



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES**

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

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

**TEMA: “SISTEMA AÉREO TERRESTRE BASADO EN  
ROBÓTICA COOPERATIVA PARA DETECCIÓN DE MINAS  
ANTIPERSONALES”**

**AUTOR: CAMINO GUZMÁN, STEFANY MICHELLE  
ROJAS GÓMEZ, MILTON ISRAEL**

**DIRECTOR: DR. AGUILAR CASTILLO, WILBERT GEOVANNY.**

**SANGOLQUÍ**

**2019**



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA, Y  
TELECOMUNICACIONES**

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

**CERTIFICACIÓN**

Certifico que el trabajo de titulación "*SISTEMA AÉREO TERRESTRE BASADO EN ROBÓTICA COOPERATIVA PARA DETECCIÓN DE MINAS ANTIPERSONALES*" fue realizado por los señores *Camino Guzmán, Stefany Michelle y Rojas Gómez, Milton Israel* el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustenten públicamente.

Sangolquí, 28 de noviembre del 2019

.....  
**Dr. Wilbert G. Aguilar**  
CI. 0703844696



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA, Y  
TELECOMUNICACIONES

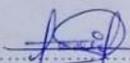
CARRERA DE INGENIERÍA EN ELECTRÓNICA,  
AUTOMATIZACIÓN Y CONTROL

AUTORÍA DE RESPONSABILIDAD

Nosotros, *Camino Guzmán, Stefany Michelle y Rojas Gómez, Milton Israel*, declaramos que el contenido, ideas y criterios del trabajo de titulación: "*Sistema aéreo terrestre basado en robótica cooperativa para detección de minas antipersonales*" es de nuestra autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 28 de noviembre del 2019

  
.....  
Camino Guzmán, Stefany Michelle  
CI. 1727495358

  
.....  
Rojas Gómez, Milton Israel  
CI. 1725651432



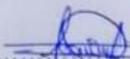
DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA, Y  
TELECOMUNICACIONES

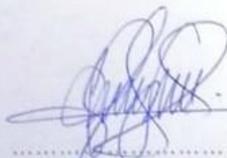
CARRERA DE INGENIERÍA EN ELECTRÓNICA,  
AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

*Nosotros, Camino Guzmán, Stefany Michelle y Rojas Gómez, Milton Israel, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: "Sistema aéreo terrestre basado en robótica cooperativa para detección de minas antipersonales" en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.*

Sangolquí, 28 de noviembre del 2019

  
.....  
Camino Guzmán, Stefany Michelle  
CI. 1727495358

  
.....  
Rojas Gómez, Milton Israel  
CI. 1725651432

## DEDICATORIA

A mis padres Mario y Nelly, quienes me han inculcado valores necesarios para alcanzar y cumplir con este objetivo, quienes jamás se han rendido y me han brindado su amor incondicional para superar cada adversidad, por lo cual dedico esta etapa que es parte de un gran camino por delante.

A mi tía Mercedes, quien me da su amor como a una hija, por tenerme paciencia y brindarme todo su afecto emocional, la cual está pendiente de mí siempre ya sea en los buenos o malos momentos, ayudándome a superarlos mediante consejos y experiencias, es por esto que dedico el esfuerzo y mis logros a ella.

A mi tía Beatriz, quien admiró eternamente por su dedicación, quien me enseña a no rendirme y seguir adelante, por el apoyo y la confianza que me brinda a diario.

A mi abuelita, quien me enseñó a ser una persona de bien junto con el conocimiento y valores que utilizó cada día, por toda la enseñanza que me sirve para cumplir los objetivos que se me presentan a diario.

A mis hermanos, Melany, Jorge, Nallely y Nadia, quienes me han brindado siempre su apoyo incondicional y me han compartido su ternura y sencillez a cada momento.

*Stefany Michelle Camino Guzmán*

## DEDICATORIA

Este trabajo quiero dedicarlo a mi madre Soledad, por brindarme amor y apoyo incondicional en todas las etapas de mi vida, sueños y metas propuestas, por ser mi fortaleza y la persona a la que siempre he admirado ya que con su perseverancia y amor hacia sus hijos ha sabido entregarnos todo lo necesario para poder salir adelante.

A mi padre Vinicio y hermano Andrés, por siempre apoyarme en las locuras que me he propuesto, por cada consejo y palabra de aliento para afrontar los problemas.

A mis abuelitos Imelda y Arturo, por ser esos seres tan maravillosos y estar siempre a mi lado apoyándome, por haber sido parte de mi formación como mis ejemplos de valores éticos que me han convertido en la gran persona que soy hoy en día, por todo el cariño y amor que me han dado. Este trabajo y mi formación como profesional no habrían sido posibles sin sus consejos de sabiduría y fortaleza.

A mi novia Jessica, por formar parte de mi vida y permitirme brindarle todo mi amor, por su apoyo incondicional en mis momentos de flaqueza.

*Milton Israel Rojas Gómez.*

## AGRADECIMIENTOS

Quiero agradecer primero a Dios y a mi familia, por haberme brindado la fortaleza necesaria para culminar con éxito mi carrera, y así afrontar las adversidades que se nos ha presentado a lo largo del camino.

A mi carrera, por la cual descubrí mi vocación la cual me proporciono las más emocionantes experiencias académicas, añadiendo a dos grandes amigos Katherine Gancino y Ariel Villacís, por su amistad y motivación hasta en los momentos más difíciles.

A mi amigo Israel Rojas, con quién compartí este trabajo ya que, con su entusiasmo, paciencia y apoyo hemos logrado desarrollar, es por esto que le quedo agradecida por las enseñanzas a lo largo de la carrera y por las palabras de aliento que siempre estuvieron presentes.

A nuestros amigos que se permitieron compartir junto a mí, el camino finaliza con este documento, por lo que agradezco ya que hicieron de mi instancia académica la experiencia más enriquecedora de la vida.

Finalmente, agradezco a mi universidad por todo el conocimiento adquirido, así como al Dr. Wilbert G. Aguilar el mismo que impartió en nosotros enseñanzas, aclarando las dudas que se iban presentando en el camino, por todo esto no me queda más que decir gracias ESPE.

*Stefany Michelle Camino Guzmán*

## **AGRADECIMIENTOS**

Agradezco a mi madre por el amor, cariño y apoyo brindado durante mi vida, y que me haya brindado la posibilidad de tener una educación de calidad pese a las dificultades, por haberme inculcado los buenos valores que destacan en mí.

Agradezco a mi padre y a mi hermano, por los momentos de felicidad que hemos vivido, por el apoyo y los consejos que me han brindado para no decaer ante las adversidades encontradas a lo largo de este logro profesional.

Agradezco a mis abuelitos por ser como mis padres hasta el día de hoy brindándome mucho amor y formarme como una persona de bien, por su apoyo y consejos siendo uno de los motivos para siempre luchar por alcanzar mis metas.

Agradezco mi novia Jessica por estar a mi lado en momentos difíciles de mi vida y de la carrera, por sus consejos y amor incondicional.

Finalmente agradezco a mi tutor de tesis el Doctor Wilbert de quien he recibido el conocimiento necesario para cumplir con el objetivo de este proyecto, le agradezco por su apoyo y guía durante este proceso.

*Milton Israel Rojas Gómez.*

## ÍNDICE DE CONTENIDOS

<b>CERTIFICACIÓN</b> .....	<b>i</b>
<b>AUTORÍA DE RESPONSABILIDAD</b> .....	<b>ii</b>
<b>AUTORIZACIÓN</b> .....	<b>iii</b>
<b>DEDICATORIA</b> .....	<b>iv</b>
<b>AGRADECIMIENTOS</b> .....	<b>vi</b>
<b>ÍNDICE DE CONTENIDOS</b> .....	<b>viii</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>i</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>ii</b>
<b>RESUMEN</b> .....	<b>i</b>
<b>ABSTRACT</b> .....	<b>ii</b>
<b>CAPÍTULO I</b> .....	<b>1</b>
<b>PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACIÓN</b> .....	<b>1</b>
1.1. Antecedentes .....	1
1.2. Justificación e importancia .....	5
1.3. Alcance .....	6
1.4. Objetivos.....	10
1.4.1. Objetivo General .....	10

1.4.2.	Objetivos Específicos.....	10
<b>CAPÍTULO II.....</b>		<b>11</b>
<b>FUNDAMENTO TEÓRICO .....</b>		<b>11</b>
2.1.	Robótica Cooperativa.....	11
2.1.1.	Robots móviles.....	11
2.1.2.	Vehículo aéreo no tripulado (UAV).....	14
2.1.3.	Robótica cooperativa.....	21
2.2.	Percepción visual .....	23
2.2.1.	Captura de imágenes de la cámara de un UAV.....	23
2.2.2.	Métodos de caracterización de una imagen .....	27
2.2.3.	Percepción visual .....	28
2.3.	Navegación servovisual .....	29
2.3.1.	Visión artificial.....	29
2.3.2.	Odometría visual .....	30
2.3.3.	Arquitectura de un sistema de visión. ....	30
2.3.4.	Transformaciones geométricas.....	31
2.3.5.	Navegación robótica.....	33
2.3.6.	Detección de obstáculos.....	35

2.4.	Guiado de sistema autónomos.....	36
2.4.1.	Planificación de ruta para cobertura de superficies.....	36
2.4.2.	Métodos de control visual.....	38
2.4.3.	Sistemas de detección de minas antipersonales .....	39
2.4.4.	Marcadores ARUCO.....	40
<b>CAPÍTULO III .....</b>		<b>41</b>
<b>GENERALIDADES DEL SISTEMA.....</b>		<b>41</b>
3.1.	Hardware del sistema .....	41
3.1.1.	Vehículo aéreo no tripulado Parrot bebop2 .....	41
3.1.2.	Plataforma móvil.....	44
3.1.3.	Estación en tierra.....	55
3.2.	Descripción y configuración del Software del sistema.....	56
3.2.1.	Entorno de desarrollo ROS (Robot Operating System).....	56
3.2.2.	Instalación de ROS en Ubuntu 16.04 LTS.....	57
3.2.3.	Driver bebop-autonomy .....	58
3.3.	Topología de red del sistema .....	64
<b>CAPÍTULO IV .....</b>		<b>65</b>

<b>SISTEMA DE NAVEGACIÓN AUTÓNOMA Y DETECCIÓN DE MINAS ANTIPERSONALES .....</b>	<b>65</b>
4.1. Sistema de Detección de Marcadores ARUCO .....	66
4.1.1. Captura y depuración de imagen .....	66
4.1.2. Detección y filtración de contornos validos .....	67
4.1.3. Identificación del marcador .....	71
4.2. Sistema de Estabilización de Posicionamiento del UAV en el entorno .....	75
4.2.1. Estimación del modelo de movimiento del UAV. ....	76
4.2.2. Diseño del controlador.....	79
4.2.3. Implementación del controlador. ....	83
4.3. Sistema de Percepción del Entorno de Trabajo .....	85
4.4. Sistema de Detección de Obstáculos .....	86
4.5. Sistema de Planificación de Trayectoria del Robot Móvil .....	88
4.6. Sistema de Control del Seguimiento de la Trayectoria del Robot Móvil .....	94
4.6.1. Estimación del modelo de movimiento del robot. ....	94
4.6.2. Diseño del controlador.....	97
4.6.3. Cálculo de la dirección y ángulo de giro del robot. ....	99
4.6.4. Implementación del controlador para posicionamiento angular del robot. ....	103

4.6.5.	Seguimiento de trayectoria.....	103
4.7.	Sistema de Detección de Minas Antipersonales. ....	104
<b>CAPÍTULO V .....</b>		<b>107</b>
<b>EVALUACIÓN, PRUEBAS Y ANÁLISIS DE RESULTADOS .....</b>		<b>107</b>
5.1.	Experimentación y resultados de Landmarks .....	107
5.1.1.	Lectura de marcadores .....	107
5.1.2.	Pruebas para detección de obstáculos .....	113
5.2.	Experimentación y resultados del sistema .....	119
5.2.1.	Pruebas para la detección de obstáculos .....	121
5.2.2.	Pruebas de iluminación y altura en el plano de trabajo.....	122
5.2.3.	Pruebas de planificación y seguimiento de trayectoria .....	123
5.2.4.	Pruebas de detección de minas antipersonales.....	126
5.2.5.	Pruebas de barrido de superficie .....	128
<b>CAPÍTULO VI .....</b>		<b>135</b>
<b>CONCLUSIONES Y RECOMENDACIONES .....</b>		<b>135</b>
6.1.	Conclusiones. ....	135
6.2.	Recomendaciones.....	137
6.3.	Trabajos Futuros.....	137

**REFERENCIAS BIBLIOGRÁFICAS .....138**

## ÍNDICE DE TABLAS

<b>Tabla 1</b> <i>Clasificación de los UAV's mediante la norma STANAG 4670.....</i>	17
<b>Tabla 2</b> <i>Modelos comerciales de cuadricópteros .....</i>	20
<b>Tabla 3</b> <i>Características técnicas del módulo TB6612FNG.....</i>	46
<b>Tabla 4</b> <i>Características de los sensores de proximidad inductivos LJ8A3-2 Z/BX .....</i>	47
<b>Tabla 5</b> <i>Características de los motores de transmisión. ....</i>	47
<b>Tabla 6</b> <i>Comparación de alternativas de configuración cinemática del robot.....</i>	49
<b>Tabla 7</b> <i>Especificaciones técnicas de la estación en tierra.....</i>	55
<b>Tabla 8</b> <i>Tópicos principales del driver Bebop_autonomy.....</i>	60
<b>Tabla 9</b> <i>Mensajes para el movimiento del dron con driver Bebop_autonomy.....</i>	61
<b>Tabla 10</b> <i>Mensajes para el movimiento de la cámara del drone .....</i>	62
<b>Tabla 11</b> <i>Comparación de reconocimiento de marcadores Aruco y Códigos QR.....</i>	108
<b>Tabla 12</b> <i>Resumen del reconocimiento de marcadores Aruco y Códigos QR.....</i>	113
<b>Tabla 13</b> <i>Pruebas experimentales para detectar obstáculos.....</i>	114
<b>Tabla 14</b> <i>Resumen de detección de obstáculos mediante el uso del color.....</i>	117
<b>Tabla 15</b> <i>Pruebas experimentales para detectar obstáculos mediante la morfología. ....</i>	118
<b>Tabla 16</b> <i>Características de interés en los escenarios de pruebas experimentales. ....</i>	120
<b>Tabla 17</b> <i>Secuencia de acciones para cada escenario de pruebas experimentales. ....</i>	120
<b>Tabla 18</b> <i>Resultados de la detección de obstáculos en diversos escenarios de iluminación .....</i>	121
<b>Tabla 19</b> <i>Pruebas de iluminación y altura. ....</i>	122

<b>Tabla 20.</b> <i>Resumen de tiempos y promedios</i> .....	123
<b>Tabla 21</b> <i>RMSE objetivo ideal vs real</i> .....	124
<b>Tabla 22</b> <i>Pruebas de seguimiento y planificación de trayectorias</i> .....	124
<b>Tabla 23</b> <i>Resultados de la detección de minas</i> .....	126
<b>Tabla 24</b> <i>Pruebas de barrido de superficie</i> .....	129
<b>Tabla 25</b> <i>Prueba 1 de efectividad de cobertura de superficie</i> .....	131
<b>Tabla 26</b> <i>Prueba 2 de efectividad de cobertura de superficie</i> .....	132
<b>Tabla 27</b> <i>Prueba 3 de efectividad de cobertura de superficie</i> .....	133

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> <i>Áreas de trabajo del sistema</i> .....	7
<b>Figura 2.</b> <i>Enjambre de robots cooperativos</i> .....	7
<b>Figura 3.</b> <i>Estructura básica de un robot móvil</i> .....	12
<b>Figura 4.</b> <i>Clasificación de los robots móviles</i> .....	13
<b>Figura 5.</b> <i>Tipos de locomoción de un robot móvil</i> .....	13
<b>Figura 6.</b> <i>Ejemplo de UAV de cuatro rotores</i> .....	15
<b>Figura 7.</b> <i>Clasificación de los UAV de acuerdo a su tipo de alas</i> .....	16
<b>Figura 8.</b> <i>Clasificación de los UAV mediante el sistema de control</i> .....	17
<b>Figura 9.</b> <i>Aplicaciones de los UAV</i> .....	18
<b>Figura 10.</b> <i>Configuración de rotores en: a) Cruz (+) b) Equis (X)</i> .....	19

<i>Figura 11.</i> Ángulos de navegación de un cuadricóptero .....	19
<i>Figura 12.</i> Robótica cooperativa .....	22
<i>Figura 13.</i> Proceso de formación de una imagen digital.....	24
<i>Figura 14.</i> Imagen en representación de píxeles .....	25
<i>Figura 15.</i> Modelo de la cámara pinhole.....	25
<i>Figura 16.</i> Proyección de puntos en la imagen.....	26
<i>Figura 17.</i> Diagrama de bloques de un sistema de visión artificial.....	31
<i>Figura 18.</i> Traslación de una imagen. ....	32
<i>Figura 19.</i> Rotación de una imagen.....	32
<i>Figura 20.</i> Ejemplo de transformación afín.....	33
<i>Figura 21.</i> Típico camino en zigzag .....	37
<i>Figura 22.</i> Descomposición trapezoidal de superficies.....	38
<i>Figura 23.</i> Descomposición de boustrophedon. ....	38
<i>Figura 24.</i> Ejemplo de marcador Aruco de rejilla de 5x5 bits. ....	40
<i>Figura 25.</i> Parrot Bebop 2 .....	41
<i>Figura 26.</i> Dimensiones del Bebop 2 .....	43
<i>Figura 27.</i> Ángulos de visión de la cámara del Bebop 2.....	43
<i>Figura 28.</i> Ejemplo de dimensiones de la plataforma móvil.....	44
<i>Figura 29.</i> Diagrama de bloques del robot móvil.....	45
<i>Figura 30.</i> Esquema de configuración diferencial.....	48
<i>Figura 31.</i> Esquema de configuración Skid-steer .....	49
<i>Figura 32.</i> Modelo 3D de las piezas del robot. ....	50

<b>Figura 33.</b> Piezas del robot listas para ser ensambladas.....	51
<b>Figura 34.</b> Etapas para la fabricación de la PCB.....	52
<b>Figura 35.</b> Esquemático del circuito electrónico del robot.....	52
<b>Figura 36.</b> Diagrama multicapa de la PCB.....	53
<b>Figura 37.</b> PCB del robot móvil visualizado en 3D. ....	53
<b>Figura 38.</b> PCB final del robot móvil.....	54
<b>Figura 39.</b> Placa de control del robot terminada. ....	54
<b>Figura 40.</b> Proceso de ensamblaje del robot.....	55
<b>Figura 41.</b> Interacción entre ROS y OpenCV .....	60
<b>Figura 42.</b> Esquema de coordenadas del UAV. ....	61
<b>Figura 43.</b> Ejemplo de configuración del de los movimientos del drone.....	62
<b>Figura 44.</b> Topología de la red del sistema .....	64
<b>Figura 45.</b> Esquema total de sistema.....	65
<b>Figura 46.</b> Diagrama de flujo del proceso de detección de marcadores Aruco.....	66
<b>Figura 47.</b> Conversión de imagen a escala de grises.....	66
<b>Figura 48.</b> Binarización de umbral adaptativo. ....	67
<b>Figura 49.</b> Contornos de la imagen. ....	68
<b>Figura 50.</b> Depuración de contornos. ....	68
<b>Figura 51.</b> Aproximación de contorno. ....	69
<b>Figura 52.</b> Aproximación de contorno de los marcadores. ....	69
<b>Figura 53.</b> Casco convexo de una forma. ....	70
<b>Figura 54.</b> Obtención de la perspectiva del marcador.....	70

<b>Figura 55.</b> Diagrama de flujo del proceso de detección y filtración de contornos validos .....	71
<b>Figura 56.</b> Resultado de la binarización de Otsu.....	71
<b>Figura 57.</b> Erosión de la imagen. ....	72
<b>Figura 58.</b> Redimensionamiento de imagen.....	72
<b>Figura 59.</b> Partes de los marcadores ARUCO.....	73
<b>Figura 60.</b> Generación de marcadores Aruco.....	73
<b>Figura 61.</b> Orden de los bits del marcador. ....	74
<b>Figura 62.</b> Marcador encontrado sin el contorno. ....	74
<b>Figura 63.</b> Marcador en la posición deseada. ....	74
<b>Figura 64.</b> Resultado de graficar el borde en el marcador y su ID.....	75
<b>Figura 65.</b> Puntos de referencia entre fotogramas.....	76
<b>Figura 66.</b> Tren de pulsos de entrada para Roll y Pitch .....	77
<b>Figura 67.</b> Actuación del sistema para el control en Roll. ....	77
<b>Figura 68.</b> Actuación del sistema para el control en Pitch.....	78
<b>Figura 69.</b> Caracterización del modelo para la planta en Roll. ....	79
<b>Figura 70.</b> Caracterización del modelo para la planta en Pitch.....	79
<b>Figura 71.</b> Esquema de control del posicionamiento del dron mediante la posición visual.....	80
<b>Figura 72.</b> Esquema del diseño del controlador. ....	80
<b>Figura 73.</b> Respuesta al impulso para la planta en Roll. ....	82
<b>Figura 74.</b> Respuesta al impulso para la planta en Pitch.....	83
<b>Figura 75.</b> Error de posición del centro de la zona minada con respecto al centro.....	83
<b>Figura 76.</b> Controlador del dron en funcionamiento. ....	84

<i>Figura 77.</i> Numeración de las esquinas del marcador.....	85
<i>Figura 78.</i> Posición de los marcadores para delimitar el plano de trabajo.....	85
<i>Figura 79.</i> Obtención del plano de trabajo. ....	86
<i>Figura 80.</i> Binarización del espacio de trabajo. ....	87
<i>Figura 81.</i> Determinación del espacio de trabajo válido.....	88
<i>Figura 82.</i> Resultado del algoritmo de Boustrophedon para cobertura de superficies.....	89
<i>Figura 83.</i> Descripción de las medidas de la trama de la planificación de trayectoria. ....	90
<i>Figura 84.</i> Resultado algoritmo de Boustrophedon.....	91
<i>Figura 85.</i> Máscara basada en color.....	92
<i>Figura 86.</i> Superposición de máscara y planificación de Boustrophedon. ....	92
<i>Figura 87.</i> Ejemplo del algoritmo de evasión de obstáculos.....	93
<i>Figura 88.</i> Resultado del algoritmo Boustrophedon modificado para evadir obstáculos.....	94
<i>Figura 89.</i> Ejemplo de variación del ángulo de giro del robot. ....	95
<i>Figura 90.</i> Diagrama de prueba para el modelo del robot.....	95
<i>Figura 91.</i> Tren de pulsos de entrada para el control del ángulo ....	96
<i>Figura 92.</i> Comportamiento del movimiento del robot móvil ....	96
<i>Figura 93.</i> Identificación de la planta del robot móvil ....	97
<i>Figura 94.</i> Esquema de control para el posicionamiento angular del robot móvil.....	98
<i>Figura 95.</i> Respuesta al impulso unitario del controlador de giro. ....	99
<i>Figura 96.</i> Coordenadas para el seguimiento de la trayectoria para el control servovisual .....	100
<i>Figura 97.</i> Triángulo escaleno para el cálculo del ángulo $\alpha$ de giro del robot móvil.....	100
<i>Figura 98.</i> Direccinamiento del robot. ....	101

<b>Figura 99.</b> Diagrama de flujo del ángulo de rotación del robot móvil.....	102
<b>Figura 100.</b> Diagrama de flujo del control de seguimiento de trayectoria del robot móvil. ....	104
<b>Figura 101.</b> Circuito para acoplar el sensor de proximidad inductivo .....	104
<b>Figura 102.</b> Posiciones de los sensores en el marcador del robot. ....	105
<b>Figura 103.</b> Diagrama de flujo del proceso para detectar minas. ....	106
<b>Figura 104.</b> Reconocimiento de códigos QR. ....	107
<b>Figura 105.</b> Secuencia para determinar el error del control servovisual. ....	123
<b>Figura 106.</b> Separación del marcador del fondo de la imagen. ....	128
<b>Figura 107.</b> Zonas de conflicto para la cobertura de superficie. ....	134

## **RESUMEN**

En varios campos de la industria y actividades cotidianas del humano es necesario que los robots móviles puedan cubrir una superficie en su totalidad evadiendo los obstáculos existentes en el medio, y trabajar entre dos o más de ellos para cumplir una tarea específica denominada robótica cooperativa, todo esto con el objetivo de mejorar la calidad de vida de las personas. Es el caso del desminado de minas antipersonales donde el operador corre mucho riesgo de sufrir accidentes recorriendo estos campos contaminados con detectores metálicos. En consecuencia al presente trabajo de investigación se encamina para el desarrollo de un sistema de navegación mediante la planificación de trayectoria de cobertura de superficies con un robot móvil de configuración diferencial, en entornos definidos con imágenes de perspectiva cenital capturadas por un dron, esto se realiza mediante el algoritmo de Boustrophedon el cual utiliza un método de cobertura en zigzag del área, luego obteniendo la máscara de color de los obstáculos, se encuentra los puntos de intersección de entrada y salida del zigzag con los obstáculos para generar las nuevas rutas más cortas a través del borde del obstáculo. Para determinar la orientación del robot se realiza el algoritmo de reconocimiento de marcadores aruco que estiman la posición del robot en todo instante. Se utilizó un controlador PI para posicionar el dron en el centro del área minada, además para el control servo-visual se implementó un controlador PID que orienta al robot en la trayectoria.

