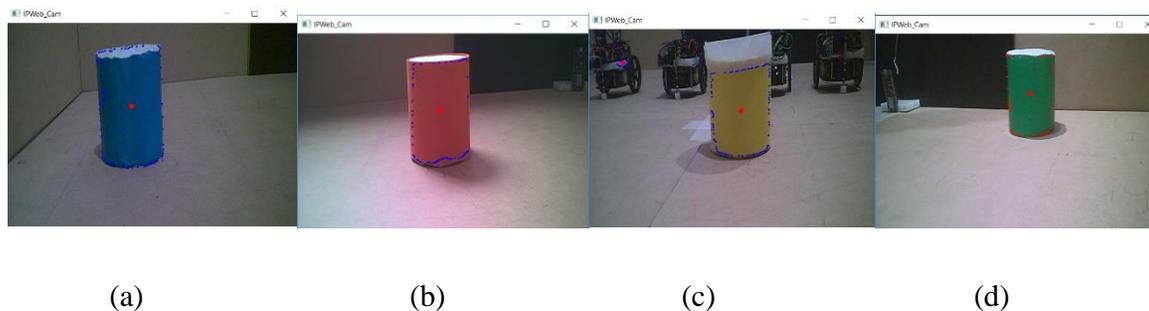


Figura 72. Secuencia de localización de objetos de color vista aérea.



(a) (b) (c) (d)

Figura 73. Secuencia de localización de objetos de color vista cámara IP.

4.6.2 Algoritmo de localización Fingerprints

El algoritmo de Fingerprints requiere una base de datos con mediciones previas del valor RSSI en el escenario de pruebas. Para lo cual el escenario es dividido en 49 celdas de 20x20 cm como se visualiza en la Figura 54 de la sección 3.3.2 y en cada celda se adquirió el valor RSSI con relación a los cuatro AP's.

Se realizaron en total tres pruebas de medición, obteniendo 300 datos en cada celda del escenario por prueba. Los datos fueron analizados usando filtro de moda, promedio y filtro de media; seleccionando los datos con menor error respecto a la medición real. En la Figura 74 se observa la secuencia de toma de datos y la orientación de los robots (flecha negra).

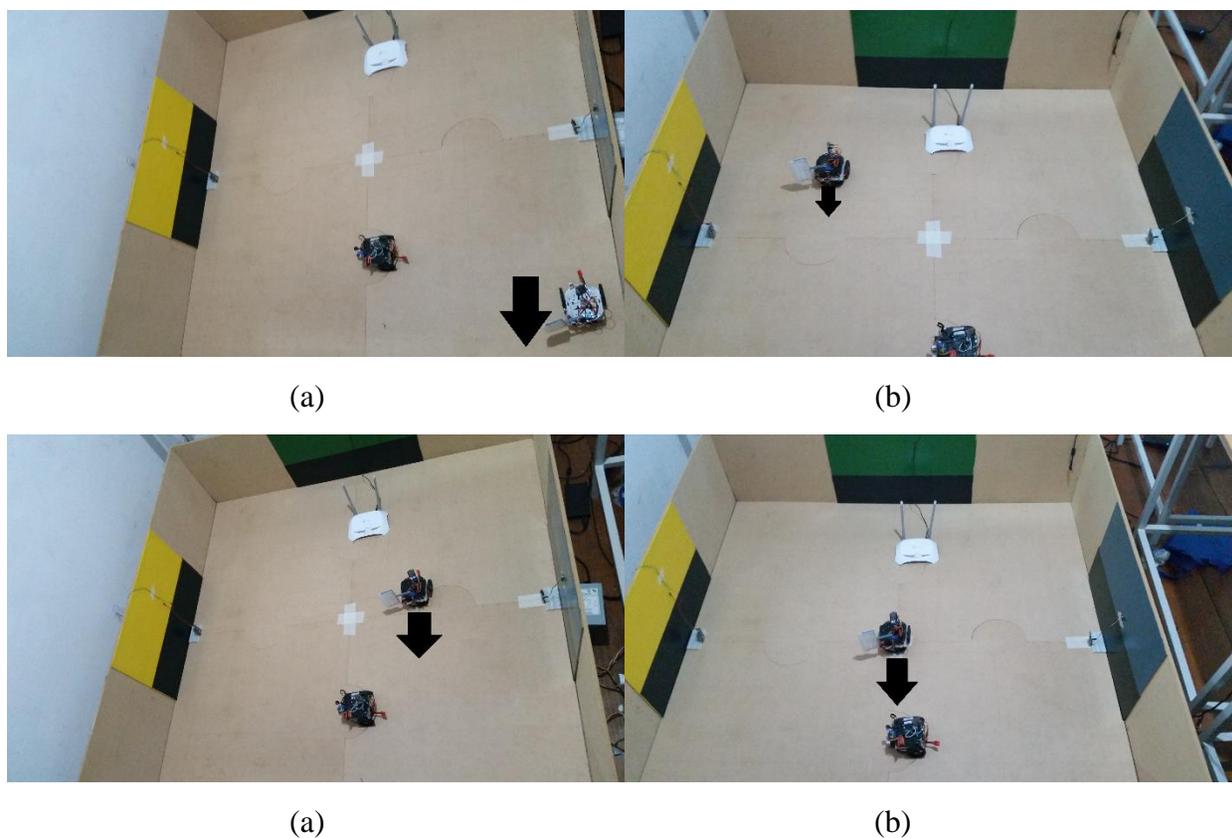


Figura 74. Secuencia de toma de datos método Fingerprints celdas (a) 1, (b) 38, (c) 31, (d) 25

En el algoritmo Fingerprints se estima la posición de los robots a través de una comparación entre el valor actual de RSSI y la base de datos, en la Figura 55 se detalla el algoritmo.

4.6.3 Algoritmo RRT para generación de trayectoria

Se implementa una versión simple del algoritmo RRT descrito en la sección 2.6; inicialmente requiere tres parámetros: punto de partida, punto de llegada, y una lista de obstáculos. En la Figura 75 se observa la generación aleatoria de tres trayectorias diferentes para dos mismos puntos.

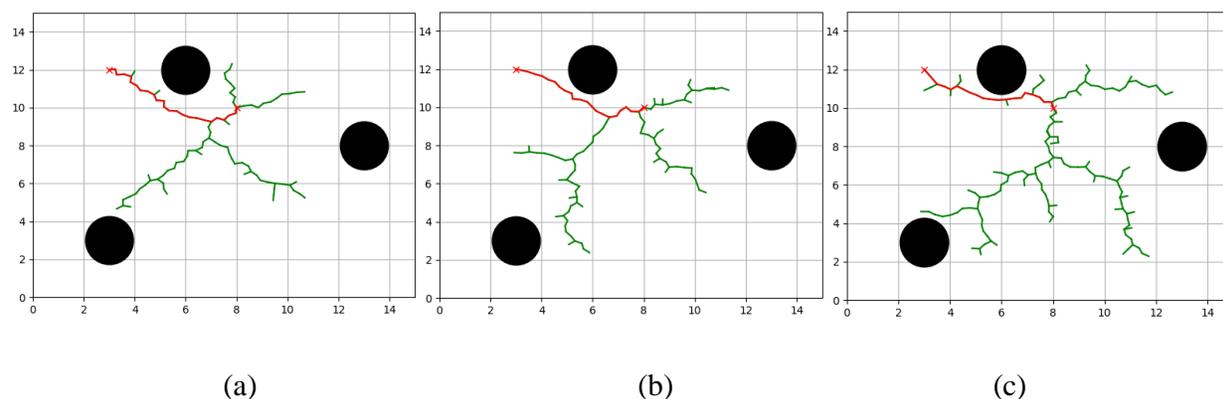


Figura 75. Generación aleatoria de trayectoria

Los parámetros iniciales de la Figura 75 corresponden a una de las posibles configuraciones del escenario donde: el punto inicial es la posición del robot más cercano al objeto azul (8,10; celda número 33), el punto final es (3,12; celda 44) y los obstáculos son los AP's (router, robot guía, refl1 y refl2), los demás robots y objetos color.

Se ha determinado un tamaño de ramificación mínima para el algoritmo RRT de $0,3 \text{ cm}$, en un área escalada de 150 cm^2 a 15 cm^2 del escenario. Se escalaron los valores del escenario para tener un mejor manejo del algoritmo RRT.

4.6.4 Algoritmo Pure Pursuit para seguimiento de trayectoria

Para enfrentar el seguimiento de la trayectoria generada se aplicó el algoritmo Pure Pursuit. Este algoritmo es de seguimiento de ruta y funciona calculando la velocidad angular y velocidad lineal que se le da al robot móvil.

La velocidad angular se utiliza mientras el robot gira al punto de mira para dejarlo frente a él, donde se aplica la velocidad lineal, para ello debe ser dotado de algunos parámetros como la posición y orientación iniciales.

Los parámetros que necesita el algoritmo son:

- La actitud inicial del robot. (posición y orientación)
- Velocidad lineal
- La distancia de mira frontal
- La ruta original

El algoritmo mueve el punto de mira hacia adelante en la ruta como se puede apreciar en la

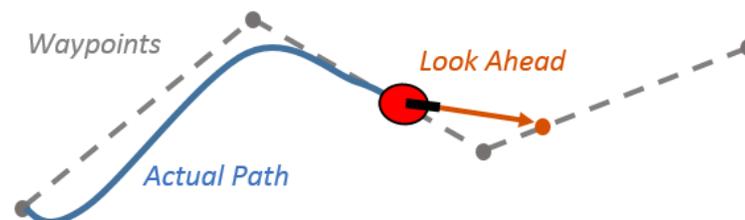


Figura 76. Seguimiento de ruta con el robot y punto de mira siguiente.

Se debe tomar en cuenta que el seguimiento no corresponde a los puntos intermedios, pero es una aproximación que permite continuar la ruta a pesar de no tocar todos los puntos en el camino.

Para la implementación del algoritmo se utilizaron como parámetros iniciales para el caso de la Figura 77:

- *La posición y orientación del robot.* Se han utilizado como posición inicial el punto (8.5, 10.5) correspondiente al casillero 38 mencionado anteriormente.
- *La velocidad lineal.* El valor de 2.5 cm/s
- *La distancia de mira frontal.* Con un valor de 8 cm
- *La ruta original.* Generada mediante RRT entre el punto inicial en (8.5, 10.5) y el punto final en (3, 11.8)

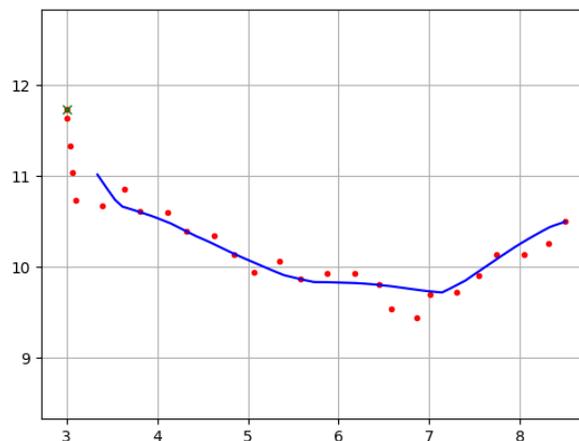


Figura 77. Seguimiento de trayectoria con el algoritmo Pure Pursuit para dos puntos del escenario.

Como se puede observar en la Figura 77 el algoritmo permite que el robot siga la ruta sin necesidad de pasar por todos los puntos intermedios, esto lo hace óptimo para ser utilizado con el algoritmo RRT que realiza ramificaciones dando un comportamiento más suavizado al seguimiento.

CAPITULO V

PRUEBAS EXPERIMENTALES DE FUNCIONAMIENTO Y RESULTADOS

5.1 Introducción

En este capítulo se evalúa el desempeño del sistema multi-robot cooperativo en el escenario estructurado, a través de una serie de pruebas experimentales detalladas más adelante.

Al realizar mediciones de visión artificial la iluminación del escenario es una de las condiciones más importantes, razón por la cual las pruebas fueron realizadas en la tarde y noche. Para estas pruebas se evaluó la detección de los objetos en dos diferentes condiciones de iluminación.

Debido a la naturaleza variante de la señal WiFi se realizó pruebas de medición del valor RSSI con respecto a cuatro AP's para ambos métodos descritos en la sección 3.3.2 las pruebas se replicaron para cada robot obrero del sistema.

Los datos obtenidos de cada prueba son analizados y presentados en gráficas, con la finalidad de evaluar el funcionamiento del sistema multi-robot cooperativo.

5.2 Pruebas realizadas al algoritmo de detección de objetos de color

La iluminación es un factor influyente en la detección de objetos de color, razón por la cual la experimentación se realizó en base a dos condiciones de iluminación:

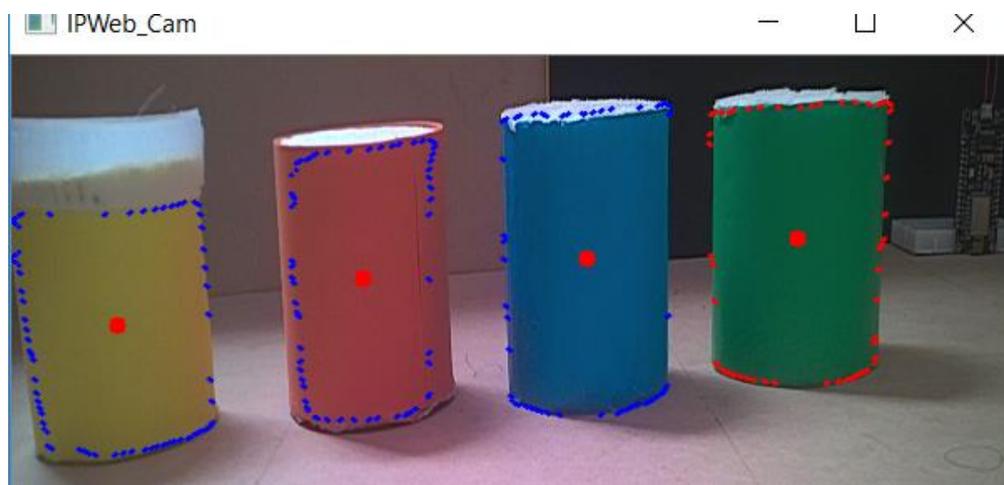
- En la tarde: Iluminación mixta (natural y artificial).