### **PALABRAS CLAVE:**

- **ROBOT MÓVIL**
- **PLANIFICACION DE TRAYECTORIA COBERTURA DE SUPERFICIE**
- **SISTEMA DE NAVEGACIÓN**

## **ABSTRACT**

In various fields of industry and everyday activities of the human it is necessary that mobile robots can cover a surface in its entirety by avoiding existing obstacles in the middle, and work between two or more of them to fulfill a specific task called cooperative robotics, all This with the aim of improving the quality of life of people. This is the case of demining of antipersonnel mines where the operator is at high risk of accidents by crossing these fields contaminated with metal detectors. For this reason, this research project is aimed at the development of a navigation system through trajectory planning for surface coverage with a mobile robot of differential configuration, in defined environments with zenith perspective images provided by a drone, this is done using the Boustrophedon algorithm which uses a simple method of zigzag coverage of the entire area, then obtaining the color mask corresponding to the obstacles, you find the intersection points of entry and exit of the zigzag with the obstacles to generate the new Shorter routes through the edge of the obstacle. To determine the orientation of the robot, the aruco marker recognition algorithm is estimated that estimates the robot's position at all times. A PI controller was used to position the drone in the center of the mined area, in addition to the servo-visual control a PID controller was implemented that guides the robot towards the point of the path.

### **KEYWORDS:**

- **MOBILE ROBOT**
- **SURFACE COVERAGE TRAJECTORY PLANNING**
- **NAVIGATION SYSTEM**

## CAPÍTULO I

### PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACIÓN

#### 1.1. Antecedentes

La robótica cooperativa fue creada para dar soluciones a varios problemas que la humanidad ha querido resolver a lo largo de la historia, y que no pueden ser desarrollados por un solo robot. Es muy utilizada para desarrollar de una manera más simple varias actividades de la vida cotidiana de los seres humanos. Se los puede encontrar en aplicaciones como exploración de espacios desconocidos, creación de mapas, operaciones de seguridad o transporte de objetos; siendo esencial que los robots siempre se encuentren en una formación específica sin importar el movimiento que el grupo realice. (Solaque, 2015). Según el trabajo de James S. Albus nos indica que este tipo de robots requieren una arquitectura de sistemas de control que vinculen los actuadores, sensores, el procesamiento sensorial, la descomposición de las tareas; y es importante que todos compartan un objetivo en común. (Albus, 2015)

Actualmente, dentro de la robótica móvil se pueden distinguir dos tipos, la robótica aérea (Aguilar, Casaliglla, & Pólit, Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles, 2017), (Orbea, y otros, Vertical take off and landing with fixed rotor, 2017), (Orbea, y otros, Math Model of UAV Multi Rotor Prototype with Fixed Wing Aerodynamic Structure for a Flight Simulator, 2017) y terrestre (Galarza, Pérez, Serrano, Tapia, & Aguilar, 2018), (Segarra, Caballeros, & Aguilar, 2018), (Aguilar W. G., Sandoval, Limaico, Villegas-Pico, & Asimbaya, 2019) cada una de ellas con capacidades diferenciadas. A pesar de que los sistemas terrestres disponen de una mayor autonomía y pueden llevar a bordo ordenadores con una alta capacidad de

computo, se ven limitados por el conocimiento del entorno. Incluso contando con múltiples sensores capaces de aportar una gran cantidad de datos sobre lo que rodea al robot, obtener esta información desde el plano del suelo implica un alto costo de tiempo dedicado a realizar una exploración exhaustiva, lo que conlleva una disminución de las probabilidades de éxito de las misiones.

Con el fin de solventar las desventajas de ambas tecnologías, se han propuesto diferentes sistemas mixtos entre robots aéreos y terrestres. Algunas de estas soluciones proponen el uso de plataformas móviles como base de operaciones para los robots aéreos, aumentando su autonomía e implementando sistemas de comunicación (Jara-Olmedo A. , Medina-Pazmiño, Tozer, Aguilar, & Pardo, 2018) que permitan que estos puedan transmitir la información que están captando para su procesamiento en tierra. Otras centradas a solventar las desventajas de los sistemas terrestres, utilizan los robots aéreos para escanear y mapear el entorno (Basantes, y otros, 2018), (Aguilar, y otros, Real-Time 3D Modeling with a RGB-D Camera and On-Board Processing, 2017), (Aguilar, y otros, On-Board Visual SLAM on a UGV Using a RGB-D Camera, 2017), (Aguilar, y otros, Visual SLAM with a RGB-D Camera on a Quadrotor UAV Using on-Board Processing, 2017), (Aguilar, Cobeña, Rodriguez, Salcedo, & Collaguazo, 2018), lo que permite una planificación eficiente del movimiento en el suelo. (Leal, 2018), (Salcedo, Aguilar, Cobeña, & Proaño, 2018).

La navegación y la localización de robots son los mayores problemas que se presentan a la hora de conseguir el movimiento de los mismos dentro de un entorno definido, para esto es necesario conocer la ruta (Aguilar, y otros, 2018), (Aguilar, Quisaguano, Alvarez, Pardo, & Proaño, 2018) que permite la navegación de forma segura evitando obstáculos (Aguilar, Casaliglla, & Pólit, Obstacle Avoidance for Low-Cost UAVs, 2017), (Aguilar, Casaliglla, Pólit,

Abad, & Ruiz, Obstacle Avoidance for Flight Safety on Unmanned Aerial Vehicles, 2017) que se puedan encontrar en el camino.

En la investigación de N. Sariff se menciona que desde 1980 hasta hoy se han desarrollado varios tipos de algoritmos de planificación de trayectoria propuestos por los ingenieros para resolver el problema de localización y navegación de un robot en un entorno definido (Buniamin, Sariff, Wan, & Mohamad, 2011), entre los avances más destacados se tiene: encontrar el camino a la meta con éxito, optimización de trayectorias, reducción tiempos y esfuerzos al seguir la ruta, algoritmos capaces de adaptarse al medio, aplicación de redes neuronales y algoritmos genéticos. Estos dos últimos no solo son capaces de encontrar rutas óptimas, sino que también son adaptables y robustos tanto en entornos estáticos como dinámicos (Aguilar W. G., y otros, 2018).

Un ejemplo claro de estos avances es el algoritmo Lazy PRM que puede minimizar el número de controles de colisión realizados durante la planificación, por lo tanto, se reduce el tiempo de ejecución del planificador (Bohlim & Kavraki, 2000). Si el robot móvil conoce su entorno y la ruta, se puede planificar la trayectoria antes de que el robot comience a moverse.

Andrés Felipe Silva en su artículo establece que al utilizar un dispositivo aéreo se deben implementar ciertas estrategias de control servo-visual para determinar la orientación, la posición y así seguir una trayectoria marcada mediante un sistema de inspección y vigilancia que procese las imágenes de la cámara a bordo con el fin de realizar una navegación autónoma o semi autónoma. (Bohórquez, Mendoza, & Cortés, 2013)

Según F. Chaumette para el control servo-visual se han propuesto diferentes algoritmos, los cuales están basados en establecer con precisión la estimación de objetos a velocidad, considerando la posición y el movimiento de la cámara. (Chaumette, Rives, & Espiau, 2014)

Los vehículos aéreos no tripulados (UAV) se han utilizado para aplicaciones militares o civiles los mismos que se desarrollan para la inspección de terrenos, prevención de desastres, vigilancia ambiental, búsqueda y rescate, cartografía aérea. (Min, Shin, & Tahk, 2015), (Andrea, Byron, Jorge, Inti, & Aguilar, 2018), (Jara-Olmedo A. , y otros, 2018), (Zurita, Aguilar, & Victor, 2019), (Pardo, Aguilar, & Toulkeridis, 2017), (Amaguaña, Collaguazo, Tituaña, & Aguilar, 2018).

Una de las aplicaciones militares de los robots móviles terrestres es el desminado de terrenos. De acuerdo con el Comité Internacional de la Cruz Roja, en el 50% de los accidentes con minas las personas mueren, y entre las heridas una buena parte termina con amputaciones, fuera de los daños psicológicos causados. Las víctimas son con frecuencia quienes las colocan, o niños que juegan con ellas.

El conflicto bélico entre el Ecuador y Perú no solo ocasionó la muerte de soldados, sino también la contaminación de territorios con minas y restos de explosivos de guerra. En su mayoría las minas fueron sembradas en la frontera común, en Ecuador las provincias afectadas son El Oro, Loja, Zamora Chinchipe, Morona Santiago y Pastaza. De acuerdo con los registros de víctimas por minas reportadas por el Programa de Acción Integral Contra las Minas de la OEA, en el Perú 300 víctimas fueron afectadas por minas, incluyendo civiles, policías y militares. En el lado ecuatoriano, se registró un total de 70 víctimas, de las cuales 51 fueron parte de las Fuerzas Armadas del Ecuador y 19 corresponden a la población civil. (Daniela, 2017)

## **1.2. Justificación e importancia**

El desarrollo de las tareas de desminado humanitario en territorio ecuatoriano inició tras la suscripción de los acuerdos de paz entre Ecuador y Perú en 1998 y posteriormente con la adhesión de ambos países a la Convención de Ottawa sobre minas antipersonales. Desde ese entonces han pasado más de 20 años de haber manifestado el compromiso de erradicar las minas de la frontera ecuatoriana y todavía no se ha podido culminar con la difícil tarea.

La zona fronteriza de Twintza ubicada en Morona Santiago se encuentra todavía contaminada por alrededor de 200 minas antipersonales cuya detonación no se produjo al momento de ser lanzadas, pero que conllevan aún un peligro latente de detonar, uno de los compromisos que acordaron Ecuador y Perú fue el desminado en sus fronteras comunes. Ecuador cuenta con la Unidad de Desminado Humanitario del Ejército ecuatoriano que se encarga de limpiar al país de explosivos, además, tiene la responsabilidad de monitorizar la amenaza de las minas antipersonales con el fin de identificar los lugares exactos donde se encuentran estas y desarrollar respuestas para la coordinación de soluciones. Por otro lado, pese a los peligros que estos campos presentan, los números de víctimas por minas antipersonales siguen siendo importantes y requieren más alternativas que ayuden a reducir el número de incidentes provocados por la falta de mecanismos de reconocimiento de estos campos minados.

Mediante este proyecto se busca desarrollar un prototipo que servirá como base para generar mecanismos que ayudarán a salvaguardar la vida del personal militar que labora en estos territorios y de las personas que se encuentran habitando en las áreas contaminadas, además de diseñar un sistema que ayudará en la detección de minas antipersonales, evitando así que el operador tenga cualquier tipo de contacto con el área contaminada y de esta forma se pueda desenvolver sin riesgo en su trabajo. Varios accidentes en el desminado se deben a que los operadores de los detectores de

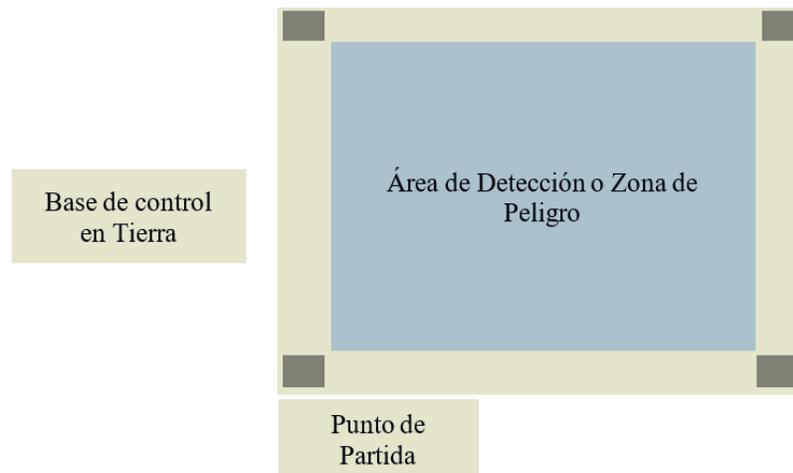
metales entran inseguros y con miedo al área de detección aumentando la probabilidad de sufrir un accidente, el prototipo que se construirá proporcionará seguridad al operador que supervisará el funcionamiento del mismo de forma remota.

Al culminar este proyecto se dará a conocer los beneficios que la robótica cooperativa facilita, contribuyendo así con el desarrollo de nuevos conocimientos y experiencia en el ámbito industrial y académico.

### **1.3. Alcance**

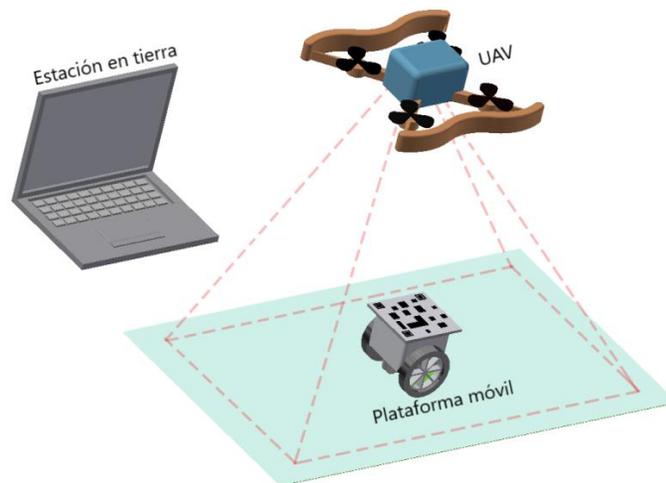
El proyecto tiene la finalidad de desarrollar un sistema aéreo-terrestre basado en robótica cooperativa para detectar minas antipersonales en un área determinada previamente por el operador, el sistema estará constituido por tres áreas como se ilustra en la Figura 1, las mismas se describen a continuación:

- Área de Detección: Zona delimitada en la cual se desplazará el robot móvil en busca de minas antipersonales.
- Punto de Partida: Localización segura en la que el operador colocará el UAV y el robot terrestre, donde partirá hacia el área de detección designada.
- Estación de Control en Tierra: Base en la que será procesada toda la información y se enviarán las acciones de control al robot móvil y al UAV.



**Figura 1.** Áreas de trabajo del sistema

Para obtener la visualización cenital de la zona de peligro y del robot móvil como se observa en la Figura 2 se realizará el posicionamiento automático de un vehículo aéreo no tripulado (UAV) multirrotor de la categoría V. (Kendoul, 2012)



**Figura 2.** Enjambre de robots cooperativos

Para el control del UAV se pretende desarrollar mediante la captura de imágenes y las condiciones de información que se proporciona de la IMU. A partir de las imágenes obtenidas se estimarán los parámetros de movimiento que servirán de consigna para el control de la trayectoria.

La planificación de trayectoria que seguirá el robot móvil recorrerá por completo toda la zona de peligro, dado el caso que existan obstáculos no dinámicos la trayectoria a seguir deberá esquivar los mismos. Se utilizarán algoritmos de descomposición de celdas en los fotogramas que previamente se pasarán por una máscara de color para determinar las zonas en las que no hay obstáculos y realizar la planificación.

Para llevar a cabo la tarea de desplazamiento en tierra se diseñará y construirá desde cero un robot móvil que pueda trasladarse en todas las direcciones y en su propio eje, con lo cual se asegurará que pueda llegar a cualquier punto de la zona de peligro.

Como parte del procesamiento de imágenes se utilizará algoritmos de detección de marcadores para determinar la posición del robot móvil y delimitar la zona de peligro, esto conlleva a realizar transformaciones geométricas entre los fotogramas consecutivos para asegurar que la vista sea totalmente cenital y el movimiento del UAV por perturbaciones no afecte en la estabilización del video y por ende en el cálculo de la posición del robot móvil ni en el control del mismo.

Los algoritmos desarrollados serán implementados en la estación en tierra mediante ROS, el cual tendrá una comunicación Wi-Fi entre el UAV y el robot móvil.

El trabajo de detección de metales se efectuará con sensores acoplados al robot móvil, los cuales estarán conectados al controlador implementado en el robot que se comunicará con la estación en tierra indicando las coordenadas en las que se detectaron posibles explosivos, tomando en cuenta la clasificación de los mecanismos de activación de las minas antipersonales el sistema está orientado a operar para los siguientes tipos:

- Minas antipersonales con mecanismos de presión.

- Minas antipersonales con mecanismos de alivio de presión.

Finalmente, el sistema se pondrá a prueba con objetos metálicos de las mismas dimensiones de estos dos tipos de minas, y en varios escenarios que permitan determinar su desempeño.

Para el presente proyecto se cumplirá con las siguientes etapas:

- Diseñar e implementar el software y hardware para el robot móvil.
- Implementar el sistema de comunicación del robot móvil y el UAV con la estación terrestre.
- Desarrollar el algoritmo de posicionamiento del UAV.
- Desarrollar el algoritmo de detección de marcadores para determinar la posición e identificación del robot móvil.
- Diseñar e implementar el algoritmo para la planificación de la trayectoria del robot móvil durante el proceso de detención de minas antipersonales.
- Desarrollar el sistema control servo-visual para el seguimiento de trayectoria.
- Diseñar la interface en la estación terrestre para visualizar la ruta definida que deben seguir el robot móvil terrestres y su seguimiento servo-visual respectivo, además de marcar los puntos con sus coordenadas donde existan posibles minas antipersonales.
- Efectuar pruebas de funcionamiento del sistema en varios escenarios para determinar la eficacia del mismo.
- Finalmente, se presentarán las conclusiones y recomendaciones obtenidas en el desarrollo del proyecto.

## **1.4.Objetivos**

### **1.4.1. Objetivo General**

- Diseñar e implementar un sistema aéreo-terrestre basado en robótica cooperativa para la detección de minas antipersonales como prueba de conceptos.

### **1.4.2. Objetivos Específicos**

- Diseñar y construir una plataforma móvil para escenarios definidos, con sensores para detección de metales a bordo.
- Implementar el sistema de comunicación del robot móvil y el UAV con la estación terrestre.
- Desarrollar el algoritmo de posicionamiento del UAV.
- Desarrollar el algoritmo de detección de los marcadores para determinar la posición e identificación del robot móvil.
- Diseñar e implementar el algoritmo de planificación de rutas del robot móvil durante el proceso de detección de minas antipersonales.
- Desarrollar el sistema de control servo-visual para seguimiento de trayectorias.
- Efectuar pruebas de funcionamiento del sistema en varios escenarios para determinar la eficacia del mismo.

## **CAPÍTULO II**

### **FUNDAMENTO TEÓRICO**

Con el fin de sustentar teóricamente las herramientas y conceptos a utilizar en el proyecto de investigación, en el presente capítulo se realiza una breve descripción del software y hardware para su desarrollo.

#### **2.1. Robótica Cooperativa**

##### **2.1.1. Robots móviles**

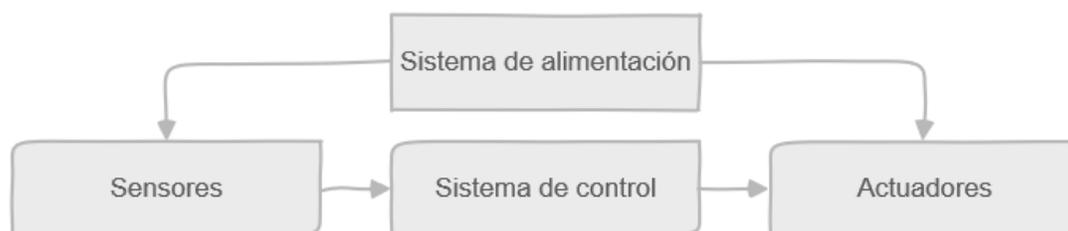
Los robots móviles tuvieron un gran auge en la década de los setenta, el avance en tecnología y diseño de configuraciones permitió la construcción del robot “Shakey” siendo el primer robot con inteligencia artificial capaz de controlar sus propios movimientos. A finales de esta década, Moravec desarrolló a “Stanford Cart” un robot capaz de seguir una trayectoria definida por una línea sobre una superficie.

Para la década de los noventa el MIT (Instituto de Tecnología de Massachusetts) desarrolló el robot “Spring Flamingo” que emulaba el movimiento del propio animal referido en su nombre, fue diseñado con el objetivo de conocer las técnicas de control necesarias para el manejo de los actuadores y de esta manera poder describir el movimiento necesario para ejecutar los algoritmos de desplazamiento. (Ortigoza & Sánchez, 2015)

En el año 2006, Hollis desarrolló el robot “Ballbot” provisto con un sistema holónimo, su movimiento es generado por una esfera ubicada en la parte interior de la estructura, empleando el uso de sensores como encoders que determinan la posición del mismo. (Barrientos, García, & Silva, 2007)

##### **2.1.1.1. Estructura**

El robot móvil presenta una estructura basada en cuatro funciones específicas tales como se observa en la Figura 3:



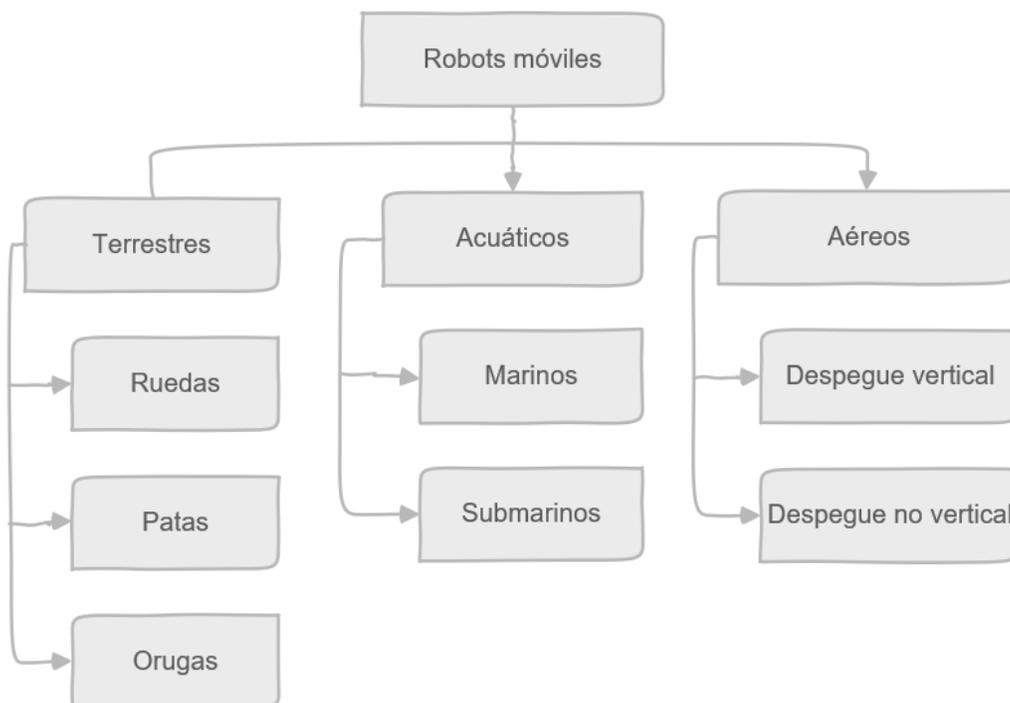
**Figura 3.** Estructura básica de un robot móvil

Fuente: (Agudelo & Garces, 2006)

- **Sistema de alimentación:** la fuente que suministra energía a los sensores y actuadores para el funcionamiento del robot móvil.
- **Sensores:** son elementos capaces de proporcionar información del entorno donde se encuentre, existen diversos tipos entre ellos se encuentran sensores como: laser, cámaras, infrarrojos, etc.
- **Sistema de Control:** permite interactuar con el entorno mediante la recepción de información de los sensores.
- **Actuadores:** elementos que permiten interactuar físicamente con el entorno, como las ruedas, motores, palancas, brazos, luces entre otros.

#### 2.1.1.2. Clasificación

Los robots móviles poseen varios tipos de movilidad, pues de esto depende que pueda realizar las tareas programadas con eficacia, teniendo en cuenta este criterio su clasificación es la siguiente:



**Figura 4.** Clasificación de los robots móviles

Fuente: (Márquez, Silva, & Marcelino, 2014)

Los robots móviles terrestres, a su vez pueden clasificarse de acuerdo al tipo de locomoción, los tres medios de movimiento son por: patas, ruedas y orugas. Aunque, la locomoción por patas u orugas han sido construidas y evaluadas en trabajos de investigación que utilizan ruedas para desplazarse (Barrientos, García, & Silva, 2007). En la Figura 5 muestra los distintos tipos de locomoción de los robots móviles:



**Figura 5.** Tipos de locomoción de un robot móvil

Fuente: (Barrientos, García, & Silva, 2007)

### **2.1.1.3. Aplicaciones de robots móviles.**

Las aplicaciones de los robots móviles tienen una amplia cobertura ya que pueden ser de uso doméstico o hasta formar parte de procesos industriales, también se encuentran sistemas de vigilancia al igual que aplicaciones militares, sectores en los que se han desarrollado en mayor escala. Algunos de los procesos en los que han sido aplicados son: exploración minera, exploración planetaria, misiones de búsqueda y rescate de personas, automatización de procesos, vigilancia, reconocimiento de terreno, agricultura, transporte, etc. (Bermúdez, 2002).

### **2.1.1.4. Tendencias**

Los avances en robots móviles proyectan que tendrán una utilización masiva en cuanto al desarrollo de tecnología militar, en el festival del 2019 organizado por los países europeos dedicados a la investigación en el campo de la robótica, se presentaron varios robots diseñados por la industria, uno de estos construido por Robomotion es destinado a realizar misiones militares, reconocimientos, vigilancia y manejo de objetos peligrosos de alto riesgo. Otro es el caso del manipulador móvil llamado X-WAM, para cumplir labores de manipulación en interiores y exteriores.

La Universidad Politécnica de Madrid en 2018, construye un robot esférico motorizado capaz de rodar de manera autónoma en misiones de exploración para entornos salvajes llenos de maleza por falta de cultivo. (European Robotics Week, 2018)

### **2.1.2. Vehículo aéreo no tripulado (UAV)**

Los vehículos aéreos no tripulados son aeronaves pilotadas remotamente, como su nombre lo indica vuelan sin tener un piloto a bordo, aunque este puede involucrarse en el control del vehículo,

también a esos vehículos se los conoce como drones (Otto, Agatz, Campbell, Golden, & Pesch, 2018).

Los vehículos tienen un sistema de control para el vuelo que incluyen estaciones terrestres, así también como la infraestructura para la comunicación, la terminal de datos, sistemas de lanzamiento y recuperación, además de la interfaz de control (Boucher, 2014).

“Las ventajas de un avión no tripulado es menor costo de fabricación, por lo que no se arriesgan vidas humanas y existe la posibilidad de adentrarse en sitios de difícil acceso” (Nadales, 2009).



**Figura 6.** Ejemplo de UAV de cuatro rotores

Fuente: (González, Herranz, & Calvo, *De los UAV a los RPAS*, 2014)

#### 2.1.2.1. Características Físicas

Los UAVs presentan características propias para su funcionamiento, es necesario conocer el sistema estructural de dichos vehículos para un correcto desempeño, existen normas de los materiales de construcción para las aeronaves, también depende del diseño de los mecanismos, los principios de vuelo, por ende, se analizarán estos parámetros a continuación:

- **Fuselaje:** es el cuerpo central donde se encuentra incorporado el resto de los componentes, el tamaño dependerá de la capacidad y la cantidad de componentes que requiera (Pappot & Boer, 2015).
- **Alas o brazos de soporte:** depende del tipo de equipo que vaya a realizar, es decir tiene que ser acorde a la categoría que cumplirá la función (Pappot & Boer, 2015).

- **Tren de aterrizaje:** es el componente donde se agrupa los amortiguadores, soportes, ruedas y otros equipos para realizar las tareas de aterrizaje y desplazamiento, es el encargado de absorber el impacto de la caída (Pappot & Boer, 2015).

### 2.1.2.2. Clasificación de los UAV

Existen diferentes criterios para clasificar a los UAV estos pueden ser las características de vuelo, capacidad de carga, tamaño, altura de vuelo, tipo de alas, etc. (Valavanis & Vachtsevanos, 2015) . En la Figura 7 se presenta la clasificación de los UAV de acuerdo al tipo de alas que ocupan para volar.



**Figura 7.** Clasificación de los UAV de acuerdo a su tipo de alas

Fuente: (Sánchez, 2017)

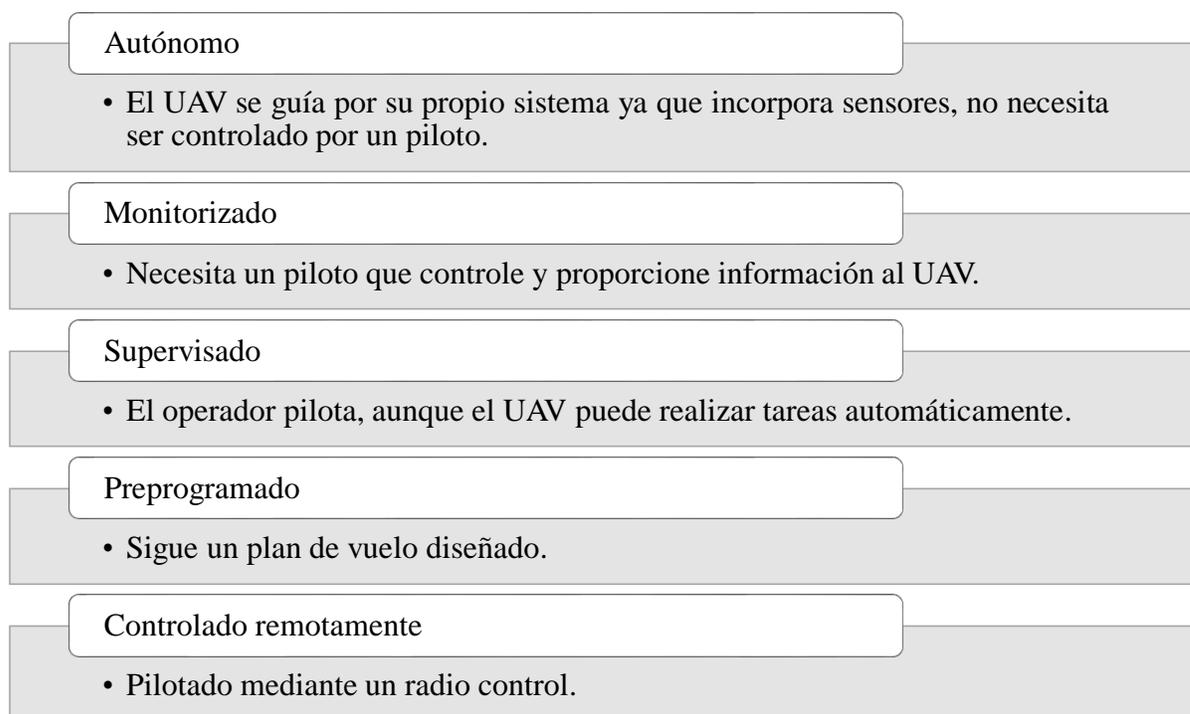
La OTAN (Organización del tratado del Atlántico Norte) mediante la normativa STANAG 4670 trata de normalizar la licencia para el manejo de los UAVs a través de su agencia de estandarización, en la Tabla 1 se da a conocer la clasificación vigente por esta entidad.

**Tabla 1***Clasificación de los UAV's mediante la norma STANAG 4670*

Clasificación UAV según la OTAN				
Clase	Categoría	Empleo	Altitud de operación	Radio de misión
Clase III > 600 kg	Combate	Estratégico	Hasta 65000 pies	Sin límite
	Hale (High altitude long endurance)	Estratégico	Hasta 65000 pies	Sin límite
	Male (Medium altitude long endurance)	Operacional	Hasta 45000 pies	Sin límite
Clase II ≤ 600 kg	Táctica	Táctico	Hasta 10000 pies	200 km
Clase I < 150 kg	Pequeño	Táctico	Hasta 1200 pies	50 km
	Mini	Táctico	Hasta 1000 pies	25 km
	Micro	Táctico	Hasta 200 pies	5 km

Fuente: (Barrientos, García, &amp; Silva, 2007)

Otra clasificación de los UAV's de acuerdo a su método de control, es decir como es pilotado el vehículo se presenta en la Figura 8.

**Figura 8.** Clasificación de los UAV mediante el sistema de control

Fuente: (Ruipérez &amp; Conejero, 2016)

### 2.1.2.3. Aplicaciones de los UAV

Existe una gran variedad de aplicaciones enfocadas a los vehículos aéreos no tripulados debido a su gran versatilidad, como son usos militares, supervisión, reconocimiento de desastres, investigación científica, mapeo y topografía, etc. Los usos más comunes de los UAV's se describen en la Figura 9.



**Figura 9.** Aplicaciones de los UAV

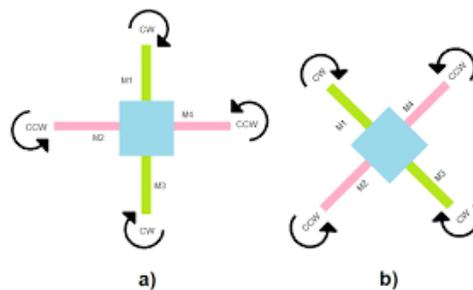
Fuente: (Nadales, 2009)

### 2.1.2.4. Cuadricópteros

Los cuadricópteros son los vehículos más comunes ya que su configuración es comercial en el área de los drones, maneja cuatro hélices impulsadas por cuatro motores, el ascenso y descenso se consigue mediante el aumento y reducción de las revoluciones del motor. Dado que la configuración de los motores va de dos en dos, es decir un par gira en sentido del reloj y el otro par en sentido contrario, esto permite que se pueda neutralizar la fuerza y de esta manera lograr el equilibrio del cuadricóptero (Ruipérez & Conejero, 2016).

### 2.1.2.5. Dinámica de los Cuadricópteros

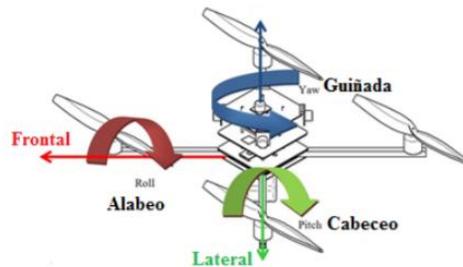
La dinámica de un cuadricóptero se obtiene a partir del momento de inercia de cada rotor, como su nombre lo indica manejan cuatro motores los cuales se encuentran ubicados a distancias equidistantes uno del otro, así también del centro de la estructura, esto permite que cada uno de los motores proporcione el empuje necesario para que el vehículo pueda elevarse y desplazarse en el aire, mismo que depende del sentido de giro de los motores. En Figura 10 se observan los tipos de configuraciones que puede adoptar un cuadricóptero. (Hernández, Pedraza, & Velázquez, 2014).



**Figura 10.** Configuración de rotors en:  
**a) Cruz (+) b) Equis (X)**

Fuente: (Hernández, Pedraza, & Velázquez, 2014)

Los cuadricópteros para ejercer su maniobrabilidad dependen de los ángulos de navegación, mismos que permiten hacer tres tipos de movimientos como se observa en la Figura 11, estos describen la orientación (Grijalva, Cueva, Ramírez, & Aguilar, 2019) del objeto en tres dimensiones y se conocen como: Roll (alabeo), Pitch (cabeceo) y Yaw (guiño) (González, 2017).



**Figura 11.** Ángulos de navegación de un cuadricóptero

Fuente: (Benito Carrasco, 2015)

- **Movimiento de Alabeo (Roll):** permite al cuadricóptero desplazarse de izquierda a derecha, el ángulo de giro es  $\varphi$ .
- **Movimiento de Cabeceo (Pitch):** permite al cuadricóptero realizar movimientos hacia adelante y atrás, el ángulo de giro es  $\theta$ .
- **Movimiento de Guiño (Yaw):** permite al cuadricóptero girar su eje vertical, el ángulo de giro es  $\psi$ .

### 2.1.2.6. Modelos comerciales

En el mercado existen variedad de soluciones en cuanto a cuadricópteros, los tres vehículos más utilizados por la oferta que brindan al usuario en sus características, pero todos con un mismo objetivo, se pueden observar en la Tabla 2.

**Tabla 2**  
*Modelos comerciales de cuadricópteros*

Características	Draganfly	Mikrokopter	Parrot Bebop
<b>Autonomía</b>	15 minutos	11 minutos	22 minutos
<b>Peso</b>	483 gr	980 gr	499 gr
<b>Sensores</b>	<ul style="list-style-type: none"> <li>• Acelerómetros</li> <li>• Giroscopio</li> <li>• Cámara de video</li> </ul>	<ul style="list-style-type: none"> <li>• Acelerómetros</li> <li>• Giroscopio</li> <li>• Hall 3D</li> <li>• Barómetro</li> </ul>	<ul style="list-style-type: none"> <li>• GPS</li> <li>• Glonass</li> <li>• Giroscopio</li> <li>• Cámara de video</li> </ul>
<b>Ejemplo</b>			

Fuente: (Nadales, 2009)

### **2.1.3. Robótica cooperativa**

La robótica cooperativa busca diseñar sistemas que permitan la ejecución de tareas empleando robots que trabajan coordinadamente aplicando el refrán “divide y vencerás”. En muchos casos enfrentar tareas a través del uso de enjambres de robots aportan mejores características de flexibilidad, eficiencia, robustez y simplicidad a la solución. Dependiendo de la complejidad de la tarea a resolver el sistema puede contener más o menos robots. (González Torres, 2015)

#### **2.1.3.1. Antecedentes**

El campo de la robótica tiene origen en la década de los 80 cuando investigadores comenzaron a desarrollar sistemas de robots múltiples (Uny Cao, Fukunaga, & Kahng, 1999), a principios del siglo XX se construye un grupo de robots con el objetivo de realizar la planificación de trayectorias en diferentes ambientes y que puedan ejecutar una tarea específica, pero aparecen ciertas fallas, una de estas es la pérdida de comunicación entre estos, por lo que tratan de solucionar proponiendo una nueva arquitectura que sea capaz de no colapsar en la comunicación de estos robots (Molina & Rodríguez, 2014). Más tarde el MIT adopta un concepto llamado “enjambre robótico” en el cual se considera que la conducta de los grupos animales se los puede aplicar a ciertas máquinas, desarrollando un grupo de robots que a diferencia del anterior que presentaba un control centralizado se lo cambió por un control distribuido, para no tener problemas en la comunicación y de esta manera resolver las tareas en conjunto (Campoverde, 2017). En la Figura 12 se presenta un ejemplo de enjambre robótico desarrollado para ayudar en tareas de rescate tras una catástrofe.



**Figura 12.** Robótica cooperativa

Fuente: (Almeida & Ochoa, 2013)

Un esquema de control robótico depende de las características de los animales ya que estos forman sociedades y agrupaciones para trascender sus capacidades individuales por lo que se pueden aumentar la eficiencia y ahorro de energía en la sincronización de movimientos individuales (Chantemargue, Courant, & Hirsbrunner, 2002).

#### **2.1.3.2. Desarrollo de la robótica cooperativa en la actualidad**

En la actualidad la robótica cooperativa es utilizada para realizar tareas complejas en las que el ser humano no puede intervenir, ya sea por el peligro o por espacios limitados, por ende, se vuelve más sencillo resolver un problema con un conjunto de robots capaces de solucionar dicho problema obteniendo una mayor exactitud y disminución en los costos (Uny Cao, Fukunaga, & Kahng, 1999).

La robótica cooperativa tiene una extensa línea de aplicaciones, pero las áreas de mayor desarrollo en la actualidad son la localización, mapeo y exploración de sitios, empleado algoritmos en un robot sencillo que luego se extienden a múltiples robots (López, Mata, & García, 2001). Los campos en los que han aplicado sistemas robóticos cooperativos se describen a continuación:

- **Medicina:** en el campo de la investigación para trasladar equipos como probetas.
- **Exploración:** la robótica cooperativa en este campo ayuda a la exploración de entornos desconocidos para evaluar peligros y de esta manera realizar las tareas.

- **Seguridad:** los robots cuentan con características que determinan su sistema de navegación mediante el uso de sensores.
- **Automoción:** permite realizar tareas repetitivas como el ensamblado de piezas.

### 2.1.3.3. Clasificación por nivel de comunicación de los sistemas robóticos

La comunicación en el rendimiento de grupos multi-robots permite que se puedan cumplir las tareas de manera eficiente, debido a que la información debe ser transmitida velozmente y sin errores, tomando en cuenta este criterio la clasificación de acuerdo al nivel de comunicación e interacción entre los robots que forman el sistema es:

- **Muy acoplados o colectivos:** Cada robot tiene la capacidad de enviar información de un robot a otro y así poder coordinar los movimientos entre estos, por lo que se necesita que sea una comunicación sincrónica y de esta manera los robots pueden realizar la tarea, ya que si uno falla el sistema no es apto para que funcione correctamente.
- **Poco acoplados o manadas:** Cada robot funciona independiente es decir toma la decisión propia de cada movimiento que va a realizar, no necesita un sistema central ya que este funciona como un sistema de control distribuido (Cautle, Berra, & Pérez, 2018).

## 2.2. Percepción visual

### 2.2.1. Captura de imágenes de la cámara de un UAV

Actualmente, la comunidad de fotogrametría y de percepción remota establece y promueve el uso de los vehículos aéreos no tripulados como una nueva y confiable alternativa de bajo costo para la adquisición de imágenes aéreas. Estos sistemas presentan una gran ventaja en comparación con otros sistemas aéreos tradicionales como son aviones y satélites, debido a su alta resolución

espacial y temporal, costos de adquisición y de operación, uso en ambientes peligrosos, ausencia de tripulación y respuesta rápida. (Escalante Torrado & Cáceres Jiménez, 2016)

### 2.2.1.1. Representación y formación de una imagen

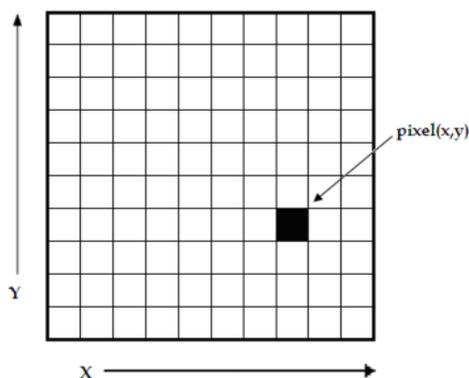
El ojo humano es un mecanismo que permite realizar el proceso de formación de la imagen, de modo análogo para formar una imagen digital capturada a través de una cámara, se debe detectar la luz del objeto tras pasar por varios filtros hasta llegar al sensor de imagen, esta luz incide y genera una pequeña señal eléctrica en cada receptor, transformando esta señal en datos digitales, que finalmente se convierten en dígitos binarios o conocidos como píxeles. (Quisaguano, 2018),



**Figura 13.** Proceso de formación de una imagen digital.

Fuente: (Gualdrón, Duque, & Chacón, 2013)

Al obtener la imagen en un plano 2D se la puede definir como una función bidimensional  $f(x, y)$ , donde  $x$  y  $y$  son coordenadas espaciales de la imagen como se muestra en la Figura 14 de tal manera que esta función representa la intensidad de la imagen en ese punto (Alvarez, 2018) misma que está dada por  $I = f(x, y)$ .



**Figura 14.** Imagen en representación de píxeles

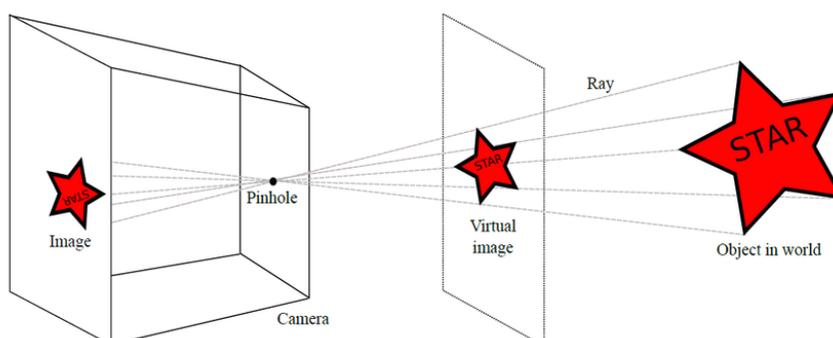
Fuente: (Lyra, Ploussi, & Georgantzoglou, 2011)

La digitalización de una imagen se representa mediante una matriz, de  $M \times N$  los cuales equivalen a la cuadrícula de los píxeles de la imagen como se observa en la Figura 14, la matriz es expresada en la forma de la ecuación (1).

$$\begin{bmatrix} f(1, N) & \cdots & f(M, N) \\ \vdots & \ddots & \vdots \\ f(1, 1) & \cdots & f(M, 1) \end{bmatrix} \quad (1)$$

### 2.2.1.2. Modelo PIN-Hole de la cámara

El modelo de la cámara pinhole es un modelo de proyección que describe la relación que existe entre una escena del mundo en 3D y la proyección en el plano 2D de la imagen, donde la apertura de la cámara se la describe mediante un punto (Scaramuzza & Fraundorfer, 2011).



**Figura 15.** Modelo de la cámara pinhole

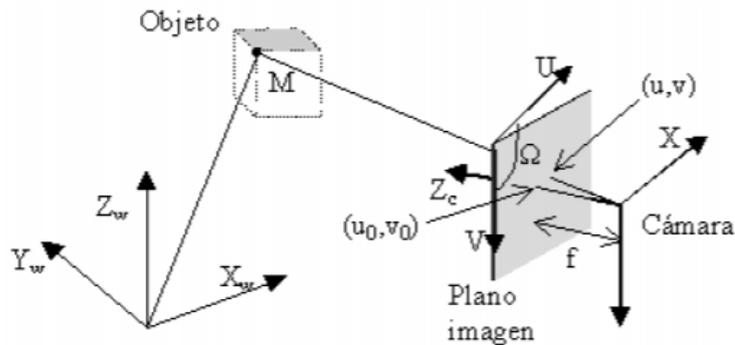
Fuente: (Fuertes, Lucena, Pérez, & Valdivia, 2016)

En la Figura 15 se muestra el modelo pinhole, donde el cubo representa la cámara que posee un pequeño agujero que es el centro de la proyección, de esta manera se visualiza en la pantalla una imagen que será invertida (Hannuksela, Barnard, Sangi, & Heikkila, 2011). La proyección se determina a través de la transformación de la imagen, que viene dada por la ecuación (2).

$$m = P \cdot M \quad (2)$$

Donde  $M$  representa el punto en el espacio de coordenadas en el mundo 3D, mientras que  $P$  es la matriz de proyección y  $m$  es el vector del punto proyectado en la imagen transformada en píxeles, como se observa en la Figura 16 .

$$M = \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad m = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (3)$$



**Figura 16.** Proyección de puntos en la imagen

Fuente: (Iser, 2003)

Los pasos para convertir una escena de coordenadas 3D en 2D son los siguientes:

- Las coordenadas 3D se debe cambiar a un plano de referencia 3D de la cámara mediante traslación y rotación de puntos.
- Proyección sobre el plano de imagen mediante el método pinhole.

- Modelo de los parámetros de distorsión del lente de la cámara.
- Conversión de unidades de milímetros a píxeles.

### 2.2.2. Métodos de caracterización de una imagen

Las imágenes se representan por medio de una matriz de dos dimensiones, donde cada pixel tiene un valor de 8 bits es decir de 0 a 255 lo que representa la escala de grises. Mientras que la imagen de color se usa más de un canal para representar cada píxel, el que generalmente se usa es el RGB donde se atribuyen los colores rojo, verde y azul el cual también se representa en una escala de 8 bits (Nixon & Aguado, 2012). Los métodos para representar imágenes generalmente son:

- **Características puntuales:** La representación de las características puntuales depende del nivel de gris de entrada en un punto. Las operaciones de dichas características se basan en la binarización, la umbralización ya que permite la manipulación de los píxeles.
- **Características globales:** La representación de las características globales permite representar a la imagen mediante un vector multidimensional que describe la información de la imagen. El vector se conforma de características como la forma, la textura y el color de la imagen.
- **Características locales:** La representación de las características locales permite representar a la imagen en función de regiones es decir ubicando características de iluminación y el punto de vista de la cámara. La imagen se representa en función de un conjunto de descriptores de características extraídos mediante regiones de interés (Ismail & Hassaballah, 2015).

#### 2.2.2.1. Proceso de caracterización de una imagen

Para obtener la caracterización de una imagen se debe pasar por tres etapas (Aguilar & Angulo, Real-time video stabilization without phantom movements for micro aerial vehicles, 2014), (Aguilar & Angulo, Robust video stabilization based on motion intention for low-cost micro aerial

vehicles, 2014), (Aguilar, Angulo, & Pardo, Motion intention optimization for multicopter robust video stabilization, 2017): detección de puntos de interés, descripción de los mismos y puesta en correspondencia, para esto se debe comparar dos imágenes que ya han sido digitalizadas, para obtener los vectores descriptores de la imagen 1 y así comparar los puntos característicos de la imagen 2.

### **2.2.3. Percepción visual**

La percepción visual es un tema de gran importancia en el campo de la robótica, de la misma manera que los seres humanos utilizan el sentido de la vista en actividades diarias y comunes los robots necesitan conocer su entorno para completar actividades propuestas por su programador.