- En la noche: Iluminación artificial.

Inicialmente los objetos de color eran esferas; sin embargo debido a errores de medición presentados por el sensor de distancia HC-SR04 se optó por cilindros. En la Figura 78 se visualiza la detección de esferas y cilindros con iluminación artificial.



(a)



(b)

Figura 78. Detección de los objetos de color con iluminación artificial (a) esferas y (b) cilindros

Para cada objeto cilíndrico de color se realizó pruebas experimentales en las dos condiciones de iluminación (artificial y mixta), como se observa en la Figura 79.

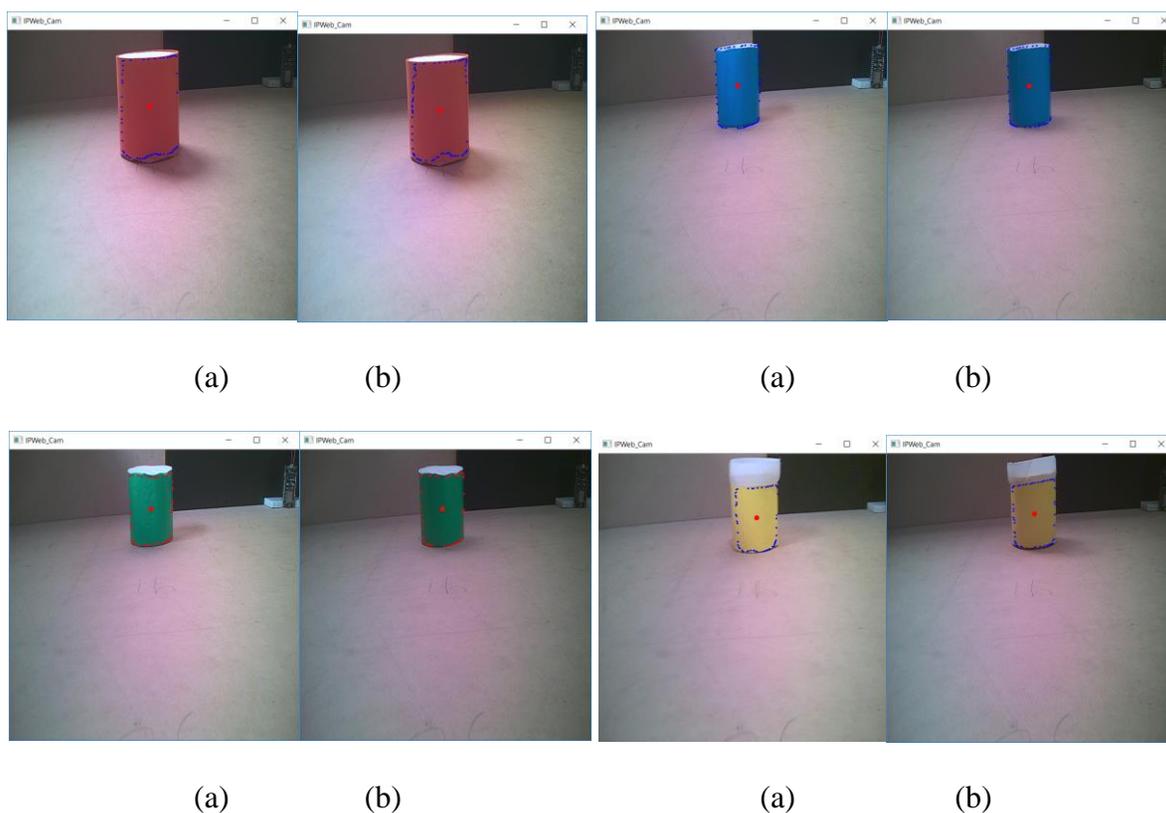


Figura 79. Detección de los objetos de color con iluminación (a) artificial y (b) mixta

A simple vista los objetos presentan mejor detección con iluminación artificial que con iluminación mixta. Experimentalmente se comprueba lo dicho anteriormente en base a datos de perímetro y área de los objetos, como se observa en la Figura 80 y Figura 81.

Como métricas de evaluación del correcto funcionamiento del algoritmo de detección de colores, se han tomado el radio y área correspondientes a los objetos. Como se puede ver en las Figuras a continuación se realizó experimentos con 200 datos de medición continua para determinar el error de detección.

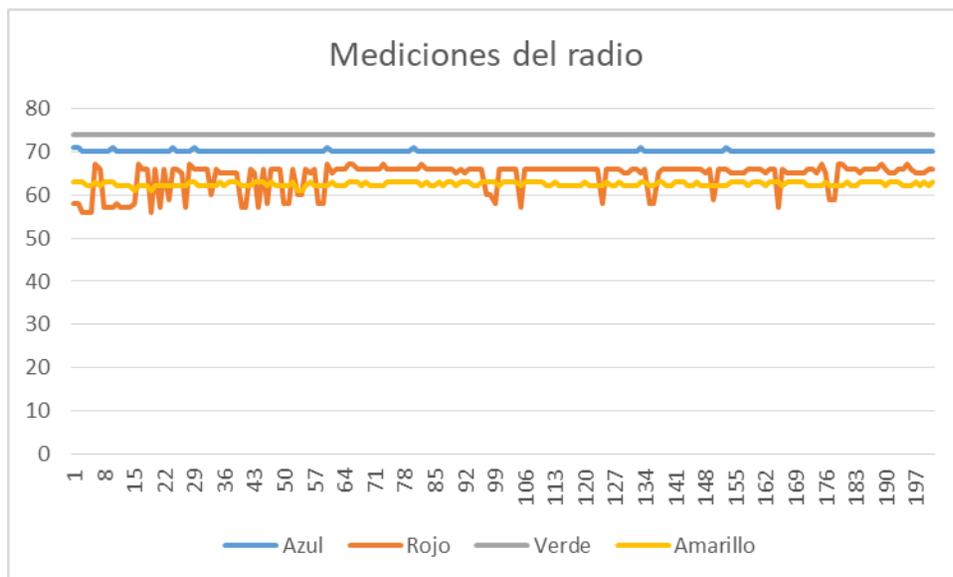


Figura 80. Mediciones del radio del cilindro para 200 datos

Como se puede apreciar en la *Figura 80* se tienen cuatro líneas de datos que son las mediciones del radio para el cilindro Azul, Rojo, Verde y Amarillo mediante visión artificial. Las mediciones son bastante estables en los colores verde azul y amarillo, pero en el color rojo se tienen ciertos saltos de error; eso es debido a pequeños cambios de iluminación y fallas en la detección que pueden ser solventados al comparar con el área detectada

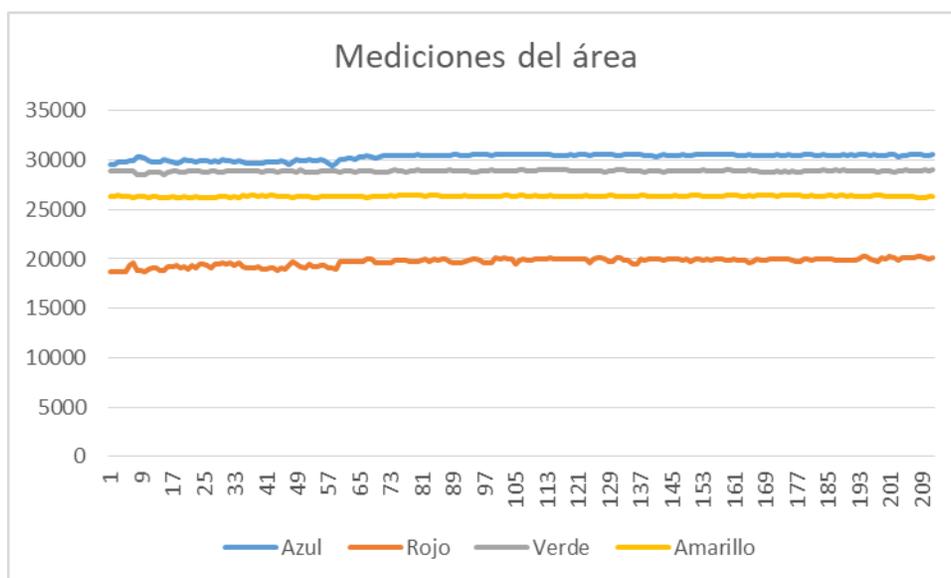


Figura 81. Mediciones del Área del cilindro para 200 datos

Al igual que en las mediciones del radio, en la *Figura 81* se tomaron 200 datos continuos discriminando el área de los cilindros Azul, Rojo, Verde y Amarillo mediante visión artificial, y como se puede apreciar el área presenta mayor estabilidad que el radio en todos los casos.

En las *Figura 82* y *Figura 83* se puede apreciar el error de medición de los datos discriminados por visión artificial en radio y área de las *Figura 80* y *Figura 81* con respecto al valor real medido de manera experimental.

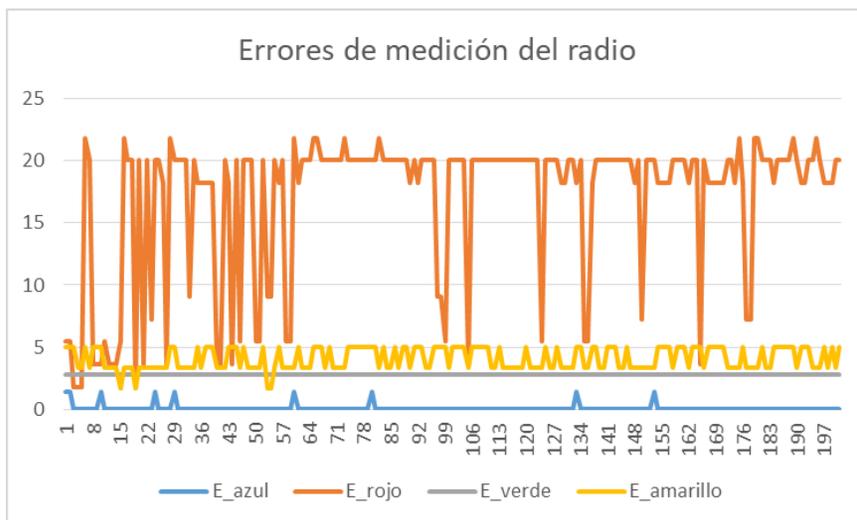


Figura 82. Error porcentual de medición del radio en el algoritmo de detección de color

Como se puede apreciar en la *Figura 82* el radio de los cilindros azul, verde y amarillo no varía en gran medida con errores de hasta 5%, al contrario que las mediciones del cilindro rojo que varían con errores de hasta 22%.

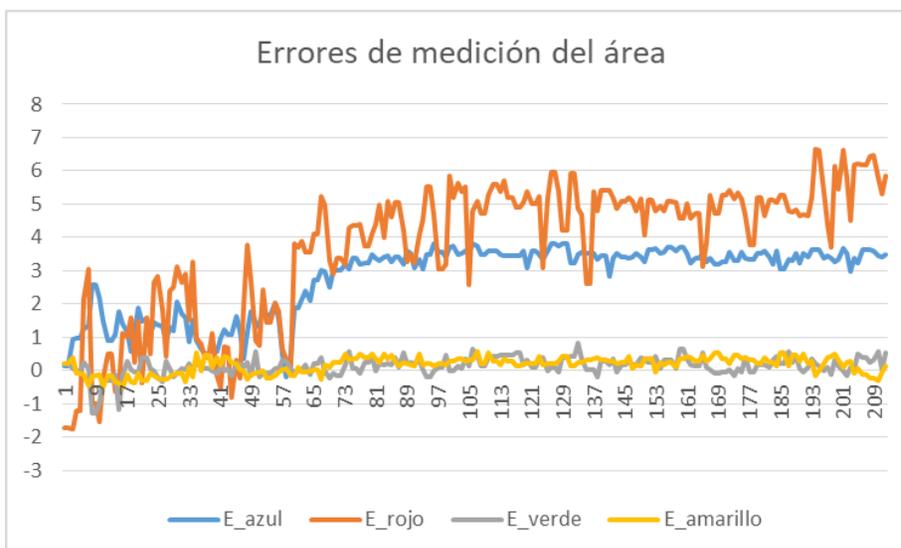


Figura 83. Error porcentual de medición del área del cilindro en el algoritmo de detección de color

En la *Figura 83* se puede apreciar que los errores de medición del área mediante visión artificial es muy aceptable con errores que no sobrepasan el 6% en el peor de los casos que es la medición del cilindro de color rojo.

5.3 Pruebas realizadas al algoritmo de localización de robots

Se realizaron múltiples pruebas para determinar cuál de los métodos mencionados en la sección 3.3.2 (Triangulación, Fingerprints) ofrecen mejores resultados al localizar robots dentro de un escenario estructurado. Las condiciones de los robots para la realización de las pruebas de medición fueron: estado estacionario y orientación fija.

5.3.1 Método de triangulación

El valor RSSI fue obtenido inicialmente de dos AP's (robot guía y router), se realizaron pruebas de medición a una distancia mínima de 15 *cm* y máxima de 1 *m*. En cada prueba se obtuvieron 300 datos, en la Tabla 2 se observa el porcentaje de error de la distancia medida y calculada mediante el valor RSSI. La distancia se calculó usando la ecuación (15) de Friss, analizada en la sección 2.5.