Un documento publicado por IEEE en el tema de Robótica y Automatización (Giusti, y otros, 2016) enfoca la percepción visual para obtener la dirección de una ruta de senderos de bosque o montaña partiendo de una imagen monocular; el robot al tiempo que se traslada en el sendero va adquiriendo la imagen desde el punto de vista de su ubicación.

El tema del senderismo no es la única aplicación en la que se puede utilizar la percepción visual, en realidad es una temática amplia con mucha investigación aplicada en entornos complejos como la fabricación, servicios de seguridad, peligro nuclear y muchos más (Shan, Li, Zhang, Wang, & Fang, 2018). Buscando como idea principal proporcionar a los robots un método para moverse dentro de un entorno ya sea controlado o no, mediante una o varias cámaras evitando colisiones.

## **2.3. Navegación servovisual**

### **2.3.1. Visión artificial**

Cuando se utiliza este término se hace referencia a buscar un equivalente simbólico en lenguaje de computadora sobre imágenes del mundo que nos rodea. Tratando que cierto evento sea capturado y procesado para utilizar la información relevante de la imagen.

Los seres humanos tratan de comprender que nuestro entorno mediante nuestros ojos, de la misma forma la inteligencia artificial proporciona un método para que los robots conozcan el entorno y cumplan con la aplicación para la que fueron diseñados.

La Pontificia Universidad Católica del Perú desarrollo un “SISTEMA DE VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO Y MANIPULACIÓN DE OBJETOS UTILIZANDO UN BRAZO ROBOT” (Sobrado Malpartida & Tafur Sotelo, 2006), lo que se busca es realizar una tarea de “pick and place” identificando piezas colocadas en una mesa, para posteriormente encontrar su ubicación. En donde se parte de una imagen para obtener la posición y orientación en el plano de la imagen y mediante el procesamiento adecuado obtener la localización en el espacio de trabajo del robot.

La manipulación de objetos con visión artificial no solo se enfoca en la investigación si no principalmente se busca aplicaciones como por ejemplo en el ensamble de piezas, asistentes robóticos médico, soldadura, pintura, carga de productos pesados, entre otros. Buscando que el ser humano realice menos esfuerzo físico y concentre su inteligencia en el desarrollo y la investigación de nuevas tecnologías.

### **2.3.2. Odometría visual**

La odometría visual permite estimar el movimiento y posición de un robot móvil a partir de las imágenes capturadas por una o más cámaras. Si se comparan las imágenes obtenidas en un instante  $t_1$  respecto a otras imágenes en un instante  $t_2$ , se puede calcular la distancia existente en un punto determinado de la imagen, logrando evidenciar que el robot ha avanzado desde su posición inicial. En sistemas de una sola cámara para determinar distancias precisas es necesario que la posición de la misma esté orientada en un ángulo determinado y a una altura conocida, ya que a partir de la tangente se puede calcular la distancia. Esta tecnología no solo es utilizada en gran número de aplicaciones con robots móviles, sino que también en androides y drones. Según el estudio de (Delgado Hernández, 2016) la odometría visual cuenta con los siguientes inconvenientes que deben ser tomados en cuenta en un sistema.

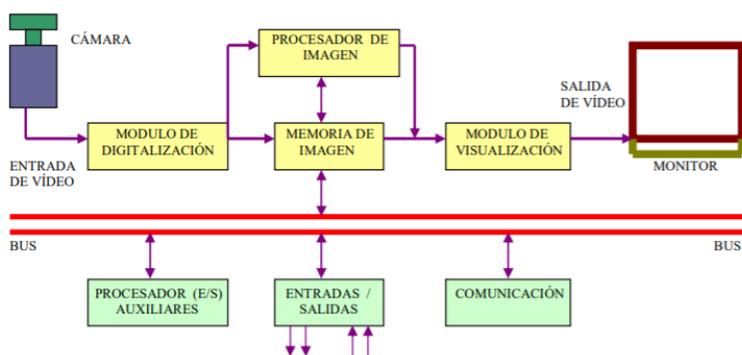
- Las cámaras: Para funcionar correctamente un sistema, es necesario que las cámaras estén calibradas.
- El entorno: Las cámaras son muy sensibles a los cambios bruscos de las condiciones del entorno como la luminosidad. Otro aspecto son las escenas dinámicas, por ejemplo, objetos que aparecen y desaparecen de la escena.

### **2.3.3. Arquitectura de un sistema de visión.**

La investigación de (Gómez, 2011) establece que la arquitectura básica para un sistema de visión por computador similar al de la Figura 17, generalmente posee 5 elementos principales los cuales se describen a continuación.

- Dispositivo de captura: Dispositivo físico que es sensible a una determinada banda del espectro electromagnético, produce una señal eléctrica proporcional al nivel de energía.

- Conversión A/D: Convierte la señal obtenida del dispositivo de captura en una señal digital.
- Memoria de video: Memoria de tipo RAM en la que se almacena la imagen digitalizada.
- Procesador: La memoria de video se acopla a un procesador de propósito general que permite operar sobre la imagen.
- Monitor: Generalmente se utiliza un monitor para visualizar las imágenes adquiridas.



*Figura 17.* Diagrama de bloques de un sistema de visión artificial.

### 2.3.4. Transformaciones geométricas

Una transformación geométrica se define como la relación de los puntos en dos imágenes, se representan como operaciones matriciales sobre los puntos del objeto y cada uno se representa como una matriz de coordenadas  $(x,y,z)$ . Todos los algoritmos que realizan transformaciones geométricas se basan en realizar una nueva distribución de los píxeles según lo que se requiera. El proceso de transformación determina las nuevas coordenadas de un píxel  $(i,j)$  en la rejilla transformada  $(i',j')$ , normalmente se utiliza el método de interpolación bilineal para calcular las coordenadas en la rejilla de destino.

#### 2.3.4.1. Traslación

Consiste en cambiar cada píxel por el correspondiente a sus coordenadas más el desplazamiento requerido en cada dirección  $t_x$  y  $t_y$ . De este modo para desplazar vertical o horizontalmente una

imagen las coordenadas en la malla de destino serían  $(i + t_x, j + t_y)$ , se puede escribir de forma general una matriz de transformación como la ecuación (4).

$$M = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \quad (4)$$

A continuación, en la Figura 18 se observa un ejemplo al aplicar una translación de posición de una imagen.



**Figura 18.** Traslación de una imagen.

Fuente: (Unipython, 2019)

#### 2.3.4.2. Rotación

Los algoritmos de rotación de imagen son complejos debido al tiempo de procesado. Dado un punto  $(i, j)$  y se rota un ángulo  $\theta$  las coordenadas  $(i', j')$  del nuevo punto se obtienen mediante la ecuación (5). (González Marcos & Pisón Ascacíbar, 2006)

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} i \\ j \end{bmatrix} \quad (5)$$

A continuación, en la Figura 19 se observa un ejemplo al rotar un ángulo  $\theta$  una imagen.



**Figura 19.** Rotación de una imagen.

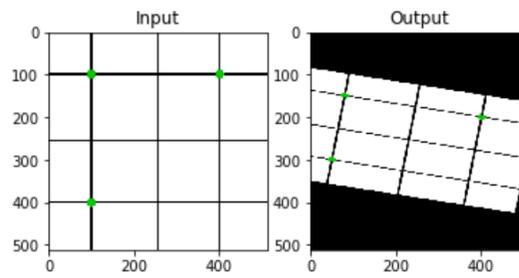
Fuente: (Unipython, 2019)

### 2.3.4.3. Transformación Afín

Una transformación afín se basa en una transformación lineal no singular. Su forma matricial se presenta a continuación en la ecuación (6).

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (6)$$

A continuación, en la Figura 20 se observa un ejemplo al aplicar la matriz de transformación afín una imagen.



**Figura 20.** Ejemplo de transformación afín.

Fuente: (Unipython, 2019)

### 2.3.5. Navegación robótica

La navegación robótica es la habilidad que permite a un robot móvil localizarse en un entorno con obstáculos y guiar el curso de su trayectoria. Existen diferentes metodologías de su aplicación, pero todas tienen el objetivo de llevar de forma segura el vehículo a su destino. Para un robot tener una abstracción del mundo que lo rodea es necesario de sensores como ultrasónicos, láser, cámaras entre otros. Realizar una tarea de navegación para un robot móvil implica recorrer un camino que lo conduzca desde una posición inicial hasta otra final, pasando por ciertas posiciones intermedias o submetas. El problema de la navegación de un robot móvil con sensores de imagen (Tenezaca, Canchignia, Aguilar, & Mendoza, 2019) se divide en las siguientes etapas:

- **Percepción del mundo:** Como se mencionó anteriormente uno de los sensores para que un robot pueda tener una abstracción del mundo que lo rodea son las cámaras mismas que proveen de imágenes al sistema permitiendo abstraer rasgos específicos del medio. (Aguilar, Manosalvas, Guillén, & Collaguazo, 2018), (Aguilar & Morales, 3D Environment Mapping Using the Kinect V2 and Path Planning Based on RRT Algorithms, 2016).
- **Localización:** Permite conocer la posición donde se encuentra el robot en el entorno que lo rodea, se divide en dos grupos principales basada en puntos de referencia y basada en el análisis de imágenes. Los puntos de referencia pueden ser artificiales (líneas, flechas, marcadores o códigos QR) o naturales (puertas, esquinas, senderos, etc.). (Aguilar, Manosalvas, Guillén, & Collaguazo, 2018), (Aguilar, Castro, Caballeros, & Segarra, 2018), (Delgado Hernández, 2016).
- **Planificación de la ruta:** Se crea una secuencia ordenada de objetivos o submetas que deben ser alcanzadas por el vehículo, esta secuencia se calcula utilizando el modelo o mapa del entorno. (Aguilar, Morales, Ruiz, & Abad, RRT\* GL Based Optimal Path Planning for Real-Time Navigation of UAVs, 2017), (Aguilar, Caballeros, Segarra, & Castro, 2018), (Salgado, Tierra, & Aguilar, Travel Planning in Public Transport Networks Applying the Algorithm A\* for Metropolitan District of Quito, 2017), (Aguilar, Abad, Ruiz, Aguilar, & Aguilar-Castillo, 2017), (Aguilar, Morales, Ruiz, & Abad, RRT\* GL Based Path Planning for Virtual Aerial Navigation, 2017).
- **Seguimiento del camino:** Se efectúa el desplazamiento del robot, según el camino generado mediante el adecuado control de los actuadores del vehículo. (Aguilar W. G., Salcedo, Sandoval, & Cobeña, 2017), (Aguilar, Angulo, & Costa-Castello, Autonomous Navigation Control for Quadrotors in Trajectories Tracking, 2017), (Aguilar & Angulo, Real-Time Model-Based Video

Stabilization for Microaerial Vehicles, 2016), (Grijalva & Aguilar, Landmark-Based Virtual Path Estimation for Assisted UAV FPV Tele-Operation with Augmented Reality, 2019).

### **2.3.6. Detección de obstáculos**

La planificación de trayectorias para robots móviles es una función principal en operaciones de desplazamiento autónomo de vehículos, el problema es encontrar una trayectoria libre de colisiones entre una posición inicial y otra final en un entorno cerrado. Los obstáculos (Aguilar W. G., y otros, Pedestrian Detection for UAVs Using Cascade Classifiers and Saliency Maps, 2017), (Aguilar W. G., y otros, Pedestrian Detection for UAVs Using Cascade Classifiers with Meanshift, 2017), (Aguilar W. G., y otros, 2017), (Aguilar W. G., y otros, 2017), (Aguilar W. G., y otros, 2017) se pueden presentar en un entorno de forma total o parcial, existen sensores comúnmente utilizados para evitar objetos en un entorno que pueden ser ultrasónicos, telémetro o cámara de video, (Bambino, 2008) mismos que permiten tener un mayor grado de autonomía del vehículo y ayudándole en la toma de decisiones que dependen del medio físico, el problema es que son susceptibles a los cambios de iluminación y requieren que los algoritmos sean ajustados de acuerdo a las condiciones del medio.

#### **2.3.6.1. Evasión de obstáculos basada en color.**

El uso del color en visión artificial es comúnmente utilizado ya que es un potente descriptor que a menudo simplifica la identificación y extracción de objetos de una escena. El reconocimiento de color no solo se puede utilizar para identificación objetos del medio en el que se desenvuelve un robot sino también para determinar la posición y orientación del mismo en su entorno, con frecuencia la localización de plataformas móviles se realizan con visión bidimensional, donde el

objetivo es destacar partes de la imagen que correspondan a objetos de interés del resto de la superficie, este proceso se denomina segmentación. (Alejando & Vicente, 2009)

#### **2.3.6.2. Evasión de obstáculos basada en forma.**

El borde de una imagen ayuda en las tareas de reconocimiento de objetos, en planificación de trayectorias para robótica móvil la frontera extraída de una imagen no solo sirve para evadir el obstáculo presente en el entorno sino también permite que el robot reconozca que objeto es. Los algoritmos recurrentes en la extracción de bordes de una imagen son los operadores Canny y Deriche, el problema con de estos métodos es que son sensibles al ruido y requieren un alto costo computacional. (Aristondo Etxeberria, 2010), (Aguilar, Álvarez, GrijalvaI, & Rojas, 2019).

### **2.4. Guiado de sistema autónomos**

#### **2.4.1. Planificación de ruta para cobertura de superficies**

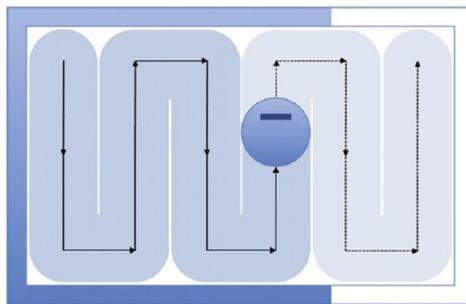
La planificación de trayectorias para cobertura de superficies es necesaria en varias áreas de aplicación, tales como limpieza de pisos, corte de césped, desminado, pintura y búsqueda de objetos. (Oksanen & Visala, 2009).

En la investigación de (Liang Cao, Huang, & Hall, 1988) se definen los criterios para cobertura de regiones con robots móviles, los cuales se describen a continuación:

- El robot debe moverse por toda el área (cubrir toda la superficie).
- El robot debe llenar la región sin superponer la ruta.
- Se requiere una operación continua y secuencial sin ninguna repetición de rutas.
- El robot debe evitar todos los obstáculos.
- Trayectorias de movimiento simples como líneas rectas o círculos para simplificar el control.
- Se desea una ruta optima en las condiciones disponibles.

### 2.4.1.1. Algoritmos de cobertura de superficies

Uno de los métodos más utilizados por la mayoría de algoritmos de cobertura de superficies es la descomposición celular la cual divide el espacio libre (espacio libre de obstáculos) en regiones simples y no superpuestas llamadas células, la unión de todas ellas cubren exactamente el espacio libre. Las superficies de cubrir con mayor facilidad son las que no presentan obstáculos ya que se puede recurrir al patrón en zigzag o también llamado “cortar césped” mostrado en la Figura 21.

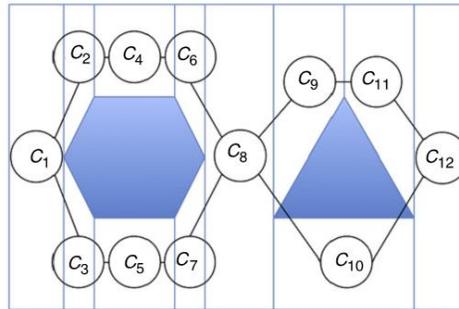


**Figura 21.** Típico camino en zigzag

Fuente: (Galceran & Carreras, 2013)

Los algoritmos basados en descomposición celular generan una ruta de cobertura en dos pasos, primero descomponen el espacio libre en celdas y almacenan la descomposición como un grafo de adyacencia, luego se calcula el recorrido a través de las células, como una secuencia que visita cada nodo. Dos de los algoritmos base para descomposición celular se describen a continuación.

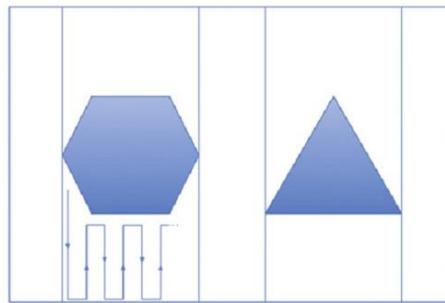
- **Descomposición trapezoidal:** Este método ofrece una planificación de trayectoria para cobertura de superficies completa, cada celda de esta descomposición es un trapecioide como se muestra en la Figura 22, lo que favorece a poder utilizar movimientos simples de ida y vuelta para cubrir cada celda y luego mediante métodos de caminata exhaustiva se determina el orden de visita para cada celda.



**Figura 22.** Descomposición trapezoidal de superficies.

Fuente: (Galceran & Carreras, 2013)

- **Descomposición de Boustrophedon:** Este método es similar a la descomposición trapezoidal, pero considera solo vértices donde el segmento vertical puede extenderse tanto para arriba como para abajo del vértice sin topar con parte del obstáculo como se observa en la Figura 23. Este algoritmo reduce el número de celular en comparación al trapezoidal por tanto se obtienen rutas de cobertura más cortas.



**Figura 23.** Descomposición de boustrophedon.

Fuente: (Galceran & Carreras, 2013)

#### 2.4.2. Métodos de control visual

Debido a la alta precisión alcanzada por los sistemas de control visual actualmente son cada vez más utilizados y desarrollados, han sido implementados en tareas como guiado de vehículos, control de manipuladores robóticos, medicina, aplicaciones espaciales y militares, recolección de frutas, ensamblado de estructuras, entre otras (García Gómez, Pomares Baeza, & Torres Medina,

2010). En el trabajo de (Guerrero Virramontes, Becerra Fermín, & Torres Muñoz, 2014) se propone una clasificación para los sistemas de control visual de acuerdo a la configuración de la cámara, las cuales se presentan a continuación:

- **Configuración eye-in-hand:** la cámara se monta sobre el efector final del robot. Existe una relación constante entre la posición de la cámara y la posición del efector final del robot.
- **Configuración eye-to-hand:** la cámara se fija en algún lugar en el espacio de trabajo, permitiendo que la imagen sea independiente del movimiento del robot.

Otra clasificación de los sistemas de control visual se basa en la forma en la que se utiliza la información extraída de las imágenes para alcanzar el objetivo, las cuales son las siguientes:

- **Control visual basado en posición:** En este método la posición del robot se recupera con respecto al ambiente de trabajo o con respecto al objetivo.
- **Control visual basado en imagen:** A diferencia del anterior en este método se extrae la posición del robot utilizando la información 2D extraída de la imagen sin necesidad de obtener la posición del robot en el espacio de trabajo.

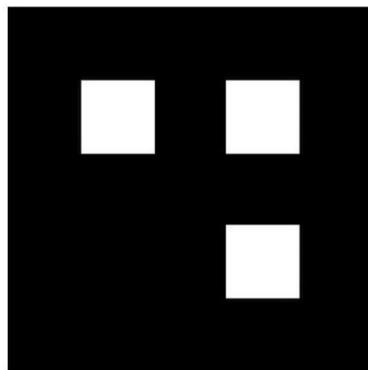
### 2.4.3. Sistemas de detección de minas antipersonales

El estudio de (MacDonald, y otros, 2003) establece que los equipos disponibles para detección de minas antipersonales en la actualidad no han cambiado en gran medida a los utilizados en la Segunda Guerra Mundial, donde un operador estaba equipado con un detector de metales de mano y una varilla con punta, el equipo primero limpiaba el área de vegetación para luego dividirse en carriles de 1 metro de ancho, en cada carril era barrido lentamente el detector de metales balanceándolo de tal manera que cubra la superficie en su totalidad, cuando el detector señalaba la presencia de una anomalía, un segundo desminador verificaba la zona en cuestión.

Las modificaciones que han sufrido estos sistemas de detección de minas antipersonales no han cambiado en su método sino en seguridad y tecnología para el operador, ya que se ha implementado prototipos robóticos que realizan la misma tarea de hace muchos años pero con precisión y ante todo lo más importante salvaguardando la vida de los desminadores, Un claro ejemplo de este avance es el proyecto realizado en la Universidad Complutense de Madrid donde se propuso un “Sistema de exploración de terrenos con robots móviles: aplicación en tareas de detección y localización de minas antipersonales.” (Ponticelli Lima & Gonzales de Santos, 2011)

#### **2.4.4. Marcadores ARUCO**

Los marcadores aruco están compuestos por un borde negro externo y una región interna codificada en un patrón binario, el cual es único y sirve para identificar a cada marcador. Dependiendo del diccionario existen marcadores con más o menos número de bits, sin embargo, más bits requiere una resolución mucho mayor para una detección correcta. En la Figura 24 se puede observar el ejemplo de un marcador Aruco con rejilla de 5x5 bits.



*Figura 24.* Ejemplo de marcador Aruco de rejilla de 5x5 bits.

## CAPÍTULO III

### GENERALIDADES DEL SISTEMA

#### 3.1. Hardware del sistema

El hardware del proyecto está constituido por 3 componentes principales, los cuales son vehículo aéreo no tripulado, plataforma móvil terrestre y estación en tierra, mismos que estarán comunicados inalámbricamente entre sí, cada uno de los componentes se describen a continuación:

##### 3.1.1. Vehículo aéreo no tripulado Parrot bebop2

El vehículo aéreo no tripulado escogido para realizar este proyecto es el Parrot Bebop 2 el cual se observa en la Figura 25, este cuenta con un peso ultraligero de 500 gramos y autonomía de 25 minutos, las prestaciones que brinda son volar, filmar y tomar fotos al mismo tiempo, se puede manejar tanto en escenarios exteriores como interiores, además tiene incorporado un sistema de estabilización de 3 ejes. Se eligió este dispositivo de por su versatilidad de manejo y su bajo costo de adquisición en comparación a otros UAV, este posee una cámara con ángulo de captura de 180° facilitando la toma de imágenes de forma cenital, que es un requerimiento indispensable para el proyecto.



**Figura 25.** Parrot Bebop 2

Fuente: (Parrot Fanpage Oficial)

### 3.1.1.1. Estructura Física

Esta versión del UAV Parrot Bebop 2 tiene ausencia de defensas en los laterales en comparación de su primera versión, su estructura está fabricada en PA12 reforzada con fibra de vidrio (20 %) y Grilamid que es un material ligero, pero le permite ser muy resistente, su tren de aterrizaje está formado por 4 patas de apoyo de goma, amortiguadores y refuerzos en ciertos lugares. Está provisto de varios sensores que ayudan en su desempeño y suministran información al operador, a continuación, se describen las funciones de cada uno de ellos.

- Sensor ultrasonidos: mide la altura del suelo en rango de 5 metros.
- Giroscopio: determina el ángulo para conocer la ubicación del dron.
- Acelerómetro: determina la posición y velocidad lineal de vuelo.
- Brújula digital: permite posicionar al dron en el espacio.

### 3.1.1.2. Conectividad

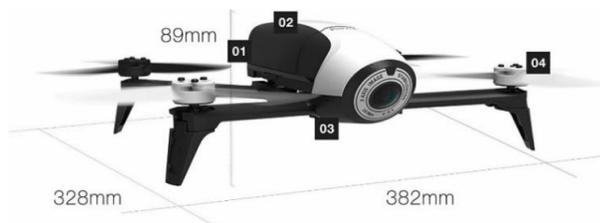
El dron se comunica mediante WiFi con estándar 802.11 a/b/g/n/ac, el cual tiene un alcance de hasta 300 metros, cuenta con una antena wifi de dos bandas MIMO, estas antenas son de tipo dipolo de 2.4 y 5 GHz, la potencia de emisión de señal es de hasta 21 dBm.

Este vehículo está diseñado para ser pilotado mediante un dispositivo celular con la aplicación *Freeflight 3* o el control de mando *Skycontroller* que aumenta su rango de acción a 2Km, sin embargo, maneja un SDK (Software Development Kit) abierto para desarrolladores el cual es compatible con el sistema operativo Linux, su alcance dependerá del tipo de antena que utilice la estación en tierra.

### 3.1.1.3. Capacidad y dimensiones

Sus dimensiones son de 328 x 328 x 89 mm para mayor detalle de sus medidas se puede observar la Figura 26, este tamaño no interfiere en la autonomía, pero sí mejora el comportamiento del

multirrotores en el aire, siendo mucho más estable en los vuelos exteriores con vientos moderados. La velocidad máxima en el eje horizontal es de  $16 \text{ m/s}$  mientras que la velocidad máxima para el ascenso es de  $6 \text{ m/s}$ .