En la *Figura 84* se aprecia la posición y orientación de los robots para la toma de datos a las distancias de 15*cm*, 50*cm*, 75*cm*, y 100*cm* con respecto a la referencia del AP ubicado en la pared azul; la flecha indica la orientación de los robots en el escenario. Cabe recalcar que a través de la tarjeta de desarrollo nodeMCU un robot obrero es capaz de obtener el valor RSSI de los cuatro AP del escenario, con un tiempo de actualización de **3 segundos** entre las medidas.

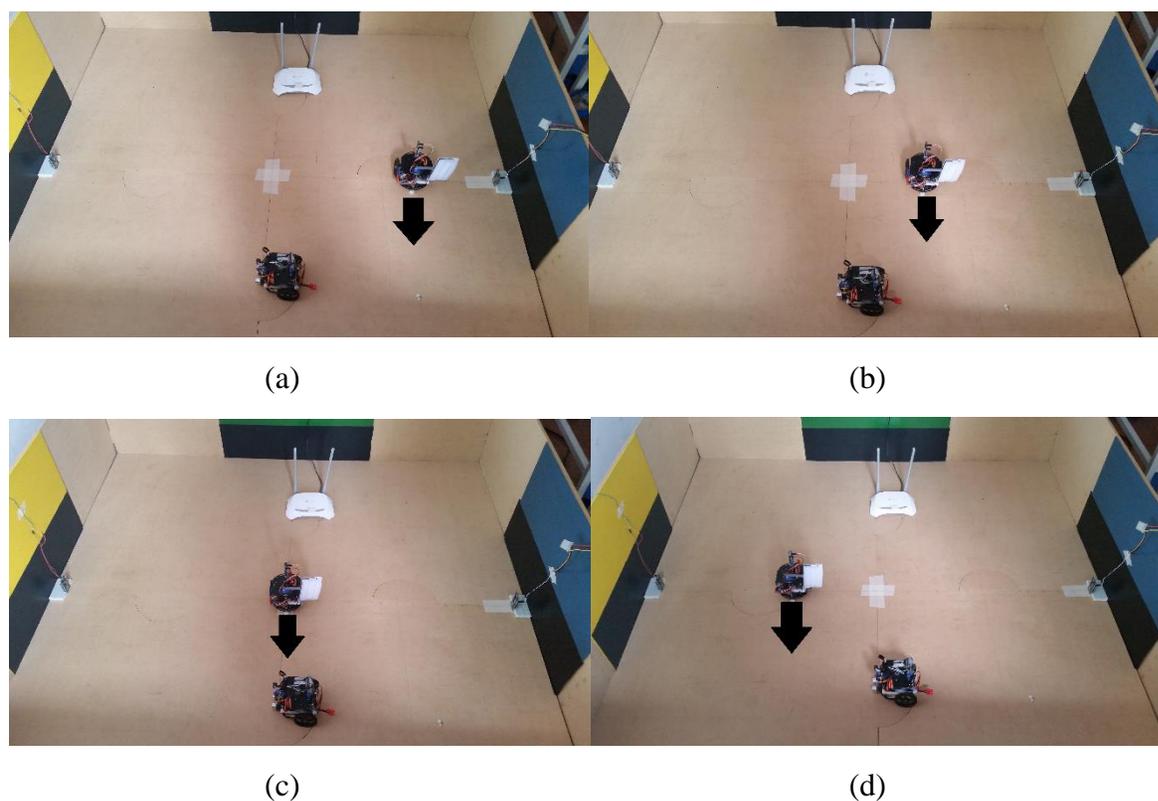


Figura 84. Posición y orientación de los robots en pruebas de RSSI (a) 15cm, (b) 50cm, (c) 75cm y (d) 100cm.

En la Tabla 8 se observa que el valor RSSI disminuye a medida que se incrementa la distancia en la mayoría de los casos, además se observa que el porcentaje de error de la distancia calculada en base a la distancia medida es grande siendo el 71% el error más alto presentado en las mediciones con respecto al AP robot guía. A pesar de existir porcentajes de errores bajos como 1.5% a la distancia de 60cm respecto al AP robot guía; utilizar dos AP's es descartado por la variabilidad de los datos presentados en la Tabla 8. Por lo que se incrementa el número de AP's con el objetivo de reducir el porcentaje de error.

Tabla 8*Análisis de la variabilidad de la señal RSSI con dos AP's.*

Router				Robot Guía			
Distancia medida [cm]	Valor RSSI [dB]	Distancia calculada [cm]	Error %	Distancia medida [cm]	Valor RSSI [dB]	Distancia calculada [cm]	Error %
10	-9	15.54	-25.30	10	-17	12.56	-25.64
20	-11	19.56	2.19	20	-22	20.49	-2.46
30	-12	21.95	26.84	30	-26	35.25	-17.50
40	-18	43.79	-9.48	40	-28	42.70	-6.74
50	-14	27.63	44.74	50	-33	42.70	14.61
60	-18	43.79	27.01	60	-35	60.90	-1.50
70	-19	49.14	29.81	70	-41	67.81	3.13
80	-22	69.41	13.24	80	-22	50.09	37.39
90	-22	69.41	22.88	90	-26	86.29	4.13
100	-20	55.13	44.87	100	-28	71.15	28.85

En la Tabla 9 se observa el porcentaje de error de la distancia experimental al incrementar el número de AP's para usar el método de triangulación. Se realizaron pruebas de medición por cada AP, en cada una se recolectó 300 muestras de cada distancia medida. Las tablas: Tabla 9, Tabla 10, Tabla 11, y Tabla 12 contiene los datos procesados con filtro de moda, además se calcula la distancia mediante una fórmula experimental basada en los datos obtenidos, debido a que la fórmula de Friss proporciona errores altos.

Tabla 9

Análisis de la señal RSSI implementando cuatro AP's respecto al router.

Router				Error Friss	Error Experimental
Distancia medida [cm]	Valor RSSI [dB]	Distancia calculada ecuación Friss[cm]	Distancia calculada ecuación experimental	%	%
10	-6	17.39	10.00	-73.93	-0.03
20	-7	19.51	20.01	2.43	-0.06
30	-11	30.93	30.12	-3.10	-0.40
40	-13	38.94	40.25	2.66	-0.64
50	-16	55.00	55.62	0.00	-1.13

En la Figura 85 se visualizan gráficamente los datos de la Tabla 9, además la ecuación obtenida de forma experimental a través de los datos adquiridos del valor RSSI con respecto al AP router, siguiendo una línea de tendencia polinómica.

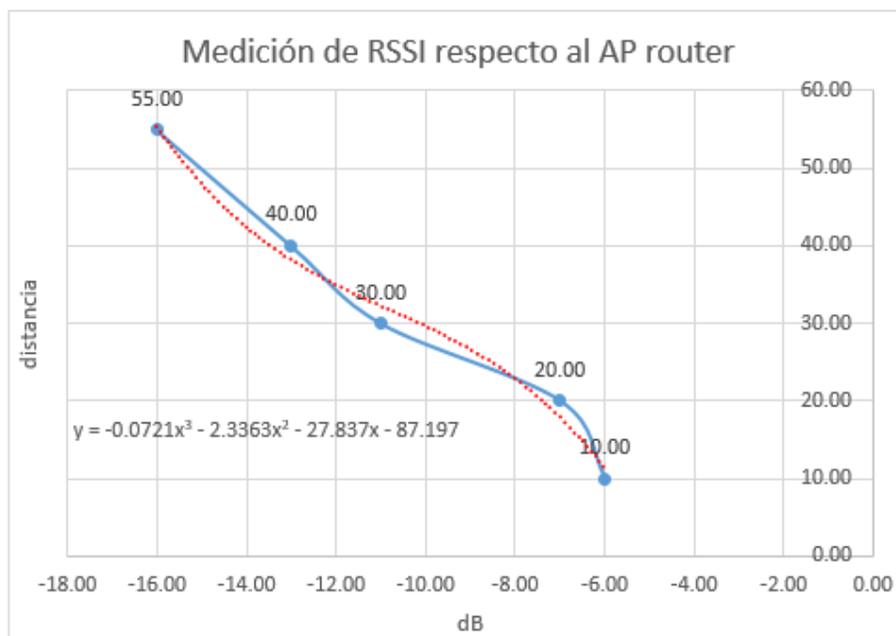


Figura 85. Medición del valor RSSI respecto al AP router

Tabla 10

Análisis de la señal RSSI implementando cuatro AP's respecto al robot guía.

Robot guía				Error Friss	Error Experimental
Distancia medida [cm]	Valor RSSI [dB]	Distancia calculada ecuación Friss[cm]	Distancia calculada ecuación experimental	%	%
10	-10	3.66	9.7781	-64.72	0.26
15	-15	6.50	13.5961	8.50	-0.14
20	-20	11.56	18.3641	8.44	1.10
25	-24	18.32	24.3745	6.68	-5.42
30	-25	20.55	26.3321	9.45	4.60
35	-28	29.03	33.5849	5.97	-6.20
45	-30	36.55	39.7501	8.45	1.32

En la Figura 86 se representan los datos de la Tabla 10, y la ecuación obtenida de forma experimental de los datos obtenidos del parámetro RSSI siguiendo una línea de tendencia polinómica.

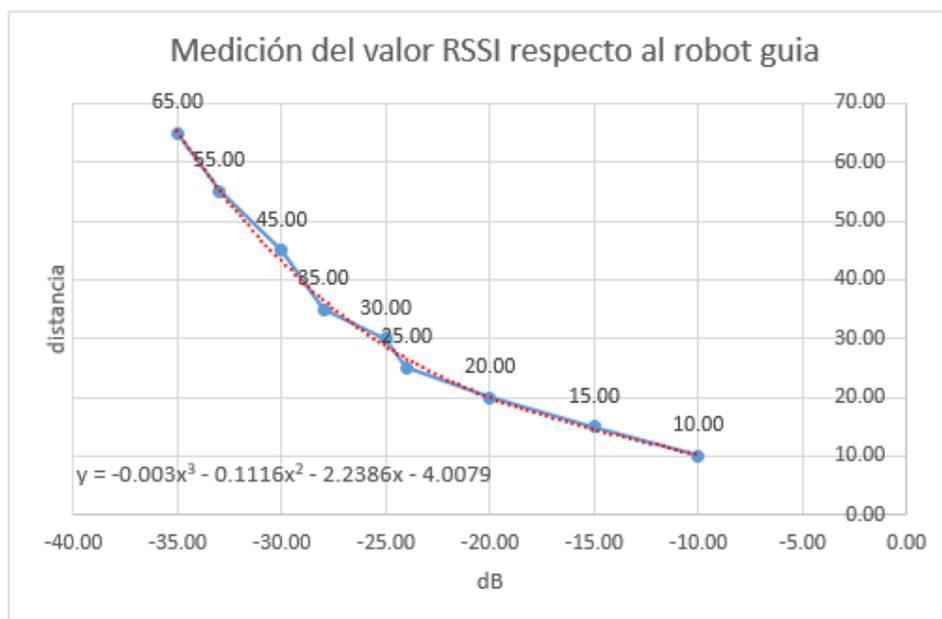


Figura 86. Medición del valor RSSI respecto al AP robot guía

Tabla 11

Análisis de la señal RSSI implementando cuatro AP's respecto a Ref1.

Referencia pared amarilla				Error Friss	Error Experimental
Distancia medida [cm]	Valor RSSI [dB]	Distancia calculada ecuación Friss[cm]	Distancia calculada ecuación experimental	%	%
10	-21	10.67	10.44	-6.70	-4.444
20	-25	16.91	20.61	15.45	-3.05
30	-32	37.86	31.42	-26.19	-4.745
40	-34	47.66	42.48	-19.15	-6.191
60	-36	60.00	60.68	0.00	-1.127

En la Figura 87 se visualizan gráficamente los datos de la Tabla 11 y adicional la ecuación obtenida experimentalmente para los valores RSSI medidos, siguiendo una línea de tendencia polinómica.

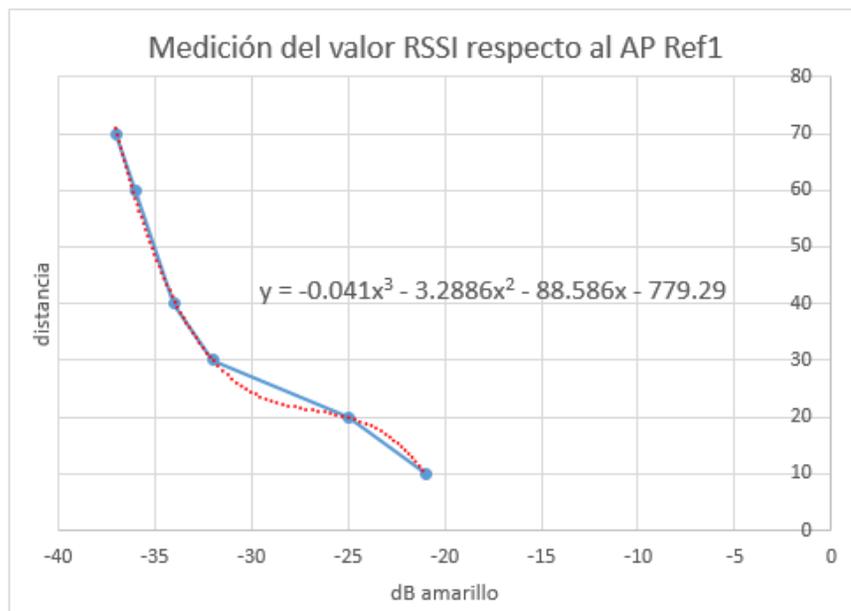


Figura 87. Medición del valor RSSI respecto al AP referencia amarillo

Tabla 12

Análisis de la señal RSSI implementando cuatro AP's respecto al Rf2.

Referencia pared azul				Error Friss	Error Experimental
Distancia medida [cm]	Valor RSSI [dB]	Distancia calculada ecuación Friss[cm]	Distancia calculada ecuación experimental	%	%
10	-22	7.55	9.97	2.45	0.0296
20	-27	13.43	19.77	6.57	0.2346
30	-31	21.29	31.46	8.71	-1.4642
40	-33	26.80	37.93	13.20	2.072
50	-37	42.48	50.86	7.52	-0.8604

En la Figura 88 se representan gráficamente los datos de la Tabla 12, además se expresa la ecuación obtenida mediante datos de mediciones del parámetro RSSI siguiendo una línea de tendencia polinómica.

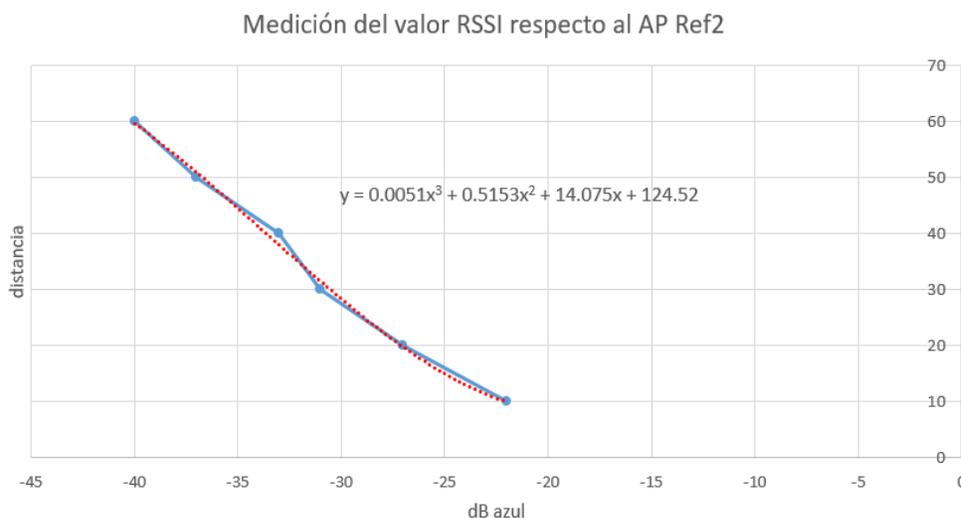


Figura 88. Medición del valor RSSI respecto al AP referencia azul

Una vez obtenidas las ecuaciones experimentales respecto a cada AP, se procede a realizar pruebas de medición en el escenario estructurado. Para estas pruebas los robots se encuentran en estado estacionario, y la orientación difiere de la utilizada para las pruebas anteriores como se observa en la Figura 89. Las pruebas de medición del valor RSSI se realizaron a diferentes distancias de los AP', las mismas se detallan en las tablas siguientes. En la Figura 89 (a) el robot obrero se encuentra a 80cm del AP ref2, en la Figura 89 (b) el robot está a 60cm del AP ref1, en la Figura 89 (c) se posiciona el robot a 60cm del AP robot guía y en la Figura 89 (d) el robot se localiza a 40cm del AP router.

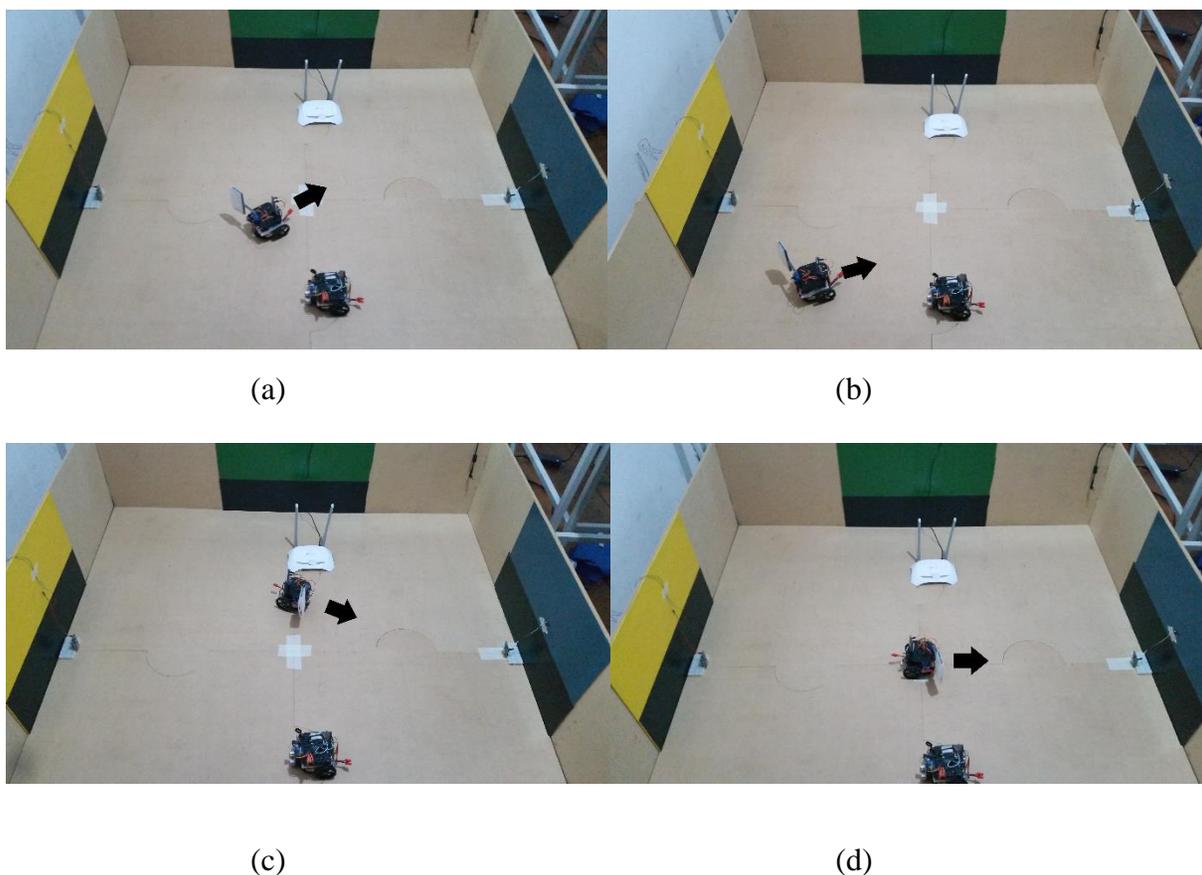


Figura 89. Medición del valor RSSI respecto a todos los AP's variando la posición y orientación del robot obrero.

Los resultados obtenidos de las pruebas de medición realizadas con todos los AP's se observan en las tablas: Tabla 13, Tabla 14, Tabla 15 y Tabla 16.

Tabla 13

Pruebas de medición de la señal RSSI respecto al AP router.

Router				Error Friss	Error Experimental
Distancia medida [cm]	Valor RSSI [dB]	Distancia calculada ecuación Friss[cm]	Distancia calculada ecuación experimental	%	%
20	-11	30.93	30.12	-54.64	-50.61
30	-13	38.94	40.25	-29.79	-34.18
40	-12	34.70	34.35	13.24	14.12
50	-15	49.02	52.89	1.96	-5.78
60	-17	61.71	52.01	-2.85	13.32

En la **Tabla 13** se analiza la variación del parámetro RSSI con respecto al AP router, donde el porcentaje de error más alto proviene de la fórmula Friss, a pesar que la fórmula experimental presenta un porcentaje de error menor en comparación Friss, el error sigue siendo muy alto para este proyecto.

Tabla 14

Pruebas de medición de la señal RSSI respecto al AP robot guía.

Robot Guía				Error Friss	Error Experimental
Distancia medida [cm]	Valor RSSI [dB]	Distancia calculada ecuación Friss[cm]	Distancia calculada ecuación experimental	%	%
10	-10	30.93	30.12	6.34	0.26
20	-15	38.94	40.25	13.50	24.89
30	-20	34.70	34.35	18.44	34.07
40	-22	49.02	52.89	25.45	43.45
50	-24	61.71	52.01	31.68	47.29

En la **Tabla 14** se indica que la fórmula de Friss para el cálculo de la distancia en función del parámetro RSSI (dB) presenta un porcentaje de error alto con respecto a las medidas reales. Por otro lado el porcentaje de error de la fórmula experimental es alto en referencia al tamaño del escenario. Se asume que estas variaciones dependen de la orientación del robot.

Tabla 15

Pruebas de medición de la señal RSSI respecto al AP Refl.

Referencia pared amarilla				Error Friss	Error Experimental
Distancia medida [cm]	Valor RSSI [dB]	Distancia calculada ecuación Friss[cm]	Distancia calculada ecuación experimental	%	%
10	-23	13.43	17.37	-3.43	-7.3656
20	-25	16.91	20.61	3.09	-0.61
30	-32	37.86	31.42	-7.86	-1.4236
40	-34	47.66	42.48	-7.66	-2.4764
50	-37	67.32	73.07	-17.32	-23.0716

Analizando la **Tabla 15**, el porcentaje de error más alto procede de la fórmula experimental, debido al cambio de orientación del robot.

Tabla 16

Pruebas de medición de la señal RSSI respecto al AP Ref2.

Referencia pared azul				Error Friss	Error Experimental
Distancia medida [cm]	Valor RSSI [dB]	Distancia calculada ecuación Friss[cm]	Distancia calculada ecuación experimental	%	%
10	-15	3.37	12.13	66.26	-21.25
30	-18	4.77	8.38	84.11	72.05
40	-26	11.97	17.28	70.07	56.81
50	-28	15.07	22.46	69.86	55.08
60	-33	26.80	37.93	55.33	36.79

En la **Tabla 16** se observa que el mayor porcentaje de error proviene de la fórmula Friss, sin embargo al cambiar la orientación del robot el porcentaje de error utilizando fórmula experimental se incrementó siendo inaceptable para este proyecto.

En la Figura 90 se visualizan gráficamente el error de medición debido a la variación de la orientación del robot con la que se adquirió los datos del valor RSSI. Al analizar la Figura 90 y las tablas (Tabla 13, Tabla 14, Tabla 15 y Tabla 16) se puede concluir, que el método de triangulación no es adecuado para localizar robots u objetos en el escenario estructurado, debido a la variación significativa del valor RSSI, lo que causa que las distancias necesarias para implementar este método presenten errores máximos del 72%.

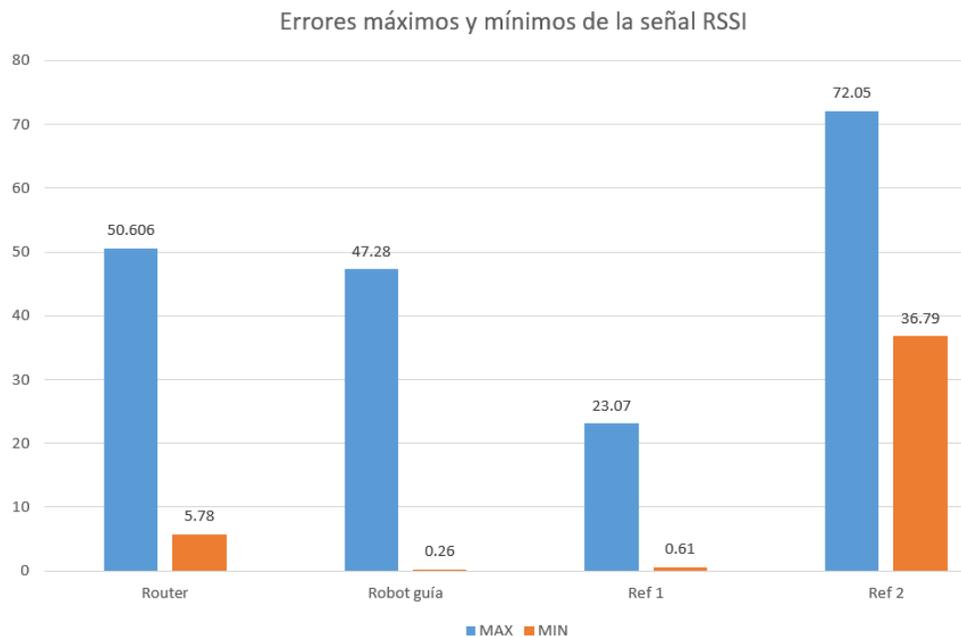


Figura 90. Análisis de errores de las pruebas de medición de la señal RSSI

5.3.2 Método de FingerPrints

Para el método de Fingerprints se dividió el escenario de pruebas en 49 celdas de $20\text{cm} \times 20\text{cm}$. En cada celda se obtuvo el valor RSSI con respecto a los cuatro AP's generando una base de datos con la posición del punto de referencia y el valor RSSI medido.

Con la base de datos creada en la **sección 4.6.2** se procede a realizar pruebas de medición en puntos conocidos y desconocidos del escenario, infiriendo la localización de los robots a través de la mejor coincidencia entre los valores medidos y almacenados.

Con el objetivo de determinar la exactitud del método se realizaron conjuntos de pruebas de 300 datos de muestreo cada una, desplazando al robot de celda en celda. La orientación del robot difiere de la utilizada en la creación de la base de datos, y se trabaja con el robot en estado estacionario.

Obteniendo el porcentaje de acierto de la medición en de cada celda, los resultados reflejan la variación del dB en cada punto con el objetivo quieto.