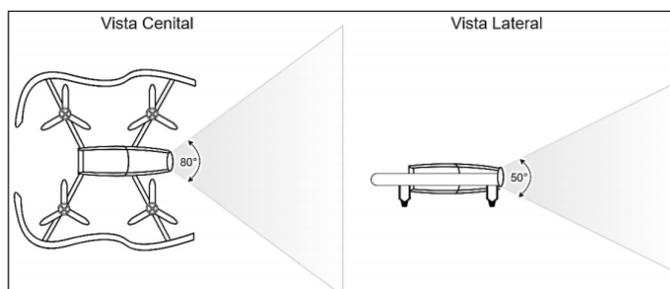


**Figura 26.** Dimensiones del Bebop 2

Fuente: (Parrot Fanpage Oficial)

#### 3.1.1.4. Dispositivo de visión del UAV

El Bebop 2 tiene una cámara Full HD, con 14mp para las fotografías y mientras que la resolución de video es de 1920x1080p. También, tiene un ojo de pez que suministra imágenes con mayor luz y nitidez en condiciones de baja luminosidad. El ángulo de visión de la cámara del dron (Ver Figura 27) para el eje vertical se encuentra en  $50^\circ$  mientras que para el eje horizontal es de  $80^\circ$ , dependiendo de la distancia de la cámara al objetivo varía el rango de visión. El video es devuelto sin distorsiones gracias a los algoritmos que maneja el dron sus imágenes en 2D son estables. (Aguilar & Salcedo, 2018)

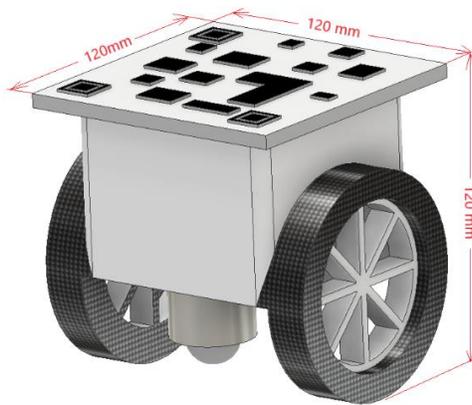


**Figura 27.** Ángulos de visión de la cámara del Bebop 2.

Fuente: (Aguilar & Salcedo, 2018)

### 3.1.2. Plataforma móvil

La plataforma móvil requerida deberá poder manejarse en todas las direcciones incluyendo el giro en su propio eje en sentido horario y antihorario, las dimensiones máximas de la misma para que pueda ingresar en lugares pequeños se ha fijado 12 x 12 x 12 cm (Ver Figura 28) , deberá poseer un marcador de identificación en la parte superior para poder localizarla en todo momento, además debe contar con sensores que detecten la presencia de metales en la superficie recorrida, su comunicación con la estación en tierra será inalámbrica. Cabe recalcar que las pruebas a realizar serán solo en escenarios controlados con superficies planas por lo que no es necesario que este sea para todo terreno.



*Figura 28.* Ejemplo de dimensiones de la plataforma móvil

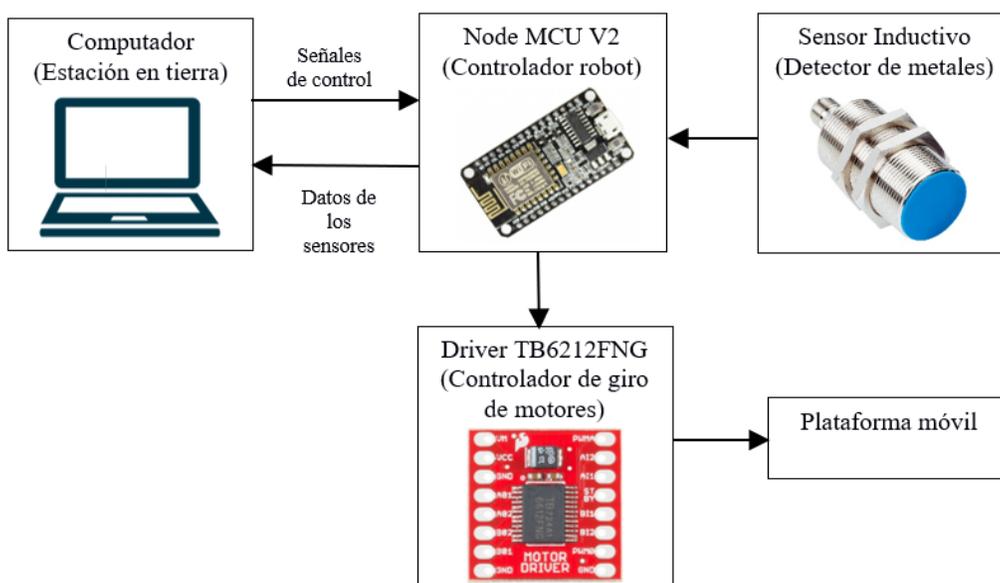
#### 3.1.2.1. Requerimientos y restricciones del robot móvil

- **Requerimientos:**
  - Tipo de terreno: plano sin crestas ni valles.
  - Robot pequeño y ligero
  - Robot con capacidad de giro en su propio eje.
  - Capacidad de controlar la velocidad en cada una de las llantas
  - Torque del robot: grande.

- Autonomía energética.
- **Restricciones:**
  - Velocidad máxima de desplazamiento: 0.2 m/s
  - Largo y ancho máximo del robot: 12cm x12cm
  - Peso: menor a 30kg

### 3.1.2.2. Requerimientos de implementación del robot móvil

Para llevar a cabo la construcción del robot móvil, se ha dividido su funcionamiento en un diagrama de bloques, los cuales se pueden observar en la Figura 29. Cada uno de los mismo se describen a continuación.



**Figura 29.** Diagrama de bloques del robot móvil.

#### Node MCU V2 Amica

Con la finalidad de controlar y comunicar al robot móvil con la estación en tierra se utilizó la tarjeta de desarrollo Node MCU V2 de Amica, la cual cuenta con un chip ESP8266 para la conectividad Wi-Fi, 16 pines GPIO, 64KiB de RAM para instrucciones, 96KiB de RAM para datos y

trabaja bajo el estándar IEEE 802.11 b/g/n Wi-Fi, que la hacen óptima para la comunicación con la estación en tierra y con sus pines digitales brinda la posibilidad de controlar el robot y a su vez verificar el estado de los sensores metálicos.

### **Controlador de motor dual TB6612FNG**

El driver TB6612FNG es usado para poder controlar la velocidad y el sentido de giro de los motores a una corriente de salida de 1A por cada canal, enviando un determinado valor de PWM para cada motor se puede controlar la velocidad del robot y también la dirección en la que se mueve. En la Tabla 3 se puede observar las principales características del módulo.

**Tabla 3**

*Características técnicas del módulo TB6612FNG.*

<b>Descripción</b>	<b>Característica</b>
Voltaje para los motores	4-13.5Vcc
Voltaje de alimentación	2.7-5.5Vcc
Corriente máxima de salida por canal	1A
Frecuencia máxima de PWM	100khz
Dimensiones	20.32 x 20.32mm

Fuente: (GeekFactory, 2019)

### **Sensor Inductivo interruptor de proximidad LJ8A3-2 Z/BX**

Los sensores inductivos tipo interruptor de proximidad son utilizados para detectar si existe un objeto metálico en el suelo en el transcurso del recorrido realizado por el robot. Se escogieron este tipo de sensores debido a que tienen salidas tipo interruptor es decir ON/OFF, razón por la cual se acoplan fácilmente con las entradas digitales de la placa de control NODE MCU V2, no es posible la utilización de sensores con salida de tipo analógica debido a que la tarjeta posee una sola entrada para lectura de los mismos y se requiere de 4 sensores acoplados al robot. En la Tabla 4 se describen las características principales de los sensores de inductivos de proximidad.

**Tabla 4***Características de los sensores de proximidad inductivos LJ8A3-2 Z/BX*

Descripción	Característica
Salida	Transductor de conmutación
Rango de detección	1.5 -2mm
Tensión de entrada	6-36Vcc
Salida de control	NPN
Objetos de detección	Metales Ferrosos
Grado IP	IP67
Corriente de salida	200 mA
Diámetro de la forma	8mm

Fuente: (Electronics Katrangi Trading, 2019)

**Motores de transmisión**

Para poder comenzar con el diseño de la plataforma es necesario contar las medidas de los motores que se acoplarán al chasis del robot, cumpliendo con los requerimientos de torque elevado y velocidad baja del robot se utilizaron motores de corriente continua de 12VDC con caja reductora, con un torque máximo de 750[oz/in] y dado que no se requiere un control preciso de posicionamiento, este motor satisface las necesidades de movilidad y tracción en las ruedas del robot, en la Tabla 5 se describen las características de los motores.

**Tabla 5***Características de los motores de transmisión.*

Descripción	Característica
Rango de operación	6-12Vcc
Torque máximo	750 oz/in
Corriente máxima	1 A
Corriente sin carga	80 mA
Velocidad sin carga	10RPM
Relación de transmisión	300:1
Tamaño del motor	1.355”Dia x 1.70” L

CONTINÚA



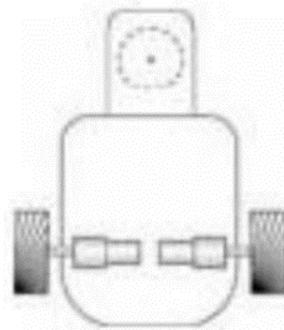
Tamaño de la caja reductora	1.45" Dia x 1.30" L
Tamaño del eje	0.23 " Dia x 0.715"L
Peso	0.504 Lbs

Fuente: (ServoCity, 2019)

### 3.1.2.3. Propuestas de la configuración cinemática del robot móvil

La elección de la configuración cinemática que llevará el robot es la parte fundamental para el diseño de la plataforma móvil, teniendo en cuenta los requerimientos y restricciones anteriormente estipulados que necesita el robot se determinaron dos posibles configuraciones que cumplen con lo requerido, a continuación, se detallan las mismas:

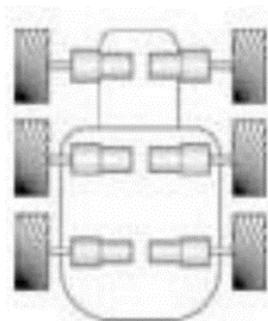
- Configuración Diferencial: Se caracteriza por ser uno de los modelos más sencillos. Este tipo de configuraciones no cuentan con ruedas directrices, solamente dos ruedas opuestas en un eje perpendicular a la dirección del robot.



**Figura 30.** Esquema de configuración diferencial

Fuente: (Jiménez, 2017)

- Configuración Skid-Steer: El robot dispone de dos o más ruedas en cada lado, no está permitido el giro de las ruedas, por tanto, los giros se llevarán a cabo de manera similar a como se hace en la configuración diferencial. Esta configuración es la misma que se utiliza para la transmisión de movimiento en robots con orugas.



**Figura 31.** Esquema de configuración Skid-steer

Fuente: (Jiménez, 2017)

#### 3.1.2.4. Selección de la configuración cinemática del robot móvil

Para definir el tipo de configuración cinemática que llevará el robot móvil, es decir como estarán distribuidos los principales elementos que lo componen como son: ruedas, plataforma y motores, se analizó cuál es la mejor opción utilizando una matriz Pugh el resultado se visualiza en la Tabla 6.

**Tabla 6**

*Comparación de alternativas de configuración cinemática del robot.*

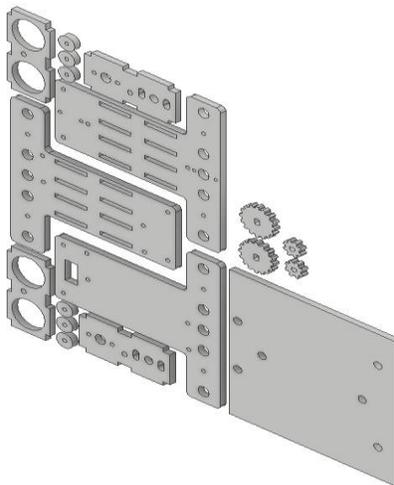
Criterios de selección	Configuración cinemática	
	Diferencial	Skid Steer
Facilidad de diseño	+1	-1
Bajo costo	+1	-1
Mantenibilidad	0	0
Menor peso	+1	-1
Tamaño reducido	+1	-1
Capacidad de desplazarse en terreno plano	0	0
Capacidad de desplazarse a velocidades bajas	0	0
Puntuación total	4	-4

Al evaluar la matriz Pugh se concluyó que la mejor alternativa para la configuración cinemática del robot es la diferencial, ya que brinda mayor facilidad para el diseño, es menos costosa su

implementación dado que para una configuración Skid-steer es necesario de la utilización de más ruedas y motores, al evitarse el uso de más de estos dos componentes su peso reducirá y por ende se puede hacer un diseño más compacto al solo utilizar dos ruedas motrices obteniendo un tamaño reducido del robot que le permita llegar a todos los puntos de la zona minada, además se asegura no superar los 30 kg que son el mínimo requerido para activar una mina antipersonal.

### 3.1.2.5. Modelado CAD de la plataforma

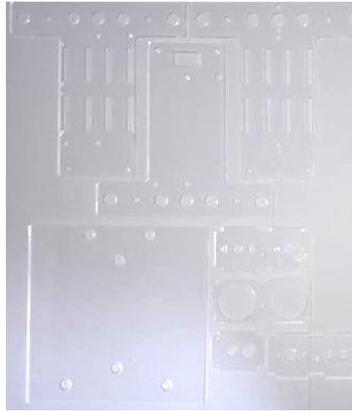
Teniendo en cuenta la exactitud que brindan las máquinas de corte laser en sus acabados y en su bajo costo de alquiler se optó por esta opción para realizar la estructura del robot. Inicialmente se diseñaron las piezas en un plano 2D en el programa Autodesk Inventor Professional 2018 y luego se transfirió las partes a un modelo 3D para ser ensambladas en el mismo programa, el despiece del chasis se puede visualizar en la Figura 32.



*Figura 32.* Modelo 3D de las piezas del robot.

Posteriormente se procedió a cortar las piezas en planchas de acrílico de 3mm de espesor debido a su durabilidad y facilidad de corte. En el anexo 1 se puede visualizar el plano de las piezas del robot

que se utilizó en la cortadora, así como también en la Figura 33 se observa las partes finalmente cortadas y listas para ser ensambladas.

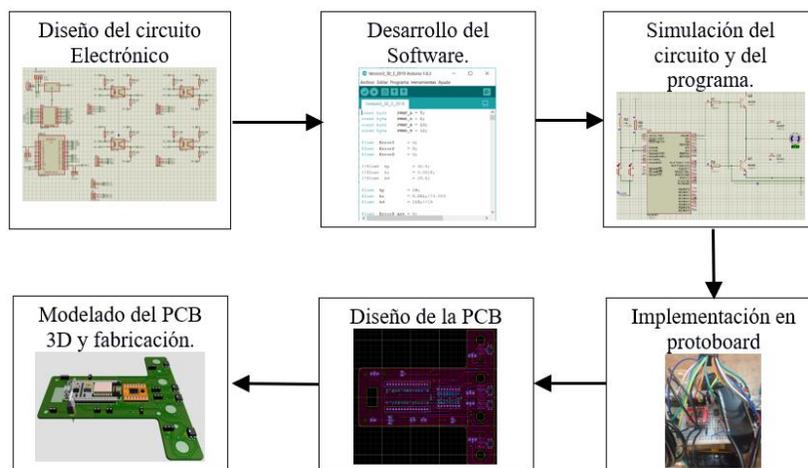


**Figura 33.** Piezas del robot listas para ser ensambladas.

#### **3.1.2.6. Diseño de la placa PCB**

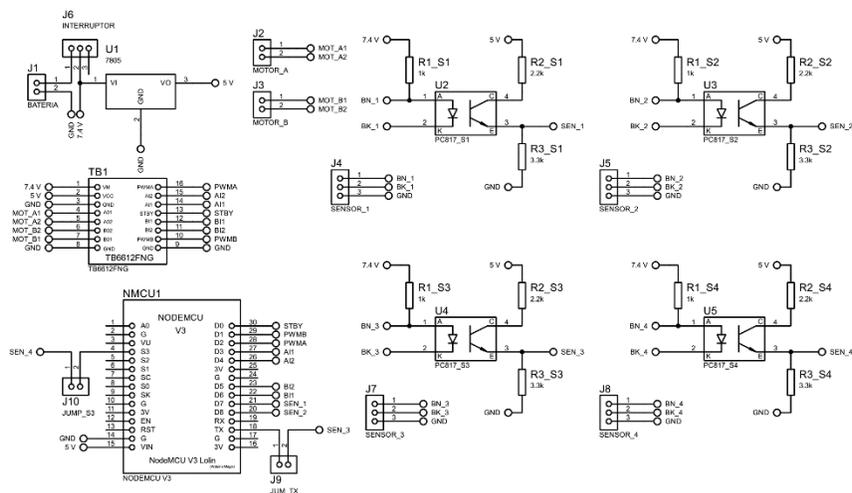
Para el diseño de la placa PCB (Printed Circuit Board) primero se tomó en cuenta las especificaciones y requerimientos que debe tener el mismo, por lo cual se inició escogiendo que elementos se van a utilizar y la interconexiones entre ellos, una vez definidos todos los componentes de la placa se realizó el esquemático del circuito, luego se desarrolló el software y verifiqué si este funciona mediante la simulación del mismo, antes de pasar a diseñar el circuito impreso se implementó el mismo en un protoboard para verificar si el funcionamiento será el adecuado al igual que en la simulación, finalmente se pasó al diseño del circuito impreso.

Para el diseño de la placa PCB se utilizó la herramienta de Proteus 8 Professional, el cual realizar el diseño y la simulación. El proceso de fabricación de la PCB se define en las etapas mostradas en la Figura 34.



**Figura 34.** Etapas para la fabricación de la PCB.

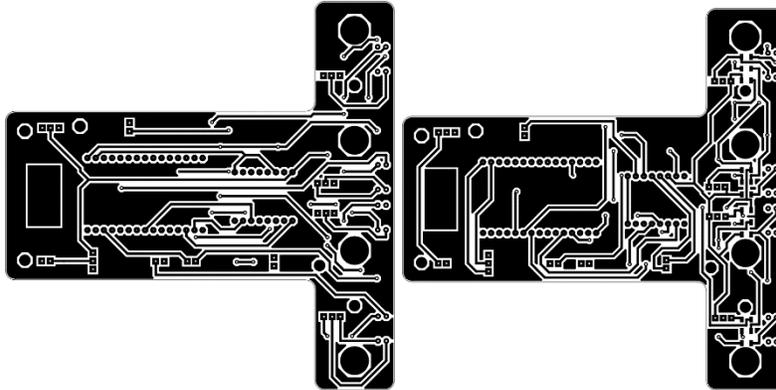
En el diseño del diagrama electrónico se utilizó la herramienta ISIS de Proteus, que permite tener una representación simbólica de los componentes del circuito y conectarlos entre sí, permitiendo poder ejecutar la simulación. El diagrama esquemático del circuito del robot se lo visualiza en la Figura 35.



**Figura 35.** Esquemático del circuito electrónico del robot.

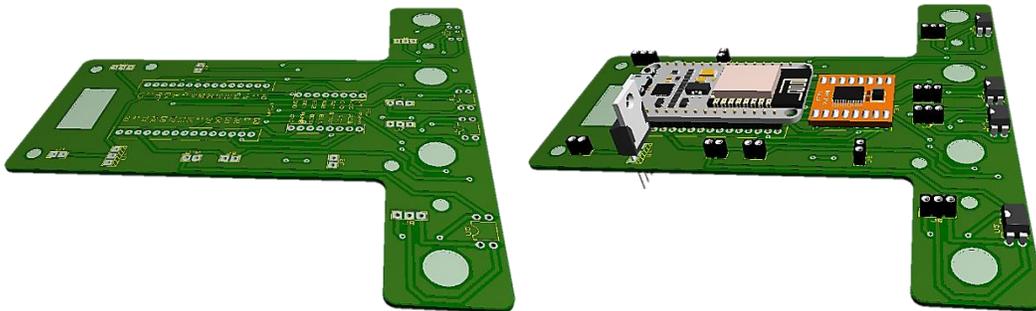
Para la etapa de diseño del PCB se utilizó la herramienta ARES de Proteus, que realiza el enrutado, ubicación y edición de componentes, pero previamente se necesita el diagrama realizado en ISIS, para indicarle a ARES como debe conectar los componentes en la PCB. Debido a la dificultad de las

conexiones del circuito y el requerimiento de acoplarse a la forma de la placa que se desea, se diseñó una placa a doble lado con vías conductoras en todos los agujeros para unir las dos capas, utilizando un diseño a doble lado como se puede ver en la Figura 36 se presenta el diseño final de los dos lados de la PCB.



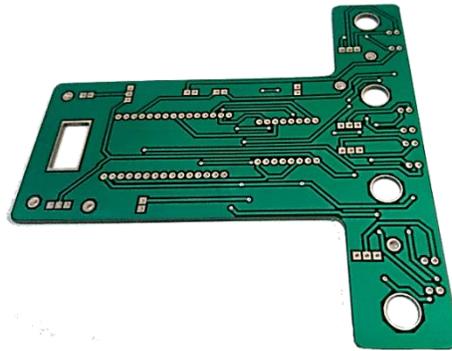
**Figura 36.** Diagrama multicapa de la PCB.

Finalmente se obtuvo el modelado 3D del diseño de la PCB en la que se puede verificar que los componentes concuerdan con el diseño, como se observa en la Figura 37.



**Figura 37.** PCB del robot móvil visualizado en 3D.

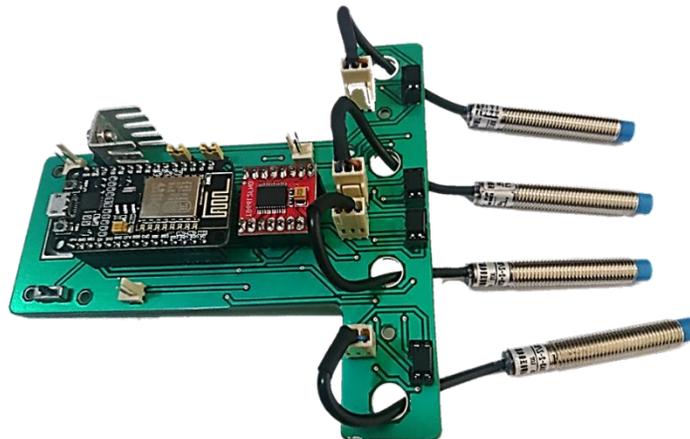
EL proceso de fabricación de la PCB, se lo realizó a través de una empresa que brinda el servicio de manufactura de prototipos electrónicos profesionales, para lo cual es necesario exportar el archivo en formato Gerber del diseño de Proteus, finalmente se obtiene la placa lista para montar y soldar los componentes como se muestra en la Figura 38.



*Figura 38.* PCB final del robot móvil.

### 3.1.2.7. Montaje del robot móvil

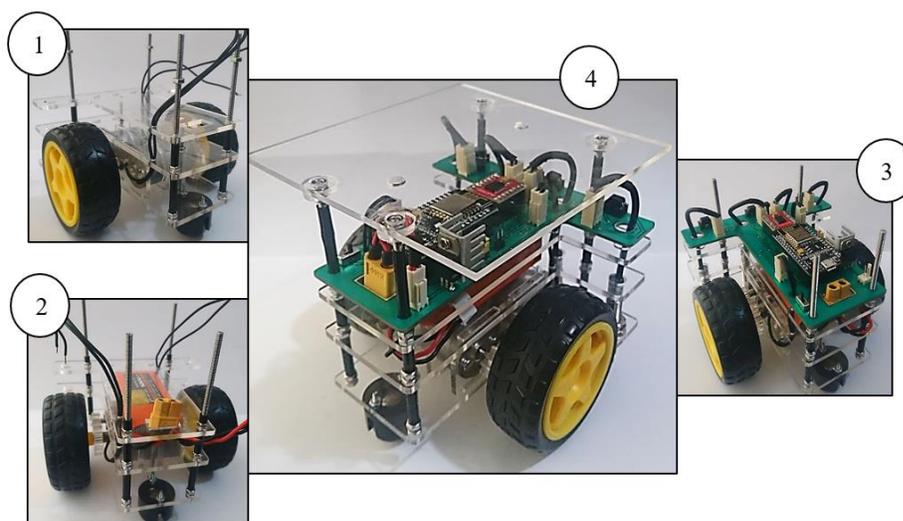
Una vez se realizó el proceso de diseño y elaboración de la PCB como se mostró en la sección anterior, por lo que se procedió a ubicar los componentes, como se muestra en la Figura 39 la placa de control totalmente terminada y funcional.



*Figura 39.* Placa de control del robot terminada.

Teniendo todas las partes que conformaran el robot, se procede a fijar y armar cada parte del mismo, el robot está constituido por 4 pisos que están distribuidos de la siguiente forma: en el piso 1 se acoplaron los motores de tracción del robot y los sensores de metal, en el piso 2 se colocó la batería, el piso 3 se destinó para la placa de control y el último piso es para ubicar el marcador identificador

del robot. El resultado final al armar los 4 pisos e interconectar todos los elementos se observa en el paso 4 de la Figura 40.



**Figura 40.** Proceso de ensamblaje del robot.

### 3.1.3. Estación en tierra

Para que sea posible la conexión con el dron y con la plataforma móvil terrestre es necesario de una estación de control remota, la cual estará formada por un ordenador con configuraciones previas de software para enviar las ordenes de control a ambos vehículos. El computador constara con Sistema Operativo Linux (Ubuntu 16.04 de 64 bits), sus características se pueden observar en la Tabla 7.