En la Figura 91 se evidencia el porcentaje resultante para cada celda del escenario de la prueba realizada, donde 100% determina que el total de mediciones dieron como resultado el punto donde se situó el robot y 0 % que el total de mediciones dieron como resultado puntos diferentes al que se situó el robot. Tomando en cuenta mediciones por otros métodos se concluye que el método de Fingerprints presenta menor variación con respecto a la posición del robot dentro del escenario.

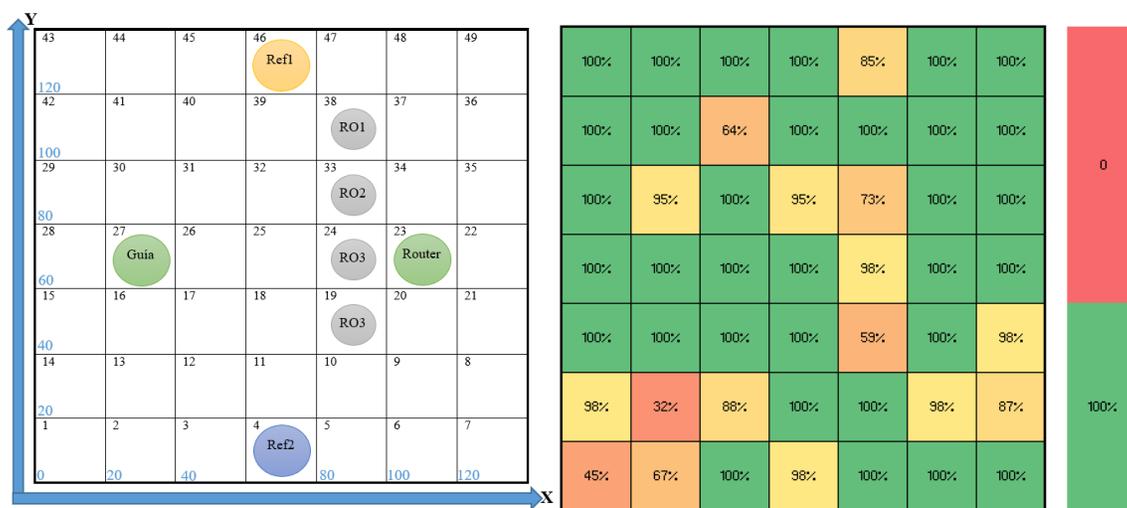


Figura 91. Porcentaje de aciertos en la medición de un conjunto de 300 datos para cada celda.

De las pruebas realizadas se puede evidenciar que en algunas celdas como por ejemplo la número 13, 19 y 40, entre otros. Las medidas resultaron fallidas perdiendo un total del porcentaje de acierto en la medición. En la Figura 92 se puede apreciar el comportamiento en tres puntos con errores paulatinos de medición, donde la dispersión de medida falla mayoritariamente en el punto 13. El porcentaje de acierto respecto a la celda donde se ubica el robot se reparte entre el punto real y puntos alrededor del mismo.

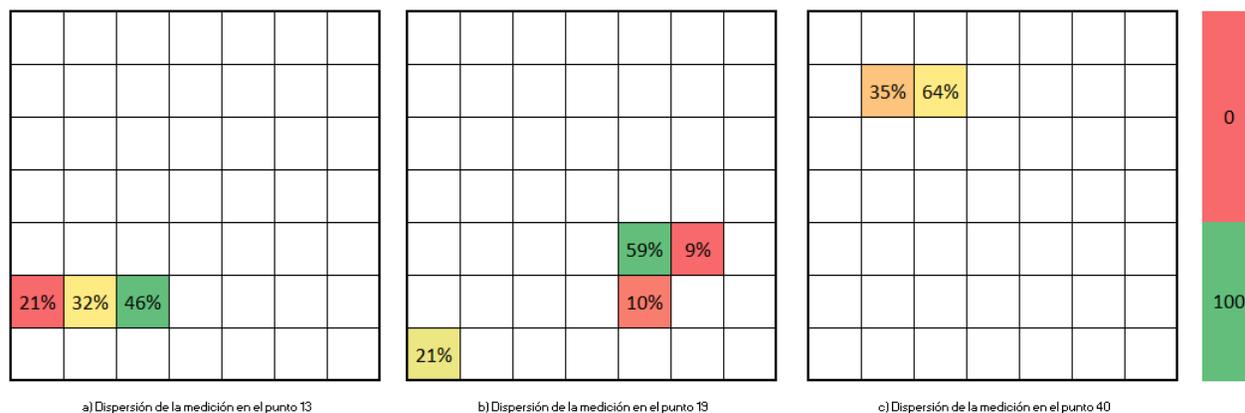


Figura 92. Dispersión de localización calculada en los puntos 13, 19 y 40.

Como se puede ver en la Figura 92, la celda de colocación del robot para la prueba de medición fue la 13 con un porcentaje de acierto en dicho punto de 32%, el resto de medidas se repartieron entre el punto 12 y 14 presentando un 46% y 21 % del total respectivamente. Los cuales en realidad se encontraban vacíos sin presencia de ningún robot.

De igual manera para el punto 19 y 40 se puede ver que solo el 59% y 64 % respectivamente son los resultados de la medición en un conjunto de 300 datos, el resto del porcentaje se ve dividido entre los casilleros vecinos.

Todas las mediciones de las Figura 91 y Figura 92 se realizaron a de manera probabilística, con un filtro modal aplicado a los datos adquiridos.

5.4 Pruebas realizadas a los algoritmos de planificación y seguimiento de trayectoria

Para evaluar los algoritmos de planificación y seguimiento de trayectoria se realizaron cinco pruebas de medición, variando la posición de los robots y la posición de los objetos de color en el

escenario bajo ciertas limitantes. Se consideraron varios factores que limitan la variación de posición tanto de los objetos de color como de los robots, que se describen a continuación:

- Los objetos de color deben estar a una distancia mínima de 15 *cm* y máxima de 80 *cm* del robot guía.
- Debido al ángulo de apertura de medición del sensor de distancia HC-SR04, los objetos de color deben tener una separación mínima de 30 *cm* y no deben ser colineales.
- Los robots obreros deben calibrarse mirando a la pared azul, y luego deben ser girados 90 ° hacia la derecha en sentido horario.
- Los robots obreros deben posicionarse adelante del router, sin invadir celdas.
- El robot guía debe posicionarse en la celda 27, y se calibra mirando a la pared amarilla.
- El router debe posicionarse en la celda 23, la referencia de la pared amarilla en la celda 46 y la referencia de la pared amarilla en la celda 4.

En la Figura 93 se observa una de las posibles combinaciones de posicionamiento de los objetos de color y robots en el escenario estructurado.

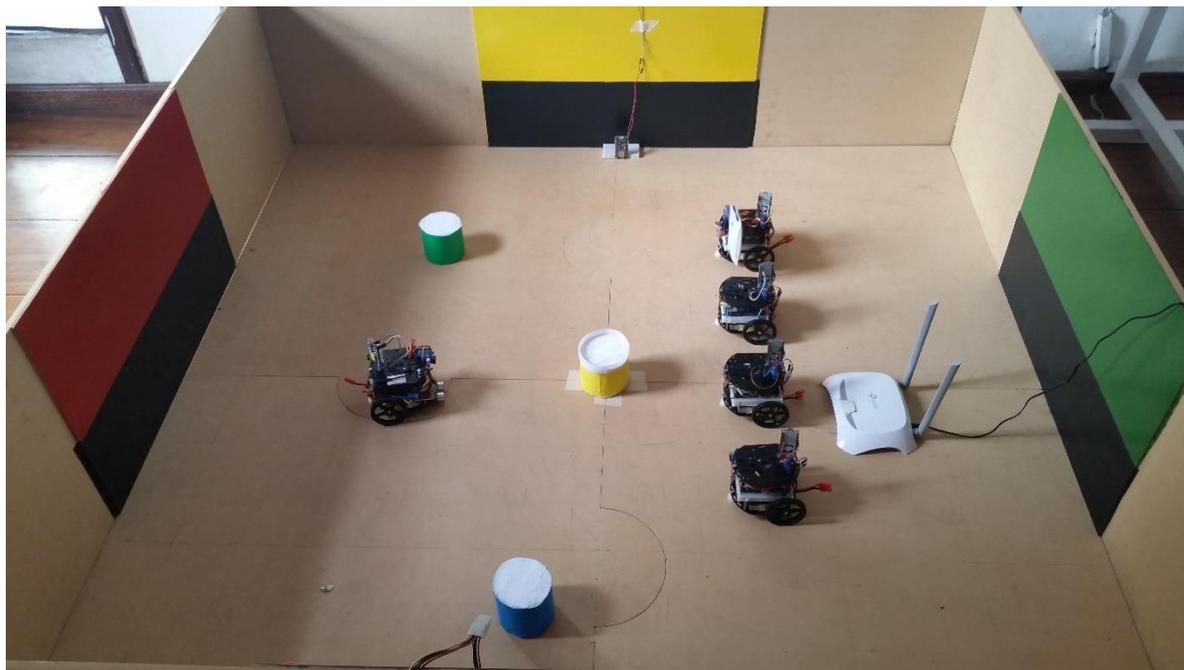


Figura 93. Localización de objetos de color y robots en el escenario

Con la posición de los objetos de la Figura 93, se genera la primera trayectoria que inicia en posición (8,10) y finaliza en la posición (6.5, 6.5). La selección de robot obrero se realiza en base al cálculo de la mínima distancia entre el objeto de color y el robot. En la Figura 94 se visualiza la generación de trayectoria mediante el algoritmo RRT.

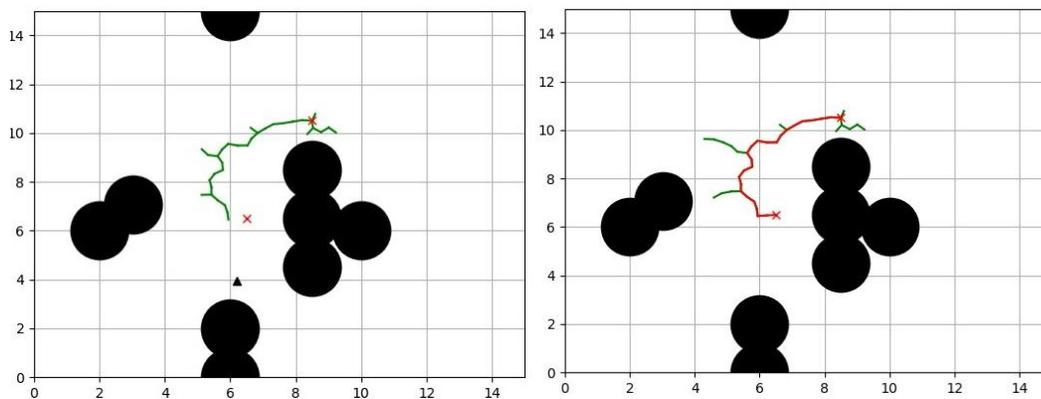


Figura 94. Generación aleatoria de trayectoria para recoger el objeto de color

Para el seguimiento de trayectoria se cuenta con realimentación en tiempo real del ángulo del desfase del robot al desplazarse obtenido del sensor de medición inercial MPU6050, el valor RSSI solamente se actualiza antes de iniciar el seguimiento de trayectoria y al finalizar el seguimiento de trayectoria. Se definió este comportamiento debido a los errores producidos por el cambio de orientación constante del robot y errores derivados del tiempo de actualización del valor RSSI obtenido de la tarjeta de desarrollo nodeMCU. Una vez finalizada la primera etapa de planificación y seguimiento de trayectoria, se envía una señal desde la PC de proceso para mover el brazo robótico y recoger el objeto de color. En la Figura 95 se observa el seguimiento de trayectoria en el PC de proceso, y en la Figura 96 se visualiza mediante una secuencia de imágenes el seguimiento de la misma realizado por el robot obrero.

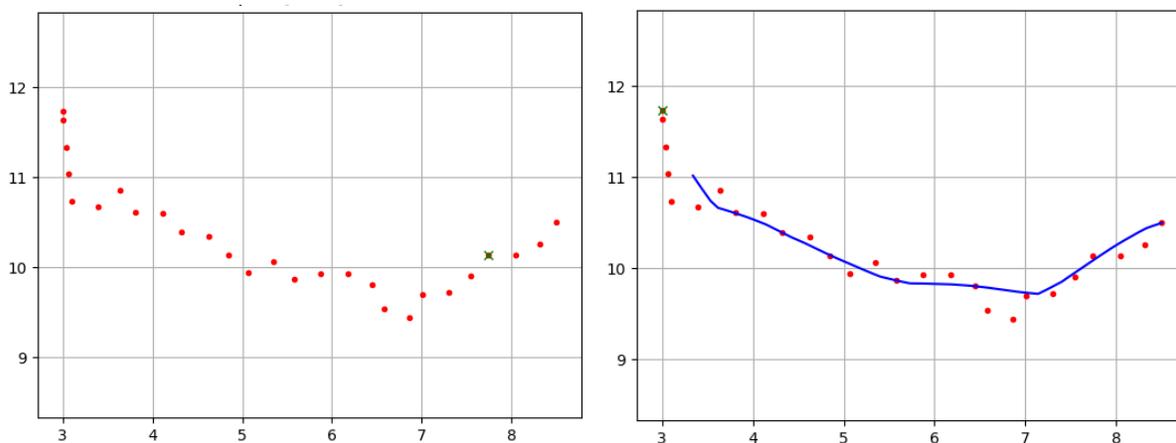


Figura 95. Seguimiento de la trayectoria para recoger el objeto de color

5.5 Pruebas realizadas a los algoritmos de planificación y seguimiento de trayectoria

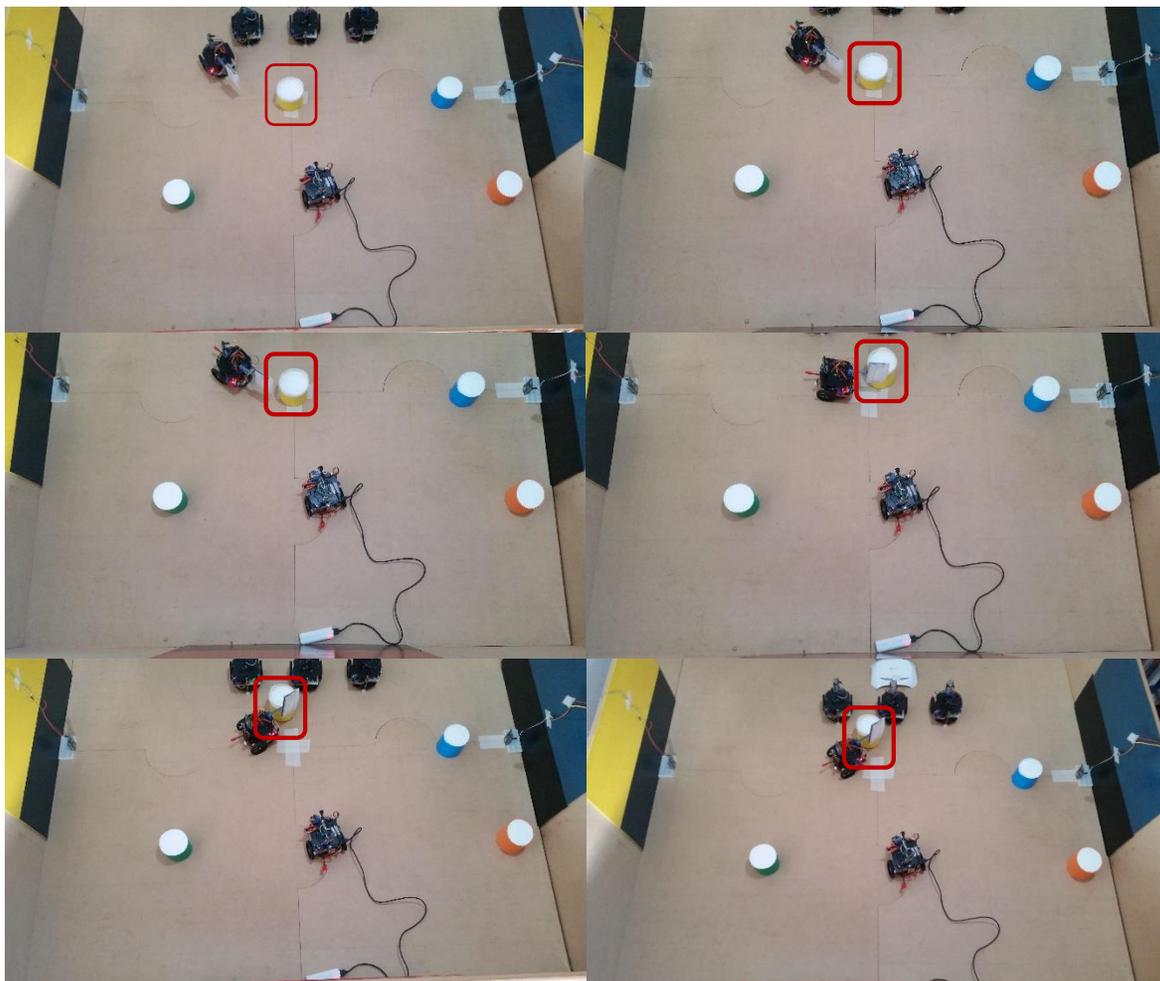


Figura 96. Secuencia de imágenes del seguimiento de la trayectoria del robot para recoger el objeto de color

De manera similar a la primera etapa de generación de trayectoria, se realiza la segunda etapa, la misma que consiste en transportar al objeto de color a su pared correspondiente, ver Figura 97, Figura 98 y Figura 99.

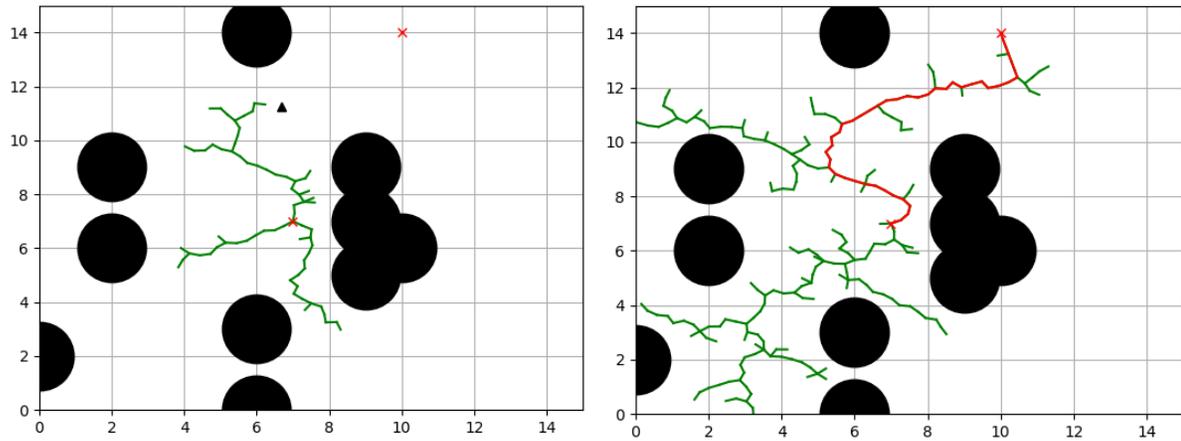


Figura 97. Generación aleatoria de trayectoria para transportar el objeto de color a su correspondiente pared.

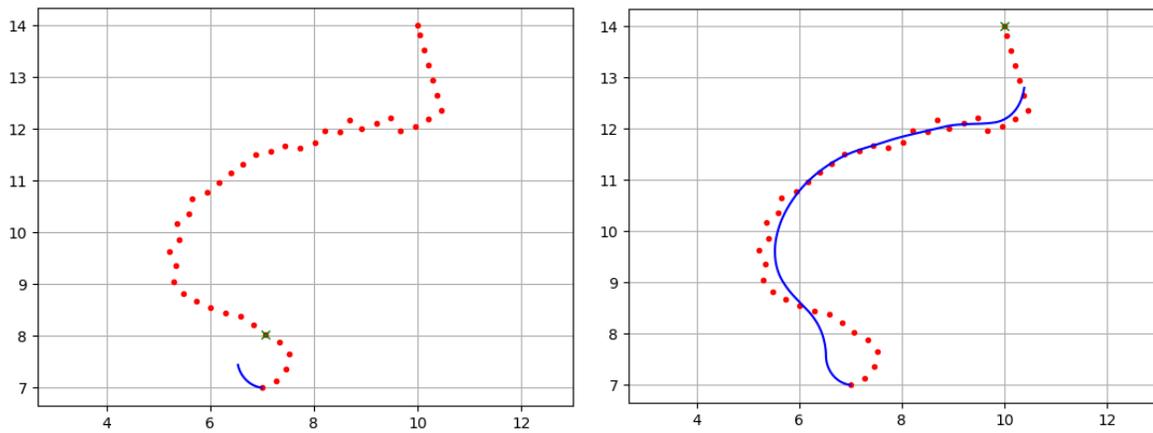


Figura 98. Seguimiento de la trayectoria para transportar el objeto de color a su correspondiente pared.

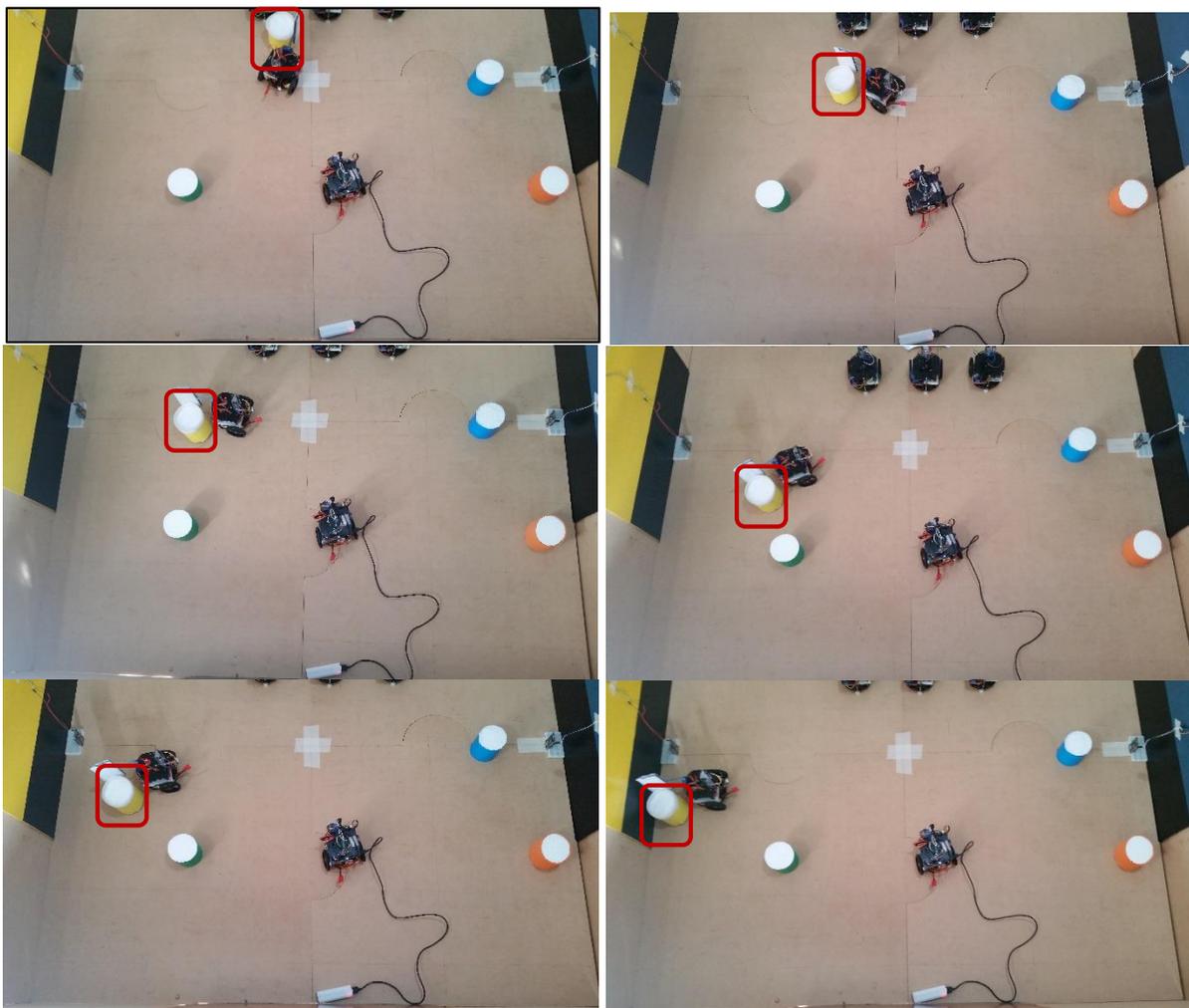


Figura 99. Secuencia de imágenes del seguimiento de la trayectoria para transportar el objeto de color a su correspondiente pared.

5.6 Análisis del sistema multi-robot cooperativo implementado

A continuación se analizará el sistema multi-robot cooperativo con métricas de tiempo de ejecución parcial y total, distancia recorrida y robustez de comunicación.

5.6.1 Análisis del tiempo de ejecución del sistema multi-robot

Para analizar el tiempo de ejecución del sistema multi-robot cooperativo se realizó tres pruebas para cada objeto de color según la distribución de la **Figura 93**, en la **Tabla 17** se detalla el tiempo parcial requerido por etapa del sistema y el tiempo total para cada objeto de color.

Tabla 17*Tiempo de ejecución del sistema.*

Prueba No.	Color	Tiempo primera etapa [s]	Tiempo segunda etapa parte 1 [s]	Tiempo segunda etapa parte 2 [s]	Tiempo tercera etapa parte 1 [s]	Tiempo tercera etapa parte 2 [s]	Tiempo total del sistema [s]	Tiempo total del sistema [min]
1	Rojo	13	53	350	20	390	826	13.77
	Azul	9	45	250	27	172	503	8.38
	Verde	7	39	287	102	615	1050	17.50
	Amarillo	9	26	304	19	278	636	10.60
2	Rojo	11	40	278	13	308	650	10.83
	Azul	8	46	210	16	152	432	7.20
	Verde	10	35	375	83	659	1162	19.37
	Amarillo	13	11	378	39	225	666	11.10
3	Rojo	11	50	312	21	359	753	12.55
	Azul	6	39	292	18	187	542	9.03
	Verde	9	26	322	72	572	1001	16.68
	Amarillo	10	16	309	46	251	632	10.53

En la Figura 100 se visualiza un diagrama de barras con los datos del tiempo de ejecución por objeto de color según la distribución de elementos del sistema de la Figura 93. Se puede observar que el mayor tiempo de ejecución es del objeto de color verde y el menor tiempo es del objeto azul para esta distribución de elementos.