**Tabla 7**

*Especificaciones técnicas de la estación en tierra*

<b>Marca</b>	HP
<b>Modelo</b>	Hewlett-Packard
<b>Procesador</b>	Intel(R) Core™ i7-4510U CPU @ 2 GHz
<b>Memoria RAM</b>	8 GB
<b>Arquitectura</b>	64 bits

## **3.2. Descripción y configuración del Software del sistema**

Linux brinda ciertas prestaciones este es el caso del SDK de Parrot. Se desarrollará el software del proyecto en Python junto con la librería de procesamiento de imágenes OpenCV en su versión 3.1.0 debido a los métodos y algoritmos que posee.

### **3.2.1. Entorno de desarrollo ROS (Robot Operating System)**

ROS es un meta-sistema operativo, ya que, aunque ofrece las funciones de un sistema operativo debe ser instalado sobre la base de otro sistema operativo basado en UNIX, sus funcionalidades son desde abstracción de hardware de bajo nivel para control de dispositivos hasta el paso de mensajes entre procesos. Su objetivo es simplificar la tarea de crear cualquier robot imaginable, y darles las facilidades para que estos sean robustos y maleables mediante una serie de bibliotecas, drivers y herramientas. En ROS cada módulo posee autonomía similar a una red de internet en topología punto a punto, es decir todo se hace mediante mensajes enviados de acuerdo a un protocolo, en este caso el XML-RCP. Es una plataforma multilenguaje donde se puede programar en C++, Python, Octave o Java. Hace uso del protocolo TCP/IP para generar un esquema cliente servidor. (Nuñez Torres, León , & Cardenas, 2016)

Los elementos básicos que son parte de una red de procesos de ROS son:

- **Nodo:** Es el elemento estructural de ROS, se puede decir que cada nodo equivale a un proceso. Los elementos de un nodo son: publicaciones, suscripciones, prestación y petición de servicios y comportamiento principal. (Nuñez Torres, León , & Cardenas, 2016)
- **Maestro:** Es el nodo principal que proporciona la configuración, este es el principal ya que sin este los demás nodos no tiene la capacidad de trabajar por lo que no existiría un paso de mensajes o servicios. (Alvarez, 2018)

- Tópicos: Son etiquetas usadas como mecanismo para identificar el contenido de un mensaje y generar interacciones con los nodos.
- Mensajes: Son los datos que se transmiten entre los nodos, estos se caracterizan por tener un tipo de formato, estos viajan por un canal denominado tópico.
- Servicios: Se utiliza un mensaje de solicitud y otro de respuesta, un único nodo puede ser el prestador de servicio los demás solo pueden solicitarlo, este permanece en escucha mientras espera una solicitud.
- Paquete: Contiene todo el código fuente de los nodos, las librerías usadas, las cabeceras y cualquier otro recurso necesario para el funcionamiento de la plataforma.

### 3.2.2. Instalación de ROS en Ubuntu 16.04 LTS

Para instalar ROS en Ubuntu 16.04 LTS, se debe comprobar que el sistema se encuentre actualizado, ya que se necesita instalar ciertos drivers para su ejecución. Para lo cual se debe preparar el sistema operativo con el fin de que se acepte los paquetes que se van a instalar del repositorio “*packages.ros.org*” para esto en la ventana del terminal se debe escribir el siguiente comando:

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu/ precise main" > /etc/apt/sources.list.d/ros-latest.list'
```

Ahora se debe ejecutar la siguiente línea de comando que pertenece a la descarga del fichero correspondiente a la clave de producto.

```
$ wget http://packages.ros.org/ros.key -O - | sudo apt-key add—
```

Los pasos anteriores permiten preparar al sistema para la instalación de ROS, ahora es necesario actualizar el repositorio, por lo que se ejecuta la siguiente línea de comando:

```
$ sudo apt-get update
```

Ahora se debe instalar la versión de ROS con la cual se desea trabajar, para esto se ejecuta en siguiente línea de comando, que descarga los paquetes necesarios para que el sistema funcione correctamente.

```
$ sudo apt-get install ros-kinetic-desktop-full
```

Antes de utilizar ROS, se debe inicializar “*rosdep*” que permite instalar fácilmente las dependencias del sistema para que pueda ejecutar los componentes principales.

```
$ sudo rosdep init
```

```
$ sudo rosdep update
```

También se debe hacer la configuración del entorno de trabajo, para esto se debe añadir esta línea antes de ejecutar o levantar el servicio.

```
$ echo "Source/opt/ros/kinetic/setup.bash" >> ~/.bashrc
```

### **3.2.3. Driver bebop-autonomy**

El paquete `bebop_autonomy` es un controlador de ROS para drones específicamente para el Parrot Bebop en sus versiones 1.0 y 2.0, realizado por investigadores de la universidad Simon Fraser, basado en Parrot ARDroneSDK3, este controlador contiene el nodo `bebop_driver` el cual provee los servidores para los servicios ofrecidos.

#### **3.2.3.1. Instalación del driver bebop-autonomy para el Parrot Bebop 2**

Este driver permite la interacción con el drone, ya que cuando se ejecuta se activan una serie de tópicos que permiten la creación de suscriptores para recibir información enviada desde una computadora, mientras que también se pueden publicar para enviar comandos directos al drone. Para la instalación del driver es necesario crear un espacio de trabajo llamado `bebop_ws` dentro del

cual se debe hacer la clonación del driver para que pueda ejecutarse correctamente, para esto es necesario ejecutar las siguientes líneas de comando:

```
$ mkdir -p ~/bebop_ws/src && cd ~/bebop_ws
$ catkin init
$ git clone https://github.com/AutonomyLab/bebop_autonomy.git src/bebop_autonomy
$ rosdep update
$ rosdep install --from-paths src -i
$ catkin build
```

### 3.2.3.2. Obtención de imágenes de la cámara a bordo del Bebop 2

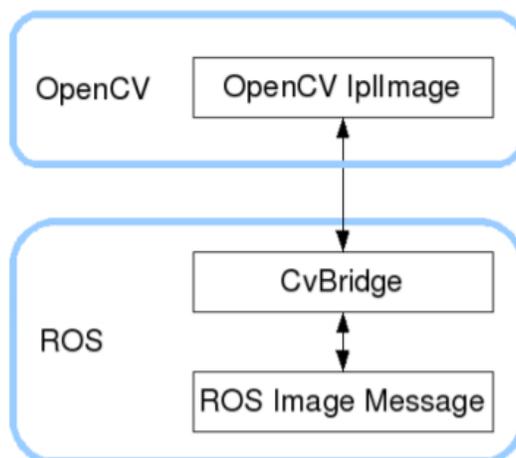
Para la adquisición de la imagen se utiliza la cámara del Bebop 2, esta se maneja mediante el entorno de ROS y por ende el video se encuentra en un formato de mensajes de tipo “/image”, está debe ingresar a una conversión de imágenes de ROS a OpenCV, para que se pueda procesar dicha imagen.

Se debe ejecutar al controlador como un nodo, denominada como “*bebop\_driver\_node*” y se encuentra en el paquete “*Bebop\_driver*”, esto permite abrir la interfaz de la cámara ROS y publicar la información de la cámara. debe escribir la siguiente instrucción en el terminal:

```
$ roslaunch bebop_driver bebop_node. Launch
```

### Conversión de imágenes de ROS a OpenCV

La imagen que se presenta en ROS se encuentra en el formato “*sensor\_msgs/image*”, por lo que se necesita cambiar este formato para procesar en OpenCV, que maneja las imágenes como matrices es por esto que se debe usar la biblioteca “*cv\_bridge*” que facilita la interacción entre ROS y OpenCV.



**Figura 41.** Interacción entre ROS y OpenCV

Fuente: (Alvarez, 2018)

### 3.2.3.3. Control de movimientos del dron

El driver de bebop\_autonomy permite controlar ciertas acciones del dron, así como, aterrizar, navegar, despegar, movimiento de la cámara, etc. En la siguiente se puede verificar los tópicos con su respectivo tipo de mensaje que se activan cuando se ejecuta el nodo master.

**Tabla 8**

*Tópicos principales del driver Bebop\_autonomy*

Topic	Tipo de mensaje	Descripción
Takeoff	std_msgs/Empty	Despegar
Land	std_msgs/Empty	Aterrizar
Reset	std_msgs/Empty	Detener motores
cmd_vel	geometry_msg/Twist	Control de movimiento
camera_control	geometry_msg/Twist	Movimiento de la cámara
Image_raw	sensor_msgs/Image	Transmisión de video

Fuente: (Heredia, 2018)

Los valores que se manejan para el tópico *cmd\_vel* se encuentran en el rango de [-1: 1], el dron ejecuta el último comando recibido mientras el controlador esté en ejecución, para que se desplace

se debe publicar un mensaje en la configuración de dicho tópico. El tipo de mensaje que se puede enviar a través del tópico es *geometrymsgs* los parámetros del mensaje describen el movimiento que debe seguir el drone estos se configuran de acuerdo a la Tabla 9.

**Tabla 9**

*Mensajes para el movimiento del dron con driver Bebop\_autonomy*

Mensaje		Descripción
<b>Linear.x:</b>	(+)	Adelante
	(-)	Atrás
<b>Linear.y:</b>	(+)	Izquierda
	(-)	Derecha
<b>Linear.z:</b>	(+)	Ascender
	(-)	Descender
<b>Angular.z</b>	(+)	Sentido antihorario
	(-)	Sentido horario

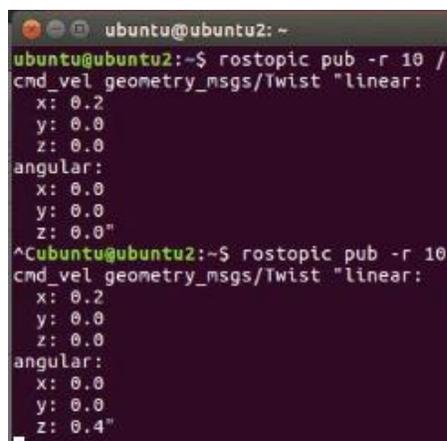
Fuente: (Monajjemi, 2017)

En la Figura 42 se indican los vectores los cuales representan la velocidad lineal y angular en los diferentes ejes, es decir mediante la Tabla 9 se puede configurar estos valores para adoptar la función que se necesite adquirir para el movimiento del drone.



**Figura 42.** Esquema de coordenadas del UAV.

En Python se realiza un objeto de tipo *publicador*, el cual será enviado por línea de comando activando el tópico */cmd\_vel*, el cual tiene un mensaje de tipo Twist, el cual permite la manipulación de los parámetros para el movimiento del dron, en la Figura 43 se muestra un ejemplo donde el dron se moverá hacia adelante con giro en sentido antihorario.



```

ubuntu@ubuntu2: ~
ubuntu@ubuntu2:~$ rostopic pub -r 10 /
cmd_vel geometry_msgs/Twist "linear:
  x: 0.2
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0"
^Cubuntu@ubuntu2:~$ rostopic pub -r 10
cmd_vel geometry_msgs/Twist "linear:
  x: 0.2
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.4"

```

**Figura 43.** Ejemplo de configuración del de los movimientos del dron.

#### 3.2.3.4. Control de movimientos de la cámara del dron con bebop-autonomy

Para el movimiento de la cámara virtual del Bebop, es necesario publicar un mensaje de tipo *geometry\_msgs/Twist* en el tópico de *camera\_control* estos mensajes sirven para que se pueda establecer una inclinación de la cámara. En la Tabla 10 se muestra la configuración de los parámetros del objeto Twist enviado a través del tópico.

**Tabla 10**

*Mensajes para el movimiento de la cámara del dron*

Mensaje		Descripción
Angular.x:	(+)	Inclinar hacia abajo
	(-)	Inclinar hacia arriba
Angular.z:	(+)	Panorámica izquierda
	(-)	Panorámica derecha

Para el desarrollo del proyecto se necesita ubicar la cámara en vista cenital es por esto que se configura el mensaje angular.x cambiando el valor a -90.

### 3.2.3.5. Comunicación ROS y ESP8266

Para la comunicación del robot con la estación en tierra se utilizó el protocolo roserial, que nos permite extender las capacidades de mensajería de ROS a través de puertos serie y sockets de red, debido a que es necesario que el sistema sea inalámbrico se utilizarán sockets, para lo cual primero es necesario instalar las librerías correspondientes en ROS ejecutando las siguientes líneas de comandos en el terminal:

```
$ roscd
```

```
$ cd ../src
```

```
$ git clone https://github.com/ros-drivers/roserial.git
```

```
$ roscd
```

```
$ catkin_make
```

```
$ catkin_make install
```

Una vez que se han instalado las librerías en ROS es necesario configurar el IDE de Arduino, por medio del cual se programará y cargará el software a la placa de control NodeMcu del robot. Para realizar la configuración se dirige al directorio local de las librerías de Arduino en el terminal y posteriormente ejecutaremos la siguiente línea de comando:

```
$ rosrund rosserial_arduino make_libraries.py
```

### 3.3. Topología de red del sistema

Es sistema tendrá una topología de red de tipo estrella donde el nodo principal será el drone por donde pasará toda la información ya sea que la estación en tierra envió los datos al drone o al robot móvil, en la Figura 44 se puede observar una representación gráfica de la conexión.

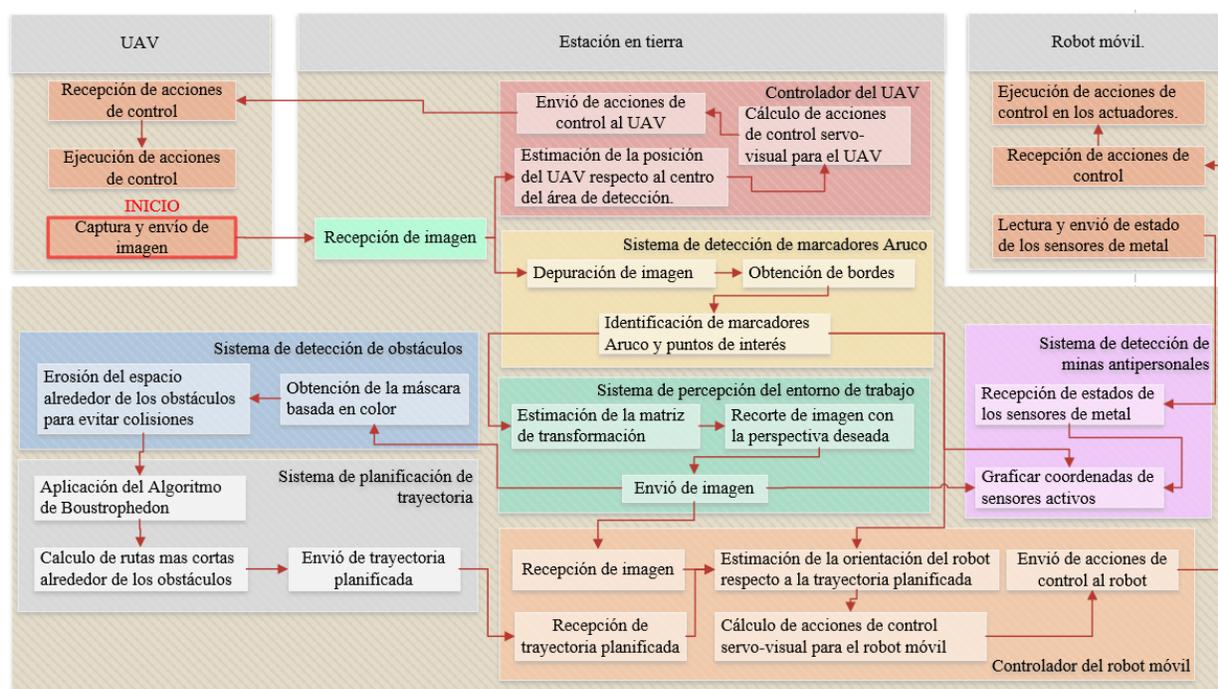


*Figura 44.* Topología de la red del sistema

## CAPÍTULO IV

### SISTEMA DE NAVEGACIÓN AUTÓNOMA Y DETECCIÓN DE MINAS ANTIPERSONALES

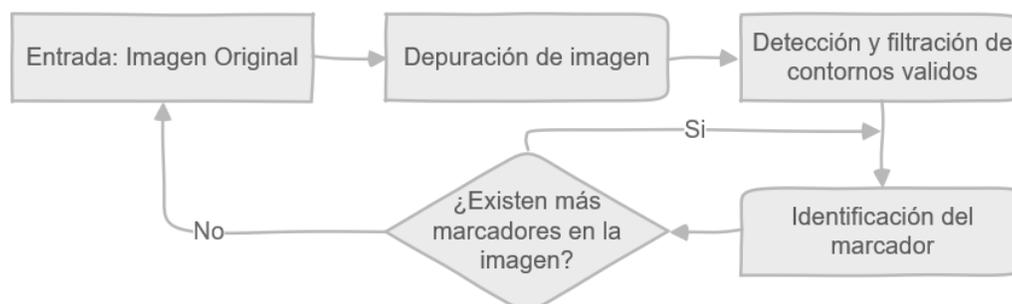
Una vez se ha construido el robot móvil, se plantea el objetivo de lograr la locomoción del mismo para recorrer la zona minada en su totalidad mediante las imágenes provistas por el UAV y a su vez en el trayecto del robot detectar posibles minas antipersonales en el suelo. En la Figura 45 se observa el diagrama de flujo de los sistemas implementados en cada componente del proyecto y como están interconectados entre ellos.



*Figura 45.* Esquema total de sistema.

## 4.1. Sistema de Detección de Marcadores ARUCO

El proceso de identificación de marcadores Aruco presentes en una imagen conlleva una secuencia extensa de procedimientos con el fin de que el algoritmo sea robusto y estable. En la Figura 46 se presenta el proceso general para detección de marcadores Aruco.



**Figura 46.** Diagrama de flujo del proceso de detección de marcadores Aruco.

Cada procedimiento descrito en el diagrama de flujo de la Figura 46 se describe a continuación:

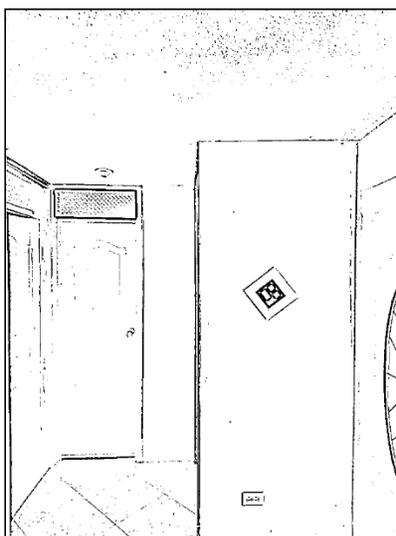
### 4.1.1. Captura y depuración de imagen

Para la lectura de marcadores ARUCO se inicia tomando la imagen enviada por el drone por medio del tópico `/bebop/image_raw` del paquete `bebop_autonomy`, posteriormente se convierte la imagen a escala de grises como se muestra en la Figura 47, mediante los algoritmos de OpenCV en Python.



**Figura 47.** Conversión de imagen a escala de grises.

A partir de la imagen en escala de grises se procederá a binarizar la imagen mediante el algoritmo de umbral adaptativo, el cual calcula un umbral para pequeñas regiones de la imagen y los va comparando por píxeles, si el valor del píxel es mayor que el valor umbral de la región se le asigna un 1 caso contrario será 0. La principal ventaja de utilizar el umbral adaptativo es que la iluminación variable de la imagen no afecta en el resultado de la umbralización, como se muestra en la Figura 48.



*Figura 48.* Binarización de umbral adaptativo.

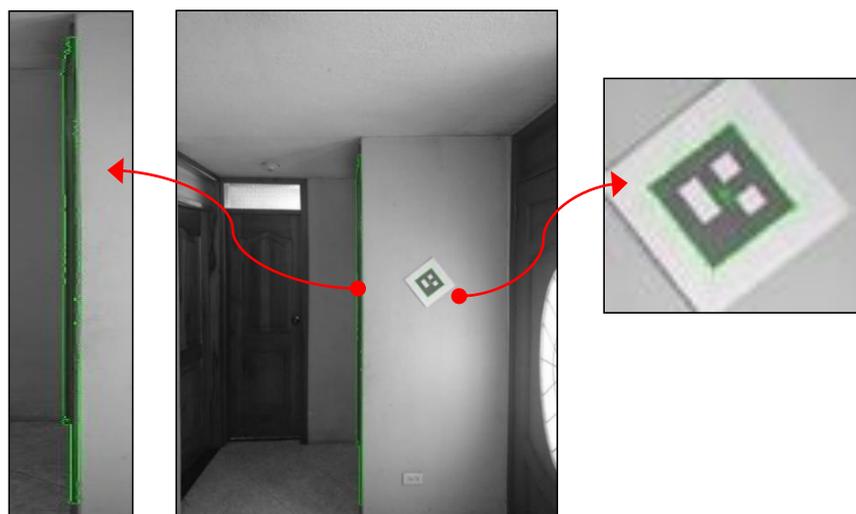
#### **4.1.2. Detección y filtración de contornos validos**

Con la imagen de la Figura 48 se procederá a encontrar los contornos con el algoritmo de (Suzuki & Abe, 1985), el cual determina como están relacionados los bordes internos y externos de un objeto, este algoritmo se encuentra en las librerías de OpenCV y se obtiene como resultado tres listas: la primera es la imagen, la segunda es la lista de todos los contornos y cada elemento de esta lista es una matriz de tipo Numpy de coordenadas (x,y) que representan los puntos del límite del objeto, la tercera lista es la información sobre la jerarquía del contorno, es decir si existen contornos anidados este algoritmo los clasifica como padres e hijos designando el contorno más externo como el padre. En la Figura 49 se muestra el resultado obtenido del proceso.



**Figura 49.** Contornos de la imagen.

A partir de los datos del proceso anterior se seleccionará los contornos que tengan dos o más jerarquías, dado que los marcadores tienen cuadriláteros dentro de ellos deberán cumplir con la condición, con esto se elimina la mayoría de los contornos innecesarios. Como se puede observar en la Figura 50 y comparándola con la Figura 49 la depuración es casi completa solo han quedado dos posibles lugares en los que podrían existir marcadores.



**Figura 50.** Depuración de contornos.

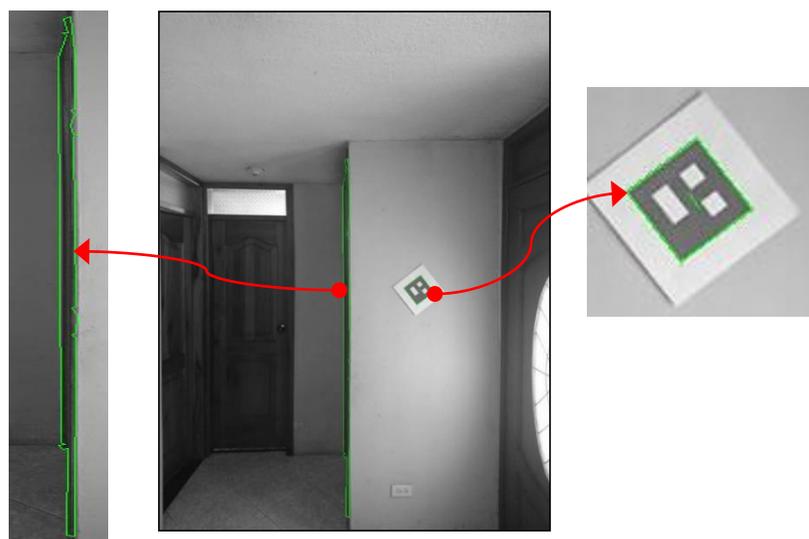
En base a los contornos obtenidos en el punto anterior donde posiblemente existen marcadores, se realiza una aproximación mediante el algoritmo de Douglas-Peucker, obteniendo desde el contorno de la forma base otro con menos número de vértices dependiendo de la precisión que se le especifique en el parámetro  $\epsilon$  del algoritmo, como se puede observar en el ejemplo de la Figura 51 el cuadrilátero de color verde presenta una aproximación con un  $\epsilon$  de 10% y a continuación se lo realiza con un  $\epsilon$  de 1% que se asemeja mucho más a la forma original, para el proyecto de investigación se encontró que un valor de  $\epsilon$  de 2.5% resulta adecuado.



**Figura 51.** Aproximación de contorno.

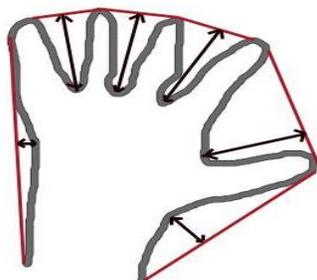
Fuente: (OpenCV-Python Tutorilas, 2019)

El proceso de aproximación ayuda a definir la forma de cuadrilátero de los contornos, como resultado de esta función se obtiene una matriz de vértices con una considerable menor cantidad de puntos que representan los contornos de los objetos, se puede observar el resultado en la Figura 52.



**Figura 52.** Aproximación de contorno de los marcadores.

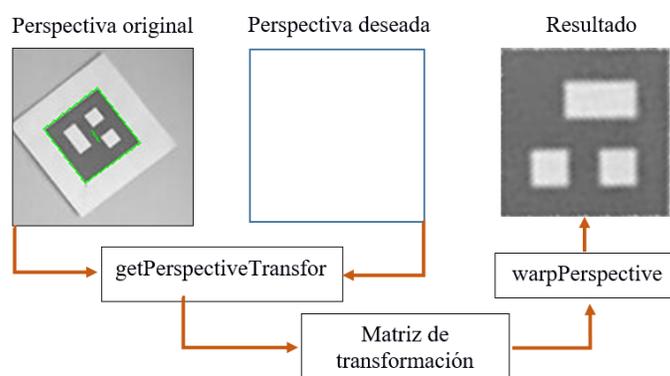
Se procederá a corregir los defectos de convexidad que tengan las formas para asegurarnos de que sea un cuadrilátero no convexo, por ejemplo, la mano presentada en la Figura 53 tiene defectos de convexidad, la línea roja representa el marco convexo de la mano y las flechas muestran los defectos.



**Figura 53.** Casco convexo de una forma.

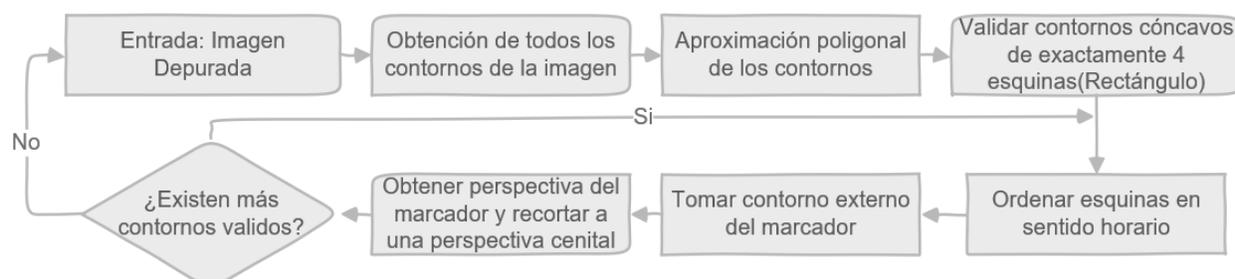
Fuente: (OpenCV-Python Tutorilas, 2019)

Mediante el algoritmo de Sklansky se corrigen los defectos de convexidad que tengan las formas y se valida que las mismas estén formadas por 4 segmentos, cualquier contorno que cumpla con la condición será dado como válido para un posible marcador. En el caso de la Figura 52 solo existió un contorno que cumplió con lo requerido, mismo que se procesó para obtener la matriz de transformación a un espacio cuadrado definido y con perspectiva cenital, posteriormente se realizó el recorte de la zona de interés. El resultado y proceso se observa en la Figura 54.



**Figura 54.** Obtención de la perspectiva del marcador.

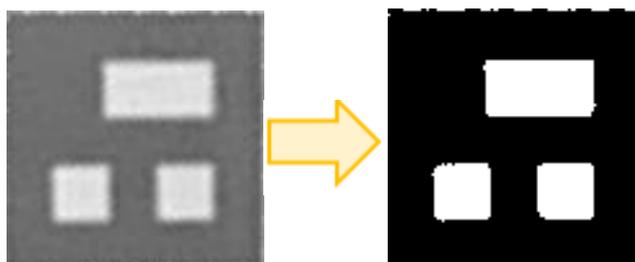
En el diagrama de flujo de la Figura 55 describe el procedimiento para la detección y filtración de contornos validos descrito en la presente sección.



**Figura 55.** Diagrama de flujo del proceso de detección y filtración de contornos validos

#### 4.1.3. Identificación del marcador

Ahora que se tiene solo la imagen de la zona del posible marcador, se la acondicionará para poder ser decodificada. Se inicia aplicando el algoritmo de binarización de Otsu que es un método de segmentación empleada cuando existe una clara diferencia de los objetos y el fondo, el algoritmo encuentra un umbral optimo desde el histograma de la imagen para luego comparar cada pixel si es mayor que el umbral será 1 caso contrario 0, de este modo se tiene la imagen binarizada como se observa en la Figura 56.



**Figura 56.** Resultado de la binarización de Otsu.

Para obtener una visualización más limpia de los rectángulos del marcador se utilizará la operación morfológica de erosión, con anterioridad se debe definir un kernel es decir la matriz de pixeles que se desea deslizar a través de la imagen, el algoritmo toma un pixel si todos los elementos del kernel

correspondiente al pixel son 1 el pixel será 1 caso contrario se pone a cero o también llamado erosionar. En la Figura 57 se visualiza una mejora notoria en la definición de los rectángulos y también elimina el ruido en los bordes de la imagen.



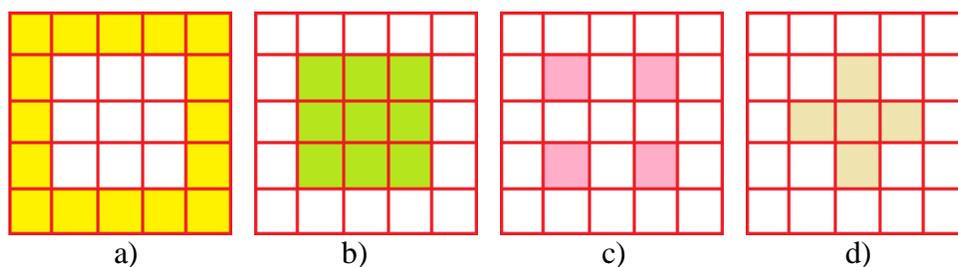
*Figura 57.* Erosión de la imagen.

En el último paso para acondicionar la imagen es redimensionarla al tamaño de la cuadrícula que tiene nuestros marcadores para el caso es de 5x5 cuadros, a través de este proceso se logra conseguir una matriz de 5x5 píxeles y a la vez binaria para proceder a decodificar el marcador. El resultado al redimensionar la imagen se muestra en la Figura 58.



*Figura 58.* Redimensionamiento de imagen.

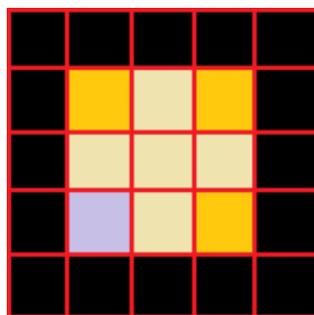
Los marcadores ARUCO con rejilla de 5x5 están dispuestos de la siguiente forma: los cuadros de los bordes sirven para contener el dato y siempre tienen un valor de 0 (Figura 59-a), la matriz de 3x3 del centro contiene la posición y el dato (Figura 59-b), las esquinas de la matriz central son para posicionar al código (Figura 59-c) y la cruz de la matriz son los datos (Figura 59-d).



**Figura 59.** Partes de los marcadores ARUCO.

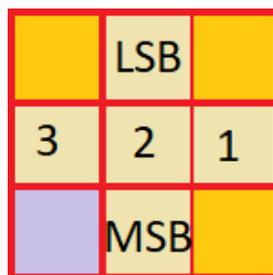
Se pueden generar 64 marcadores con la rejilla de 5X5, codificados del 0 al 63 y dispuestos de la siguiente manera:

- Los primeros 32 marcadores del 0 al 31 tienen todos los cuadros naranjas de la Figura 60 en 1 y el morado en 0, y se generan 32 combinaciones con los 5 bits de la cruz marrón.
- Los otros 32 marcadores del 32 al 63 surgen al tener los cuadros naranjas de la Figura 60 en 0 y el morado en 1, al igual que en el anterior caso se los combina con los 5 bits de la cruz marrón.



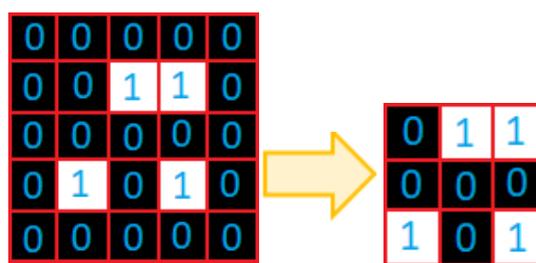
**Figura 60.** Generación de marcadores Aruco.

En la Figura 61 se presenta los bits para decodificar el marcador y se los toma en el siguiente orden: MSB 1 2 3 LSB.



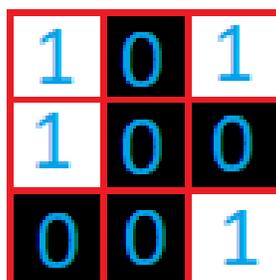
**Figura 61.** Orden de los bits del marcador.

Siguiendo con el proceso de decodificar los datos de la matriz de 5x5 pixeles se eliminará el contorno de la misma, teniendo como resultado una matriz de 3x3 presentada en la Figura 62.



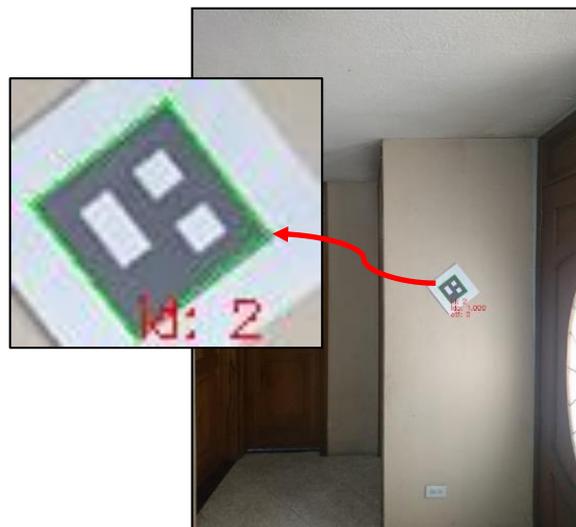
**Figura 62.** Marcador encontrado sin el contorno.

A continuación, se orienta la matriz de acuerdo con el orden descrito, para lo cual se verificará cuantas esquinas hay en 1 y cero, si hay 3 o 1 el marcador es posiblemente uno correcto, en el caso de nuestra matriz obtenida es necesario rotar 90° en sentido antihorario con la función rot90, el análisis de la posición de las esquinas nos brindará la información de cuantas veces se debes rotar la matriz para tener el resultado de la Figura 63.



**Figura 63.** Marcador en la posición deseada.

Luego se decodificarán los datos de la matriz de 3x3 como se indicó anteriormente obteniendo el número binario 00010 o en decimal 2. Para encontrar el ID del marcador se debe tener en cuenta el número de esquinas en uno del mismo si son 3 no se suma nada al valor binario encontrado si es una sola esquina en blanco se le suma el valor binario 10000 o en decimal 32. Finalmente, el ID de nuestro marcador encontrado es 2 ya que posee 3 esquinas en 1, dado el caso que exista más de 1 marcador en la imagen se realizará todo el proceso para cada uno y posteriormente se graficará el contorno y escribirá el ID en la imagen original como se muestra en la Figura 64.



**Figura 64.** Resultado de graficar el borde en el marcador y su ID.

#### **4.2. Sistema de Estabilización de Posicionamiento del UAV en el entorno**

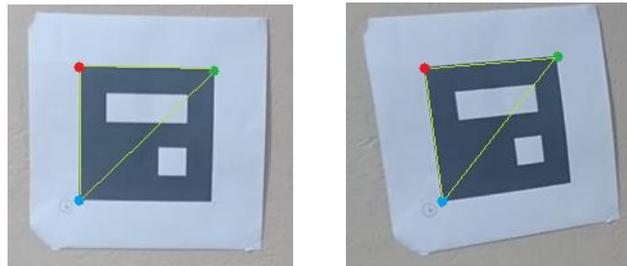
Con el fin de evitar que el dron se desplace del campo de visión que contiene el área minada debido a perturbaciones externas y además tener una perspectiva cenital, se procedió a realizar un controlador para el UAV que lo ubique en el centro del área minada y se mantenga en esa posición durante todo el proceso de barrido del área.

#### 4.2.1. Estimación del modelo de movimiento del UAV.

Basándose en el estudio de (Aguilar & Salcedo, 2018) en el que se logró obtener un modelo matemático para el Bebop 1, en este trabajo se aplicará el mismo procedimiento, pero para el UAV Bebop 2 mediante el movimiento entre fotogramas consecutivos provistas de su cámara a bordo con una tasa de transmisión de 30Hz.

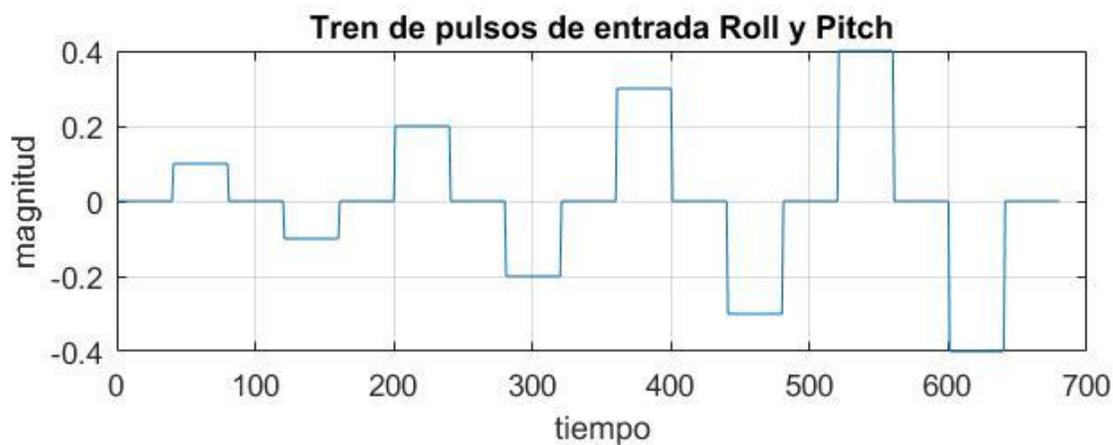
Para estimar el movimiento del UAV se registra un video durante el vuelo y luego se realizó el proceso de comparación desde la segunda imagen capturada, calculando la matriz de transformación que se ajuste al movimiento del marcador ubicado en el suelo.

La transformación affine es la relación entre dos imágenes tomando como referencia tres puntos en cada una de ellas. Como base para establecer la deformación entre imágenes consecutivas se utilizó el algoritmo desarrollado en la sección 4.1 con el cual se establecerán 3 esquinas del marcador como se muestra en la Figura 65 tomándolas como referencia en cada fotograma.



**Figura 65.** Puntos de referencia entre fotogramas.

La toma del video se realizó con una inclinación de  $-90^\circ$ , y se envió al UAV una señal tipo tren de pulsos con varias amplitudes permisibles para pitch y roll en el rango de  $[-1$  a  $1]$ , debido al tiempo de procesamiento que requiere el algoritmo de la sección 4.1, para altas velocidades de movimiento del UAV no es posibles reconocer de forma adecuada los marcadores por lo que el tren de pulsos se envió con valores de  $[-0.4$  a  $0.4]$ , la señal de control se presenta en la Figura 66.

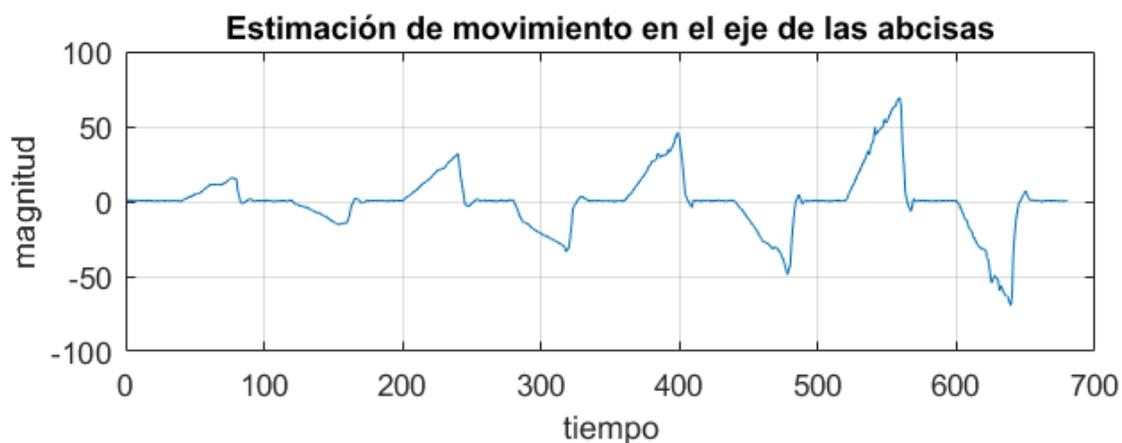


**Figura 66.** Tren de pulsos de entrada para Roll y Pitch

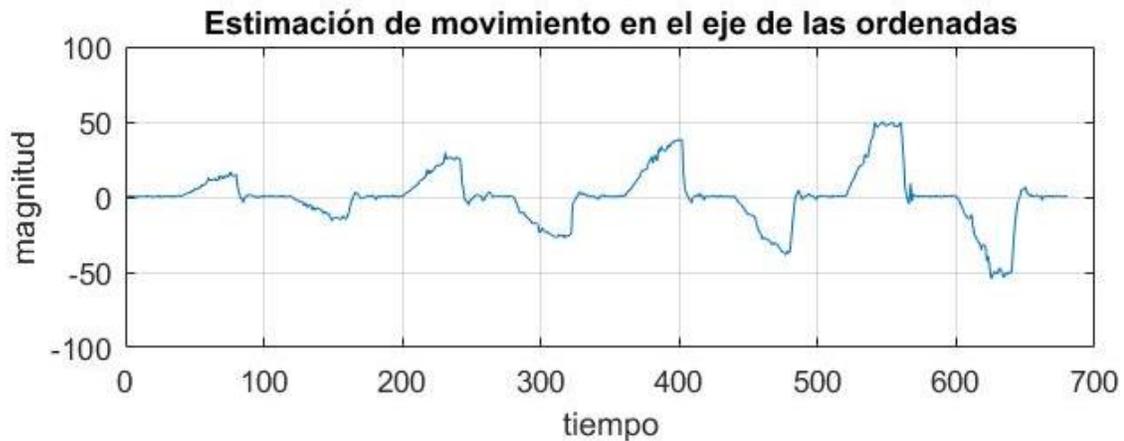
Mediante la función `getAffineTransform` de OpenCV se obtiene una matriz de 2 filas por 3 columnas, con los parámetros de traslación x,y en la tercera columna de la matriz como se puede apreciar en la ecuación (7).

$$M = \begin{pmatrix} a & b & t_x \\ c & d & t_y \end{pmatrix} \quad (7)$$

Una vez realizado el proceso entre todos los fotogramas consecutivos se guarda los valores de desplazamiento estimados por la transformación affine en cada instante de tiempo como se muestra en la Figura 67 y Figura 68.



**Figura 67.** Actuación del sistema para el control en Roll.



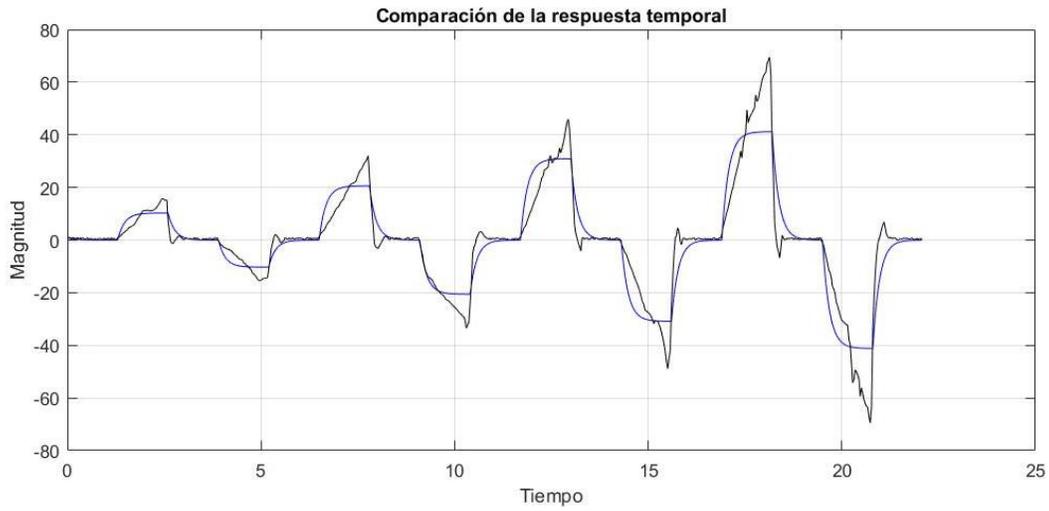
**Figura 68.** Actuación del sistema para el control en Pitch.

Para estimar la planta se utiliza la función “ident” que permite describir la dinámica de un sistema a partir de datos temporales, en el parámetro de entrada se colocó la señal de control obteniendo a la salida los desplazamientos deseados, el tiempo de muestreo se tomó de acuerdo con la frecuencia de envío de imágenes de la cámara del drone la cual es de 30Hz consiguiendo la función de transferencia de la ecuación (8).

$$G(s) = \frac{K_p}{1 + \tau_p s} = \frac{A}{B} \quad (8)$$

Con una ganancia de  $K_p = 103.02$ , y constante de tiempo  $\tau_p = 0.1734$  para Roll y para Pitch  $K_p = 95.5274$  y  $\tau_p = 0.15096$ .

A continuación, se visualiza en la Figura 69 y Figura 70 el modelo con los datos obtenidos para Roll y Pitch.



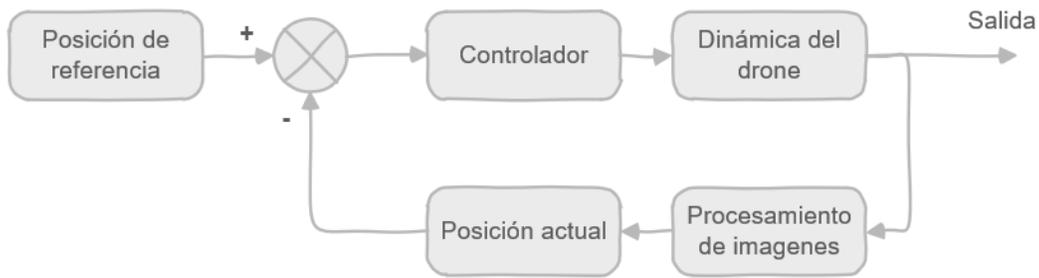
**Figura 69.** Caracterización del modelo para la planta en Roll.



**Figura 70.** Caracterización del modelo para la planta en Pitch.

#### 4.2.2. Diseño del controlador.

Con el fin de posicionar al dron en el centro del área minada se implementó un controlador que permita mantener la posición hasta que se concluya el barrido, en la Figura 71 se muestra el diagrama de control para la posición del dron.

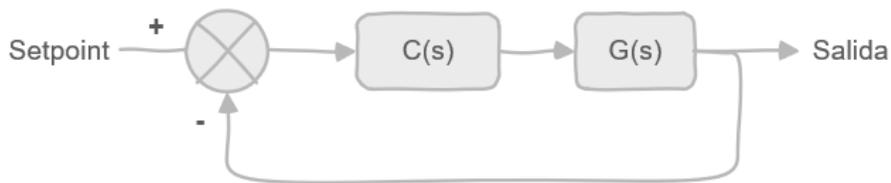


**Figura 71.** Esquema de control para el posicionamiento del dron mediante realimentación de la posición visual.

A partir del estudio de (Aguilar & Quisaguano, 2018) en el que se plantea el uso de un controlador PI para el manejo del dron con buenos resultados se diseñará este controlador para el dron Bebop 2 mediante el método de ubicación de polos en el que se determina las ganancias del controlador PI de la forma de la ecuación (9).

$$C(s) = \frac{K_c(1 + \tau_i s)}{\tau_i s} = \frac{D}{E} \quad (9)$$

El lazo de control representado en su forma paralela se visualiza en la Figura 72.



**Figura 72.** Esquema del diseño del controlador.

La función de transferencia de lazo cerrado está dada por la expresión:

$$H(s) = \frac{C(s)G(s)}{1 + C(s)G(s)} \quad (10)$$

Reemplazando las ecuaciones (8) y (9) en (10):

$$H(s) = \frac{\frac{D A}{E B}}{1 + \frac{D A}{E B}} = \frac{D A}{E B + D A} \quad (11)$$

$$H(s) = \frac{K_c K_p (\tau_i s + 1)}{\tau_i \tau_p s^2 + \tau_i (1 + K_c K_p) s + K_c K_p} \quad (12)$$

Siendo el polinomio característico del sistema de control:

$$p(s) = s^2 + \frac{1}{\tau_p} (1 + K_c K_p) s + \frac{K_c K_p}{\tau_i \tau_p} \quad (13)$$

Se desea que el polinomio característico tenga la forma general deseada de una función de transferencia de segundo orden:

$$p(s) = s^2 + 2\zeta\omega_n s + \omega_n^2 \quad (14)$$

Tomando un máximo sobre impulso de  $M_p = 10\%$ :

$$M_p = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \quad (15)$$

Resolviendo se tiene que el factor de amortiguamiento es:

$$\zeta = 0.5912$$

Como condición de diseño se propone un tiempo de establecimiento de 1.5 segundos con una tolerancia de 2% obteniendo que:

$$t_s = \frac{4}{\zeta\omega_n} \quad (16)$$

Para hallar la frecuencia natural  $\omega_n$  se despeja de la ecuación (16) y se reemplaza los valores encontrados y propuestos como diseño.

$$\omega_n = \frac{4}{\zeta t_s} = \frac{4}{1.5 \times 0.5912} = 4.5106 \quad (17)$$

Reemplazando  $\omega_n$  y  $\zeta$  en el polinomio característico deseado de la ecuación (14):

$$p(s) = s^2 + 5.331s + 20.34 \quad (18)$$

Igualando los coeficientes de la ecuación (13) y (18) se tiene que:

$$K_c = \frac{5.331\tau_p - 1}{K_p} \quad (19)$$

$$\tau_i = \frac{K_c K_p}{20.34\tau_p} \quad (20)$$

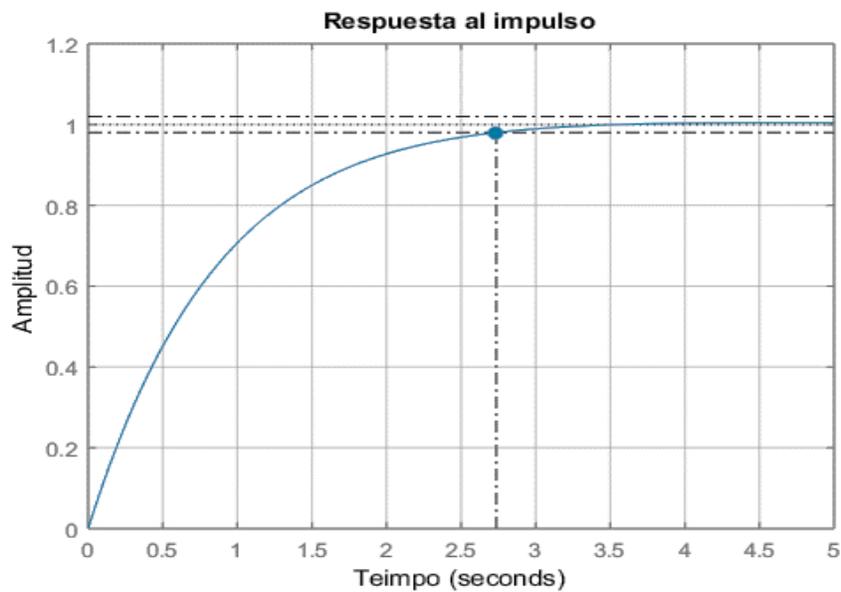
Remplazando los valores de  $K_p$  y  $\tau_p$  de Roll y Pitch en las ecuaciones (19) y (20) se determina las constantes del controlador para Roll.

$$K_c = -0.00073 \quad \tau_i = -0.021322$$

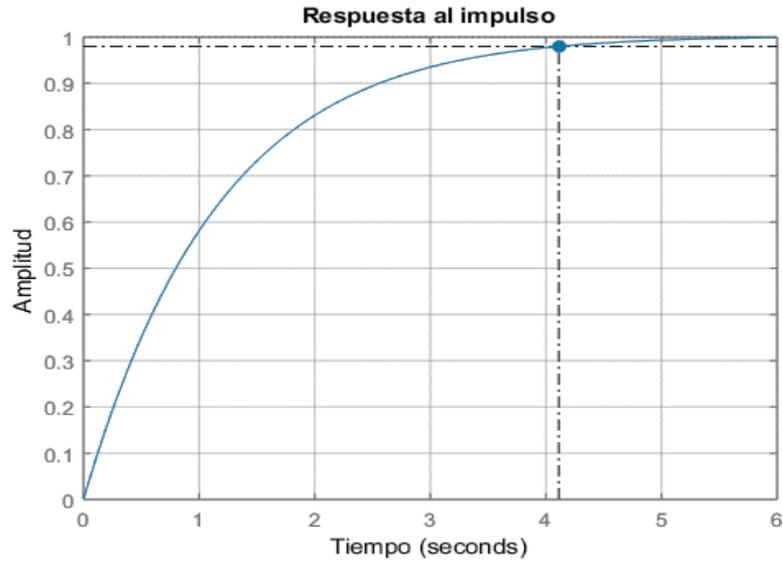
Ganancias para Pitch:

$$K_c = -0.00204 \quad \tau_i = -0.06346$$

Mediante la respuesta al impulso unitario en la Figura 73 y Figura 74 se comprueba el funcionamiento del controlador de acuerdo a los parámetros impuestos.



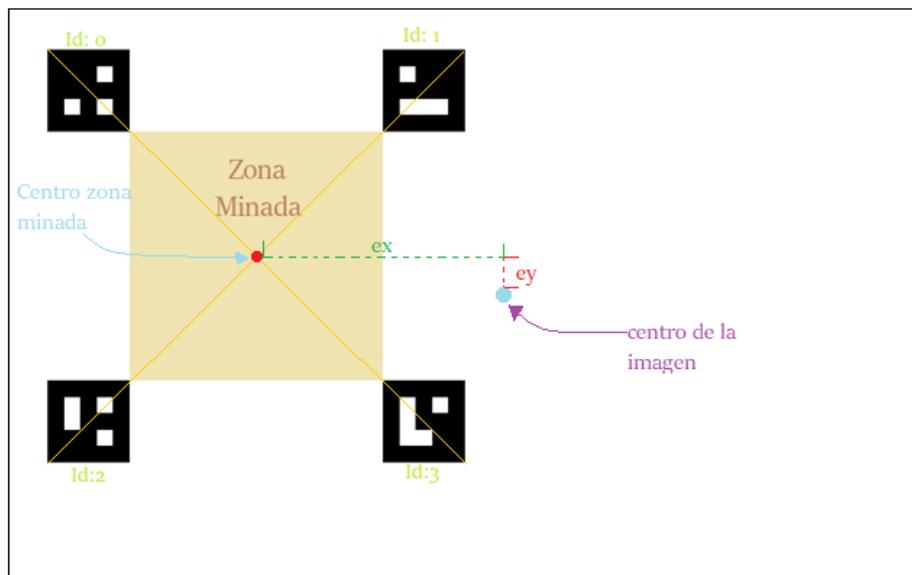
**Figura 73.** Respuesta al impulso para la planta en Roll.



**Figura 74.** Respuesta al impulso para la planta en Pitch.

#### 4.2.3. Implementación del controlador.

Para posicionar el dron en el centro del área minada se tomó como setpoint el centro de la imagen y de error en  $x$  e  $y$  las distancias existentes entre el centro del plano al centro de la imagen como se observa en la Figura 75.



**Figura 75.** Error de posición del centro de la zona minada con respecto al centro.

Se ha implementado dos controladores digitales PI de la ecuación (9) reemplazando las constantes calculadas, donde tiene un intervalo discreto

$$m(k) = m(k - 1) + K_c(e(k) - e(k - 1)) + K_i\tau_s e(k) \quad (21)$$

El error  $e(x)$  es para el controlador de Roll y el  $e(y)$  para el de Pitch, el algoritmo de control se describe a continuación:

---

**Algoritmo 1.1. Algoritmo del controlador implementado**

---

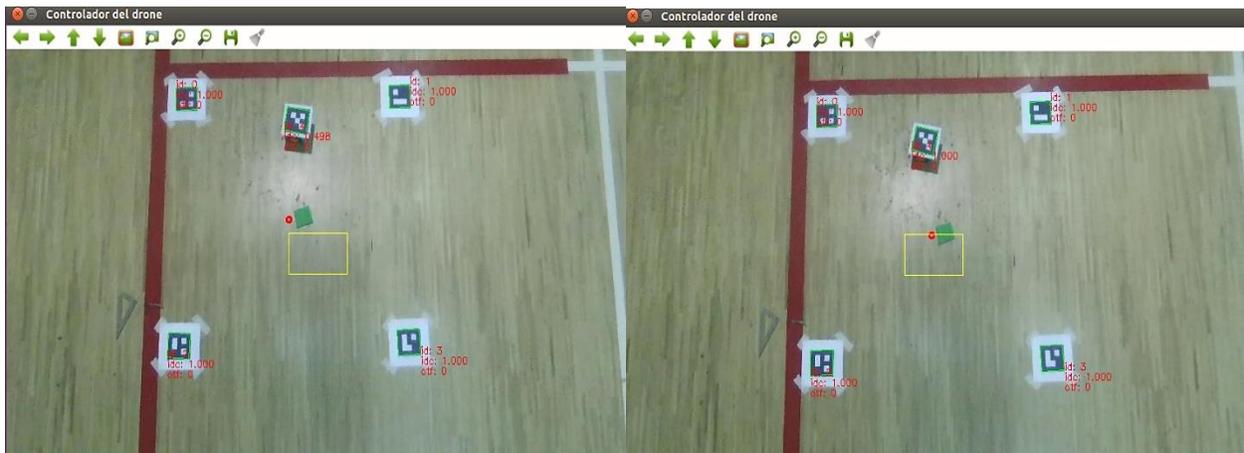
```