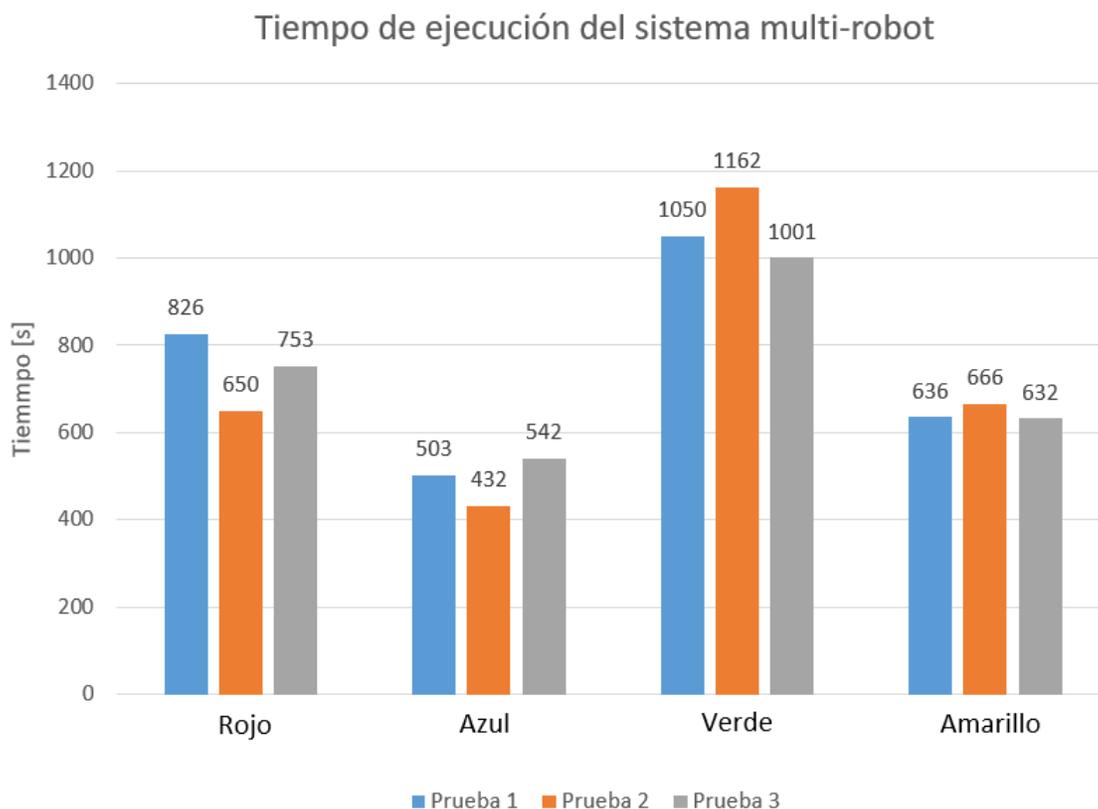


Figura 100. Análisis parcial del tiempo de ejecución del sistema.

En la Figura 101 se observa un diagrama de barras con los datos del tiempo de ejecución totales del sistema. Con los datos de las tres pruebas realizadas se puede decir que el sistema multi-robot cooperativo completa las tareas de localización y desplazamiento de los objetos de color en un tiempo máximo de ejecución de 50.25 minutos, y en un mínimo de 48.5 minutos. El tiempo de ejecución total del sistema es alto debido a diversos factores. Se debe considerar que sistema trabaja con un robot a la vez; al generar una trayectoria con el método RRT no siempre se obtiene una óptima trayectoria a seguir. También se considera que el nivel de batería disminuye en la ejecución del sistema, afectando a la velocidad de los robots.

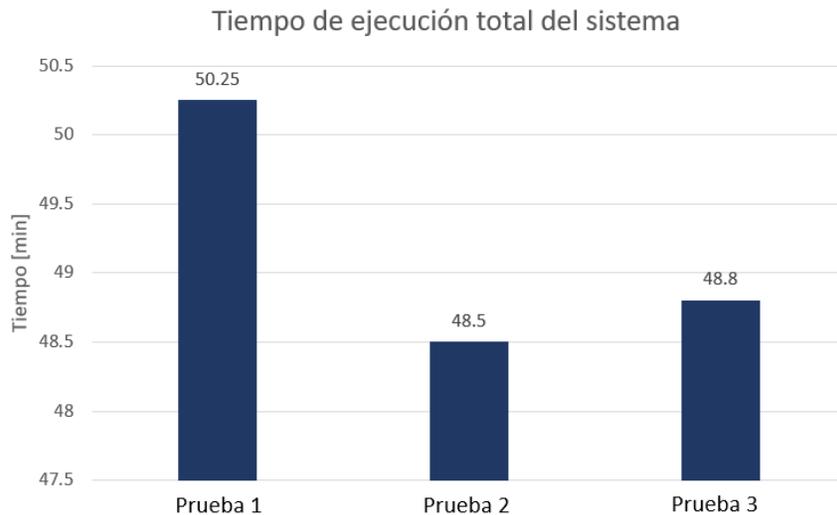


Figura 101. Análisis total del tiempo de ejecución del sistema.

5.6.2 Análisis del subsistema de comunicación

Se utiliza el comando ping para determinar la accesibilidad de los host de la red de trabajo, en la Tabla 18 se analiza la accesibilidad y latencia de cada elemento de la red inalámbrica utilizada en este proyecto de titulación.

Tabla 18

Latencia en los elementos de la red de comunicación inalámbrica.

Referencia pared azul			Paquetes			Tiempo de ida y vuelta [ms]		
No.	Identificación del dispositivo	IP	Enviados	Recibidos	Perdidos	Min	Máx	Media
1	Router	192.168.1.1	4	4	0	2	19	10
2	Robot guía	192.168.1.102	4	4	0	2	5	3
3	Cámara IP	192.168.1.105	4	4	0	3	10	7
4	Obrero 1	192.168.1.111	4	2	2	56	87	71
5	Obrero 2	192.168.1.112	4	2	2	14	37	25
6	Obrero 3	192.168.1.113	4	2	2	3	89	53
7	Obrero 4	192.168.1.114	4	2	2	51	83	67
8	PC	192.168.1.115	4	4	0	1	1	1

Como se observa en la tabla anterior se analiza la latencia de los elementos de la red, especialmente de las tarjetas nodeMCU, es decir el tiempo que tardan en transmitir un paquete dentro de la red. Las tarjetas nodeMCU correspondientes a los robots obreros presentan pérdidas de paquetes y retrasos en la red al tener medias mayores a $2ms$. Ocasionando lentitud de transmisión y recepción de datos, cierre inesperado de conexiones y transmisión de información incompleta.

Estos problemas son enfrentados el proyecto de titulación siendo asociados a hardware, es decir los problemas existentes en la comunicación inalámbrica del proyecto se deben a las tarjetas nodeMCU utilizadas para transmisión y recepción de datos inalámbricos entre los robots y la PC de proceso central.

CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

Se diseñó e implementó un sistema centralizado para la cooperación de 5 robots en un ambiente controlado de $1.5m^2$, donde se manejó la jerarquía de un robot como guía para detectar los objetos a moverse y dar las posiciones a una estación central de procesamiento, seguido por cuatro obreros que realizan la tarea de llevar dichos objetos a la pared correspondiente.

La comunicación entre los robots y la estación central de procesamiento se realizó mediante señales WiFi con la incorporación de módulos NodeMCU que permiten crear puntos de acceso independientes y conectarse a otras redes, de manera que cada robot puede trabajar conectándose a una red establecida para el sistema dando una topología estrella para el sistema multi-robot cooperativo.

Para solventar la localización de objetos se realizó un sistema de reconocimiento basado en visión por computador para 4 diferentes patrones, entre los cuales se detectan objetos amarillos, verdes, azules y rojos que son capturados visualmente mediante una cámara empotrada en el robot guía.

El reconocimiento robusto de los objetos de color en el escenario sometido a diferentes condiciones de iluminación se obtuvo aplicando filtros del espacio de color HSV en el procesamiento, disminuyendo de esta manera el efecto del factor ambiental en el sistema.

Con el análisis del comportamiento del algoritmo de reconocimiento de colores se determinó que las mediciones a lo largo del tiempo tienen mejor rendimiento con los colores azul, verde y amarillo con errores promedio de medición en radio de 1%, 3% y 5% respectivamente, mientras el error en el cilindro de color rojo podía ascender hasta un 20%, y errores de hasta 6% en área del color rojo denotando valores satisfactorios para la detección.

El reconocimiento termina siendo satisfactorio a pesar de los errores en el color rojo debido a que la detección supera el 50% del objeto, dando la posibilidad de determinar su posición centrándolo y tomando la distancia con un sensor ultrasónico.

Se estableció necesaria la utilización de una IMU y un servo motor para mantener la dirección de medición de la tarjeta nodeMCU en cada uno de los robots guía y obreros debido a que las mediciones de RSSI se deben realizar de manera dirigida, así el movimiento de los robots no afecta tan bruscamente a la intensidad de señal recibida.

El método de triangulación utilizado con dos AP's y antenas direccionadas presentó errores de hasta un 72.05% en la medición de distancia, esto debido a errores de medición de la intensidad de señal, por tal razón se optó por la utilización de cuatro AP's y el método de localización "FingerPrint".

El uso de una fórmula experimental y un filtro modal en la aproximación a las medidas de RSSI permitió reducir el error generado en los AP's para un máximo error de 14% en casos puntuales, debido a que las mediciones con la aplicación de la fórmula de Friss mostraban valores con un error cercano al 50% en estado estacionario.

A pesar de los resultados corregidos por filtrado, en las pruebas de medición con leves cambios de orientación del robot, tanto por Friss como por método experimental se obtuvieron errores altamente desfavorables, de hasta un 50 y 70 % dando lugar a que las limitaciones de hardware del sistema no pueden solventar la aplicación de detección y localización en movimiento del robot de manera continua por este método al haber superado la variabilidad prevista de 10%.

Debido a las limitaciones de hardware la realimentación de la posición se realizó de manera muy lenta teniendo cerca de tres segundos sin conocimiento de su posición al robot, por tanto se realizó realimentación al principio y al final del seguimiento para verificar la posición y corregir errores de posición.

Los algoritmos de robótica cooperativa basados en una estación central ayudan al sistema a mejorar su rendimiento respecto al uso de robots específicos ya que son más adaptables al entorno y realizan tareas con menor esfuerzo global.

6.2 Recomendaciones

La detección de objetos fusionando sensores de visión y de ultrasonido es una solución altamente eficiente si se toma en cuenta el comportamiento de dichos sensores y se determina los conjuntos con los que pueden trabajar satisfactoriamente, en este caso con cilindros para la recuperación de la señal ultrasónica y el uso de filtros morfológicos y de espacio de color para la detección visual.

La utilización del método RSSI para localización y triangulación se recomienda para aplicaciones de comportamiento estacionario en la medición con grandes espacios de medida debido a que tiene errores pronunciados dependientes de interferencias externas y limitaciones de hardware.

En caso de utilización de métodos RSSI para localización se recomienda usar antenas omnidireccionales ya que los errores de medición aumentan cuando se tienen diferentes ángulos de recepción.

Se recomienda el uso de algoritmos y sistemas de robótica cooperativa en aplicaciones de diferente índole en el campo de la clasificación, localización, entre otras; ya que permiten al sistema realizar distintas tareas al mismo tiempo ya que cuenta con varios agentes robóticos, optimizando procesos y mejorando tiempos de producción.

REFERENCIAS BIBLIOGRÁFICAS

- Acevedo, J. J., Arrue, B. C., Maza, I., & Ollero, A. (2011). Distribución óptima de múltiples robots en vigilancia de perímetros. *IEEE/RSJ International Conference*, 24(6), 1394 –1404. Recuperado a partir de http://personal.us.es/imaza/papers/conferences/acevedo_robot11_web.pdf
- Ada, Lady. (2018). *Adafruit VL53L0X Time of Flight Micro-LIDAR Distance Sensor Breakout*. Recuperado a partir de <https://learn.adafruit.com/adafruit-vl53l0x-micro-lidar-distance-sensor-breakout>
- Álvarez Romero, L. M., & Figueroa Montenegro, J. E. (2010). Implementación de algoritmos de navegación utilizando la plataforma del irobot create y módulos de comunicación inalámbrica xbee. Recuperado a partir de <http://repositorio.espe.edu.ec/handle/21000/2643>
- Apolo Peñaloza, A. K., & Velasco Barrera, A. A. (2013). *Desarrollo de un sistema de robotica cooperativa entre dos elementos robots tipo Robonova*. Universidad de las Fuerzas Armadas - ESPE.
- Aranda, J., Moreno, D., Chaos, D., Díaz, J. M., Muñoz, R., & De La Cruz, J. M. (s. f.). *Control coordinado de formaciones de vehículos autónomos no tripulados*. Recuperado a partir de <https://intranet.ceautomatica.es/old/actividades/jornadas/XXIX/pdf/300.pdf>
- Benasulin, D. A., Modesti, M. R., & Tanburi, D. O. (s. f.). *Delimitación de Zona de Riesgo, por medio de una Implementación de indicación RSSI con tecnología Wireless*. Recuperado a partir de <http://www.investigacion.frc.utn.edu.ar/sensores/aplicaciones/publicaciones/037.pdf>
- Carrascal Quintero, J. A. (2010). Almacenamiento de Reactivos y Sustancias Químicas. Recuperado a partir de http://www.usbcartagena.edu.co/phocadownload/facultades/salud/guia_almacenamiento_reactivos.pdf
- Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., & Urtasun, R. (2016). Monocular 3D Object

Detection for Autonomous Driving. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2147-2156. Recuperado a partir de https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Chen_Monocular_3D_Object_CVPR_2016_paper.pdf

Chicala, J., Jacho, R., Atencia, L., & Vintimilla, B. (2006). Reconocimiento y seguimiento de objetos móviles en un sistema de fútbol robótico. *Revista Tecnológica ESPOL*, xx. Recuperado a partir de https://www.researchgate.net/profile/Boris_Vintimilla/publication/28791943_Reconocimiento_Y_Seguimiento_De_Objetos_Moviles_En_Un_Sistema_De_Futbol_Robotico/links/00b7d532d34fdda76e000000/Reconocimiento-Y-Seguimiento-De-Objetos-Moviles-En-Un-Sistema-De-Fut

Choset, H., Lynch, K. M., Hutchinsom, S., Kantor, G., Burgard, W., Kavraki, L. E., & Thrun, S. (2005). Sampling-Based Algorithms. En MIT (Ed.), *Principles of Robot Motion* (pp. 197-262). London.

Colmenares, M. C. (2014). Basis for management of hazardous chemicals in teaching laboratories of the university of carabobo's chemical engineering school. *Ingeniería y Sociedad*, 9, 30-42. Recuperado a partir de <http://servicio.bc.uc.edu.ve/ingenieria/revista/IngenieriaySociedad/a9n1/art03.pdf>

Danelljan, M., Khan, F. S., Felsberg, M., & Van De Weijer, J. (2014). Adaptive Color Attributes for Real-Time Visual Tracking. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1090--1097. Recuperado a partir de http://openaccess.thecvf.com/content_cvpr_2014/papers/Danelljan_Adaptive_Color_Attributes_2014_CVPR_paper.pdf

De La Cruz, C., Carelli, R., & Gava, C. C. (2006). Control centralizado de formacion usando una cámara omnidireccional. *IV Jornadas Argentinas de Robótica*.

Diego, L. G. D., & Fernando, G. B. D. (2012). Tesis Doctoral Nuevas aportaciones en algoritmos de planificación para la ejecución de maniobras en robots autónomos no holónomos. Huelva. Recuperado a partir de

http://rabida.uhu.es/dspace/bitstream/handle/10272/5501/Nuevas_aportaciones_en_algoritmos_de_planificacion.pdf?sequence=2

DigiKey. (2017). Placas de desarrollo Teensy 3.5. Recuperado 20 de octubre de 2018, a partir de <https://www.digikey.com/es/product-highlight/s/sparkfun/teensy-3-5-development-boards>

Dong, Q., & Xu, X. (2014). A Novel Weighted Centroid Localization Algorithm Based on RSSI for an Outdoor Environment. *Journal of Communications*, 9, 279-285. <https://doi.org/10.12720/jcm.9.3.279-285>

Efrain Ernesto, A. V., Arturo, Z. L., & Juan, V. C. (2015). *Implementación de reconocimiento de objetos por color y forma en un robot móvil. Research in Computing Science*.

Etxeberria, J. A. (2010). *Algoritmo de reconocimiento de forma y color para una plataforma robótica*. Universidad del País Vasco / Euskal Herriko Unibertsitatea. Recuperado a partir de <http://www.ehu.es/documents/1545039/1570316/10jaristondo.pdf>

Fernández, T. M., Rodas, J., Iglesia, D. I., & Escudero, C. J. (s. f.). *Estabilización del RSSI en una red de sensores Bluetooth usando múltiples antenas*. Recuperado a partir de http://gtec.des.udc.es/web/images/pdfConferences/2007/ursi_fernandez_carames_2007.pdf

García-Alegre, M. C., Ribeiro, A., Guinea, D., & Sandoval, F. (2014). Localización de vehículos: fusión de medidas gps y odometría. *Mundo Electrónico*, 46-54. Recuperado a partir de <https://www.researchgate.net/publication/237469731>

Gonzales, R. C., & Woods, R. E. (2007). Color Image Processing. En Prentice Hall (Ed.), *Digital Image Processing* (3.^a ed., pp. 282-344). New Jersey.

González, J. I., & Cabrera Gámez, J. (2003). *Estudio experimental de métodos de calibración y autocalibración de cámaras*. Recuperado a partir de <http://gias720.dis.ulpgc.es/Gias/josep/TesisJosep.pdf>

González Jiménez, J., & Baturone, A. O. (2015). Estimación de la Posición de un Robot Móvil. *Automatica*, 29, 3-18. Recuperado a partir de https://www.researchgate.net/profile/Javier_Gonzalez-

Jimenez/publication/267222718_Estimacion_de_la_Posicion_de_un_Robot_Movil/links/54ba1f370cf24e50e93dd5de.pdf

Gonzalo Rafael Rojo Aguirre. (2014). *Seguridad y medio ambiente en planta química. QUIE0108*. IC Editorial. Recuperado a partir de https://books.google.es/books?id=RaenBAAAQBAJ&dq=ambientes+de+manipulacion+de+sustancias+tóxicas&lr=&hl=es&source=gbs_navlinks_s

How, J., King, E., & Kuwata, Y. (2004). Flight Demonstrations of Cooperative Control for UAV Teams IEEE Control Systems Magazine View project Gradient Projection Anti-windup View project Flight Demonstrations of Cooperative Control for UAV Teams. <https://doi.org/10.2514/6.2004-6490>

Hoyo Mondeja, H. A. (2012). *Localización de dispositivos móviles usando el nivel de intensidad de la señal recibida (RSSI)*. Universidad Central «Marta Abreu» de Las Villas. Recuperado a partir de http://dspace.uclv.edu.cu/bitstream/handle/123456789/1154/Hector_Alejandro_Hoyo_Mondeja.pdf?sequence=1&isAllowed=y

Inglett, J. E., & Rodríguez-Seda. (2017). Object transportation by cooperative robots. *SoutheastCon*, 1-6.

InvenSense Inc. (2013). *MPU-6000 and MPU-6050 Product Specification Revision 3.4*. Recuperado a partir de https://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3.4.pdf

ITEAD Studio. (2010). *Ultrasonic ranging module: HC-SR04*. Recuperado a partir de www.iteadstudio.com

Jiménez Camacho, E. (2009). *Medición de distancias por medio de procesamiento de imágenes y triangulación, haciendo uso de cámaras de video*. Escuela de Ingeniería, Universidad de las Américas Puebla.

Khan, A., Rinner, B., Cavallaro, A., Khan, A., Rinner, B., & Cavallaro, A. (2016). Cooperative Robots to Observe Moving Targets: Review Cooperative Robots to Observe Moving Targets: A Review. *IEEE TRANSACTIONS ON CYBERNETICS*, 20. Recuperado a partir de

<http://qmro.qmul.ac.uk/xmlui/handle/123456789/31982>

La Serna Palomino, N., & Román Concha, U. (2009). Técnicas de Segmentación en Procesamiento Digital de Imágenes. *Revista de Ingeniería de Sistemas e Informática*, 6(2), 9-16. [https://doi.org/10.1016/S0186-1042\(14\)70160-3](https://doi.org/10.1016/S0186-1042(14)70160-3)

Lara, A., Urrea, C., & Barbosa, K. A. (2015). Controller design of industrial cooperative robots. En *2015 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)* (pp. 73-78). IEEE. <https://doi.org/10.1109/Chilecon.2015.7400355>

Lin, L., Wang, X., Yang, W., & Lai, J.-H. (2015). Discriminatively Trained And-Or Graph Models for Object Shape Detection. *IEEE Transactions on pattern analysis and machine intelligence*, 37(5), 952-972. <https://doi.org/10.1109/TPAMI.2014.2359888>

López, D., Gómez-Bravo, F., Cuesta, F., & Ollero, A. (2010). Planificación de trayectorias con el algoritmo RRT. Aplicación a robots no holónomos. *Revista Iberoamericana de Automática e Informática Industrial*, 3, 56-67.

López Rodríguez Julio César. (2007). *Estructura, funcionamiento y aplicación de las Cámaras IP*". Universidad Autónoma del Estado de Hidalgo . Recuperado a partir de [http://dgsa.uaeh.edu.mx:8080/bibliotecadigital/bitstream/handle/231104/1747/Estructura, funcionamiento y aplicación de las cámaras IP.pdf?sequence=1](http://dgsa.uaeh.edu.mx:8080/bibliotecadigital/bitstream/handle/231104/1747/Estructura,funcionamiento_y_aplicación_de_las_cámaras_IP.pdf?sequence=1)

LUA. (s. f.). The Programming Language Lua. Recuperado 20 de octubre de 2018, a partir de <https://www.lua.org/>

Maria Garcia, E. (2008). Técnicas de localización en redes inalámbricas de sensores. *Treball publicat l'espai web de la Universidad de Castilla-La Mancha*. Recuperado a partir de https://www.researchgate.net/profile/Eva_Garcia30/publication/228705728_Tecnicas_de_Localizacion_en_Redres_Inalambricas_de_Sensores/links/58f45797458515ff23b48c42/Tecnicas-de-Localizacion-en-Redes-Inalambricas-de-Sensores.pdf

Martin, E., Vinyals, O., Friedland, G., & Bajcsy, R. (2010). *Precise Indoor Localization Using Smart Phones*. Recuperado a partir de

http://www1.icsi.berkeley.edu/~fractor/papers/friedland_120.pdf

Mejía, J. L. M. (2015). *Diseño de un sistema de video-vigilancia inalámbrico para la ciudad de Cayambe*. Escuela Politécnica Nacional.

Morales, L., Pozo, D.; Rosero, J., Sandobalin, S., & Rodríguez, M. (2014). Mapeo de Laberintos y Búsqueda de Rutas Cortas Mediante Tres Mini Robots Cooperativos. *Revista Politécnica*, 101-106.

Murray, R. M. (s. f.). Recent Research in Cooperative Control of Multivehicle Systems. <https://doi.org/10.1115/1.2766721>,

Nieto Duran, J. D., Santos Quintero, L. F., Escobar Vargas, L. J., & Salinas, S. A. (2017). Geolocalización para pacientes con alzhéimer: una propuesta. *Visión electrónica*, 11.

nodeMCU. (2014). NodeMcu - An open-source firmware based on ESP8266 wifi-soc. Recuperado 20 de octubre de 2018, a partir de http://nodemcu.com/index_en.html#fr_54747661d775ef1a3600009e

Owner, H. (2017). *Introduction to NodeMCU ESP8266*. Recuperado a partir de www.einstronic.com

PJRC. (s. f.). Teensy USB Development Board. Recuperado 20 de octubre de 2018, a partir de <https://www.pjrc.com/teensy/>

Possegger, H., & Mauthner, T. (2015). In Defense of Color-based Model-free Tracking. *n Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2113-2120. Recuperado a partir de https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Possegger_In_Defense_of_2015_CV_PR_paper.pdf

Revista Negocios de Seguridad. (2007). Cámara IP. *Negocios de Seguridad*, 31, 84-104. Recuperado a partir de http://www.rnds.com.ar/articulos/031/RNDS_084W.pdf

Ricolfe Viala, C. (2006). *Caracterización y optimización del proceso de calibrado de cámaras basado en plantilla bidimensional*. Universidad Politécnica de Valencia. Recuperado a partir

de <https://riunet.upv.es/bitstream/handle/10251/1858/tesisUPV2482.pdf>

Robert, F., Toibero, M. J., Vassallo, R., & Carelli, R. (2011). Control estable de formación basado en visión omnidireccional para robots móviles no holonómicos. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 8(1), 29-37. Recuperado a partir de https://ac.els-cdn.com/S169779121170005X/1-s2.0-S169779121170005X-main.pdf?_tid=4478cb1d-2adc-4595-83d2-87696ece2885&acdnat=1536666195_16f55682bffe7e999a4b1421fdb9836

Rodríguez, M., Pece, J. P., & Escudero, C. J. (2014). *Blueps: Sistema de localización en interiores utilizando Bluetooth*. In *XX Simposio Nacional de la URSI*. España. Recuperado a partir de <https://www.researchgate.net/publication/228885933>

Romero Macas, P. A. (2017). *Análisis de la posición y tiempo de detección para dispositivos móviles en un sistema de posicionamiento local en espacios interiores basado en el estándar IEEE 802.11 mediante la técnica Fingerprints y el filtro de Kalman, para aplicaciones de robótica c*. Universidad de las Fuerzas Armadas. Recuperado a partir de <https://repositorio.espe.edu.ec/bitstream/21000/13261/1/T-ESPE-057299.pdf>

Rugeles, J., & Leon, D. (2010). Técnicas de localización de nodos inalámbricos mediante redes de sensores. *GISSIC*. Recuperado a partir de http://www.iiis.org/CDs2013/CD2013SCI/CISCI_2013/PapersPdf/CA150PJ.pdf

Segura, J. L. (2010). *Sistema de auto localización a partir de una red WIFI*.

Soria, C., Carelli, R., Kelly, R., & Zannatha, J. M. I. (2004). Control de Robots Cooperativos por Medio de Vision Artificial. *XVI Congreso de la Asociación Chilena de Control Automático*.

SparkFun. (s. f.). *Advanced Arduino Comparison Guide - SparkFun Electronics*. Recuperado 20 de octubre de 2018, a partir de https://www.sparkfun.com/advanced_arduino_comparison_guide

ST. (2018). *VI53L0X*. Recuperado a partir de www.st.com

STMicroelectronics. (2000). *L298 dual full-bridge driver*. Recuperado a partir de

https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf

TP-Link. (2018). TL-WR840N | Router Inalámbrico N 300Mbps | TP-Link Colombia. Recuperado 26 de octubre de 2018, a partir de https://www.tp-link.com/co/products/details/cat-9_TL-WR840N.html

Vezhnevets, V., Sazonov, V., & Andreeva, A. (2003). A Survey on Pixel-Based Skin Color Detection Techniques. *GraphiCon Conference*.

Viala, C. R., José, A., & Salmerón, S. (2008). Procedimiento completo para el calibrado de cámaras utilizando una plantilla plana. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 5, 93-101. [https://doi.org/10.1016/S1697-7912\(08\)70126-2](https://doi.org/10.1016/S1697-7912(08)70126-2)

Xuesong, S., Wu, C., & Ming, L. (2008). Wireless Sensor Networks for Resources Tracking at Building Construction Sites. *Tsinghua Science & Technology*, 13(S1), 78-83. Recuperado a partir de http://158.132.160.122/bitstream/10397/7277/1/Shen_Wireless_Sensor_Networks.pdf