if  $t = 0$ 
     $Error\_anterior = Error$ 
if  $t > 0$ 
     $ErrorI = Error - Error\_anterior$ 
     $TermI = ErrorI * Ki * T$ 
     $out = Error * Kp + TermI$ 
     $Error\_anterior = Error$ 
end

```

---

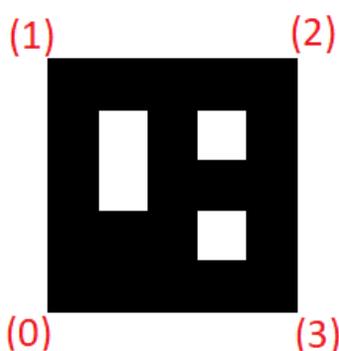
En la Figura 76 se puede visualizar en la izquierda el inicio de la posición del dron marcado con el punto rojo y en la parte derecha ya una vez el controlador ha llegado al setpoint representado por el recuadro de color amarillo se quedará de forma estable y permanente en esa posición.



**Figura 76.** Controlador del dron en funcionamiento.

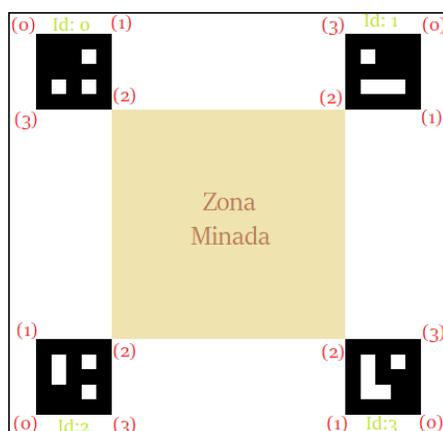
### 4.3. Sistema de Percepción del Entorno de Trabajo

Tomando en cuenta que las perturbaciones sobre el dron pueden afectar la estabilidad del video y cambiar la perspectiva cenital deseada, se definió el plano de trabajo o zona minada con marcadores en cada esquina del cuadrilátero escogido. Mediante el trabajo realizado en la sección 4.1 para detectar los marcadores se obtuvo las esquinas de los mismos, enumerándolas del 0 al 3 en cada marcador como se muestra en la Figura 77.



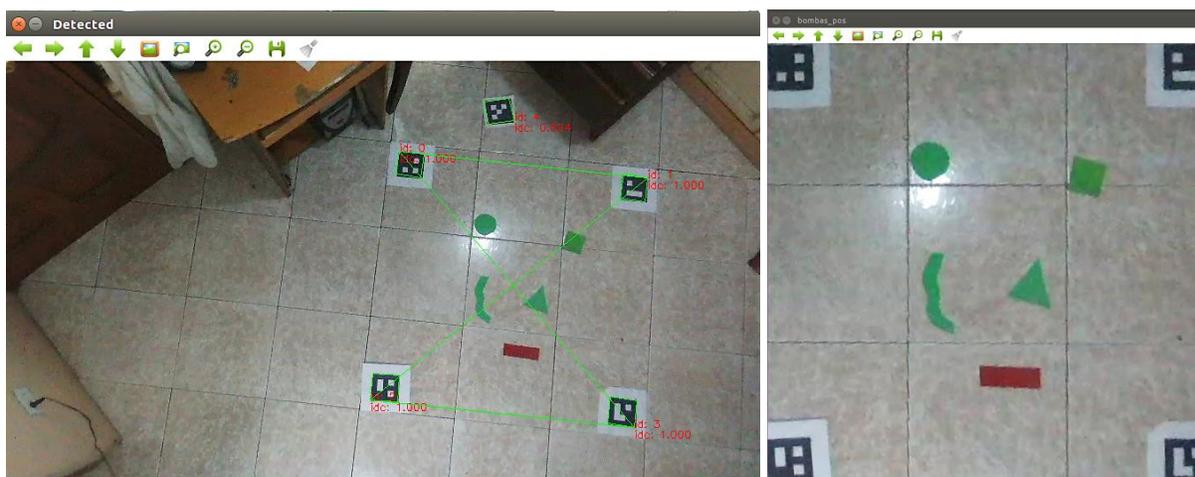
**Figura 77.** Numeración de las esquinas del marcador.

Se escogieron los marcadores codificados del 0 al 3 para ser colocados en las esquinas del plano en la posición que se muestra en la Figura 78, asegurando que las esquinas número 0 de cada marcador estén en los vértices del cuadrilátero.



**Figura 78.** Posición de los marcadores para delimitar el plano de trabajo.

Con la matriz de transformación de perspectiva y las posiciones de las esquinas número 0 de los marcadores se obtuvo la matriz de transformación para pasar el plano a uno con perspectiva cenital de 700x700 píxeles, mediante la matriz obtenida se realizó el corte del plano con el proceso descrito en la Figura 54. El resultado del proceso se puede observar en la Figura 79 con el cual ya se puede proceder a trabajar en la planificación de trayectoria.



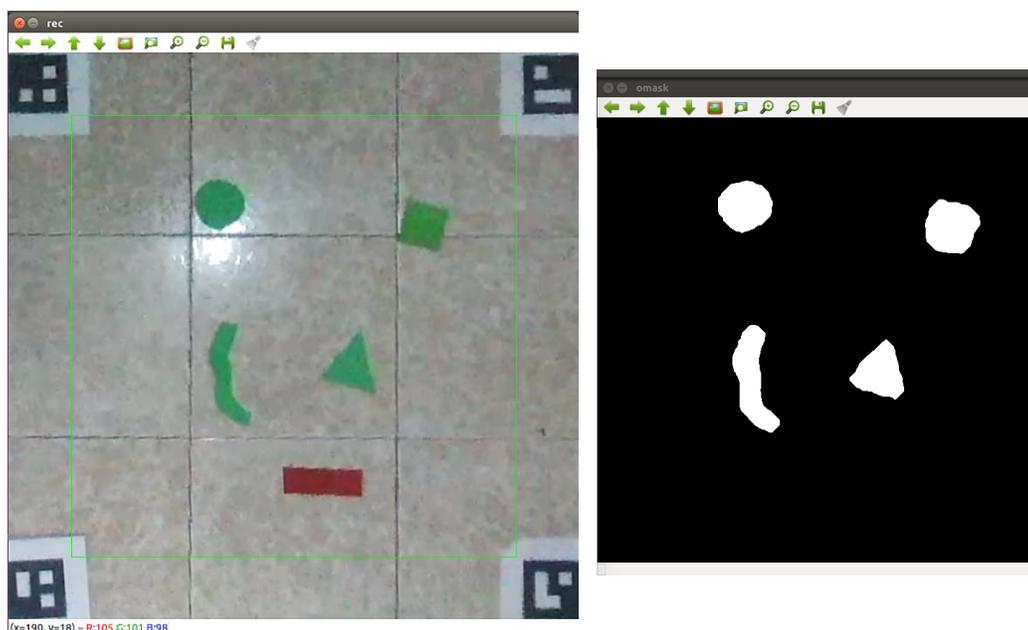
*Figura 79.* Obtención del plano de trabajo.

#### 4.4. Sistema de Detección de Obstáculos

A partir del plano de trabajo obtenido en la sección anterior se procederá a encontrar las zonas en las que puede transitar el robot a lo largo de su recorrido, por lo que es necesario determinar los obstáculos que estén en el espacio minado. En el presente proyecto se contempla trabajar solo con obstáculos no dinámicos.

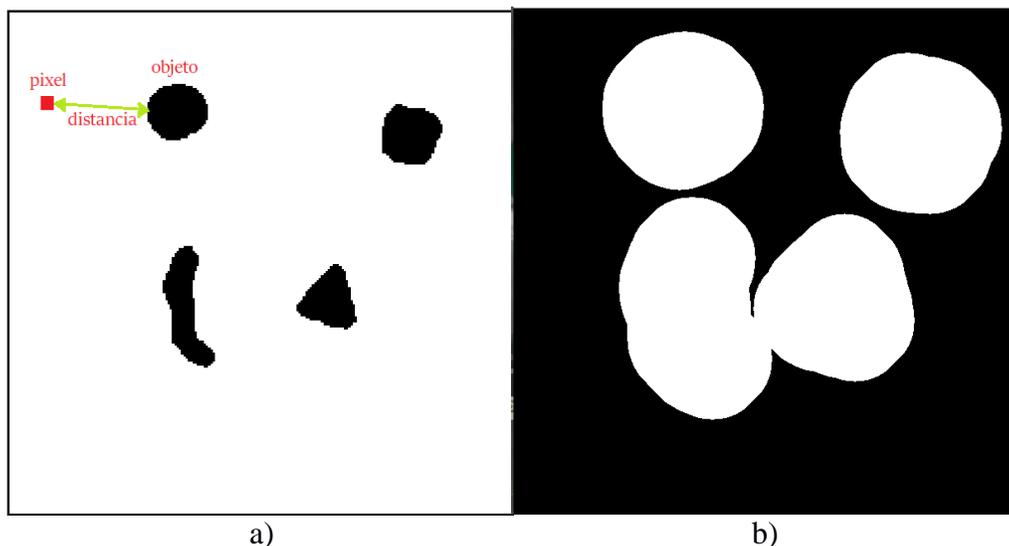
Previamente se debe definir los límites de color que se van a detectar en dos vectores como el más bajo y alto, en este caso se escogió filtrar solo la gama de colores verdes en el espacio HSV (Hue, saturation, value), la función a utilizar es `inRange` que comprueba si los elementos de la matriz de entrada están dentro de los límites escogidos, si se cumple los cambiará por 255 (blanco) caso contrario serán 0 (negro). El espacio válido para trabajar con este algoritmo es el dispuesto por las esquinas

número 2 de los marcadores representados en la Figura 78. El resultado del algoritmo se puede visualizar en la Figura 80.



**Figura 80.** Binarización del espacio de trabajo.

Como se puede apreciar el resultado obtenido del algoritmo presenta en color blanco los obstáculos aproximándose a las formas reales de los mismos y se ha ignorado el de color rojo, el espacio en negro es el permitido para que el robot pueda recorrer, pero hay que tener en cuenta que se debe dejar un espacio para que el robot no choque con los obstáculos, debido a esto se calculó la distancia precisa de cada pixel blanco de la imagen binaria (Figura 81-a) al pixel negro más cercano con la función `distanceTransform` y se validó las distancias, si esta es menor o igual que la mitad del tamaño del robot el pixel es 1 caso contrario será 0, al terminar este proceso se le sumo el resultado de la Figura 80, obteniendo el espacio final por el que recorrerá el robot como se observa en la Figura 81-b.



**Figura 81.** Determinación del espacio de trabajo válido.

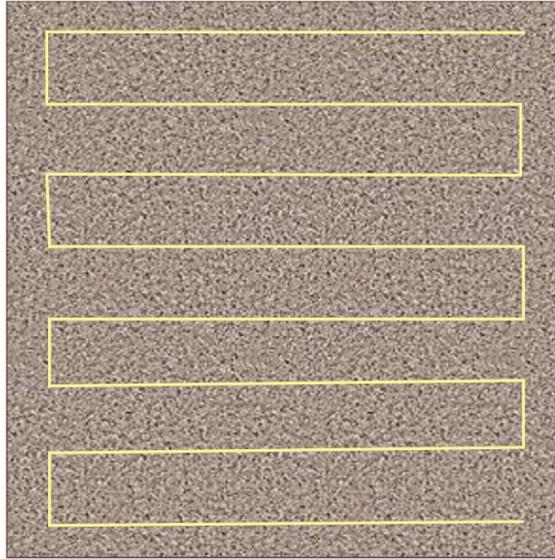
#### 4.5. Sistema de Planificación de Trayectoria del Robot Móvil

Basándose en el estudio realizado por (Colliaux & Hanappe, 2017) en el que se realiza una máquina CNC con azada giratoria como actuador final en su TCP (Tool Center Point), para efectuar el desplazamiento a lo largo de una trayectoria previamente planificada de tal manera que cubra el suelo de un invernadero y no toque las plantas, a su vez al recorrer el espacio entre ellas deberá realizar las tareas de deshierbe quitando toda la maleza que se genera alrededor. En el estudio se presenta 3 pasos para que se pueda generar de manera adecuada la planificación de trayectoria para cobertura de superficies, los cuales se describen a continuación:

- Detectar regiones del espacio de trabajo ocupadas por plantas de interés.
- Generar una ruta de herramienta que cubra el espacio de trabajo, excepto para las regiones ocupadas por plantas.
- Desplazar la herramienta a través de la ruta generada.

La solución más corta al problema de cobertura de superficie bidimensional en un dominio convexo sin obstáculos es un camino simple en zig-zag, que también es conocido como

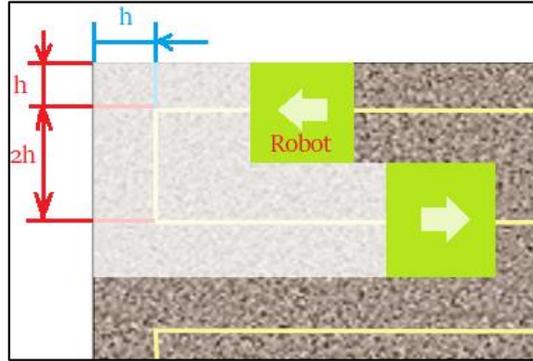
Boustrophedon, por ejemplo, el espacio que se visualiza en la Figura 82 se ha recorrido de forma total con método.



**Figura 82.** Resultado del algoritmo de Boustrophedon para cobertura de superficies.

La trama que define la trayectoria debe cumplir con dos reglas principales las cuales se describen a continuación:

- El espacio desde el borde del plano de trabajo hasta la línea de la trayectoria planificada debe ser igual a la mitad del radio del actuador final del robot o en el caso del presente proyecto la mitad del ancho de la plataforma móvil, esta distancia se simboliza con la letra “h” como se visualiza en la Figura 83.
- La distancia de separación entre los reglones del zig-zag representado como “2h” en la Figura 83 debe ser igual al diámetro de la herramienta del robot o el ancho en el caso de la plataforma móvil.



**Figura 83.** Descripción de las medidas de la trama de la planificación de trayectoria.

La aplicación de las dos reglas anteriormente descritas ofrece una planificación de trayectoria para cobertura de superficies simple pero muy efectiva, en la Figura 83 se visualiza en color blanco transparente un pequeño recorrido simulado del robot que al seguir la línea superior de color amarillo cubre la mitad del espacio del segundo reglón, luego al recorrer la segunda línea cubre la parte faltante.

A continuación, en el Algoritmo 1.2 se presenta el procedimiento para generar la cobertura de superficie en forma de zig-zag de una zona determinada sin tener en cuenta obstáculos.

---

**Algoritmo 1.2. Algoritmo de Boustrophedon implementado**

---

<b>Boustrophedon:</b>	<i># ts: Ancho de la plataforma móvil</i>
$x0 = ts / 2$	<i># Abscisa de inicio para la planificación</i>
$y0 = ts/2$	<i># Ordenada de inicio para la planificación</i>
$Ws = [alto, ancho]$	<i># Dimensiones del espacio de trabajo</i>
$x = [x0]$	<i># Arreglo para acumular las abscisas de la planificación</i>
$y = [y0]$	<i># Arreglo para acumular las ordenadas de la planificación</i>
$Ns = Ws[0]/ts$	<i># Numero de reglones de la planificación</i>
$dx1 = x0$	<i># Abscisa margen derecho del plano</i>
$dx2 = x0 - Ws[1]$	<i># Abscisa margen izquierdo del plano</i>
<b>for</b> $i = 1$ <b>to</b> $Ns$ <b>do:</b>	
$dy = (i-1)*ts + ts/2$	
<b>while</b> $i \text{ MOD } 2 \neq 0$ <b>and</b> $dx1 >= (x0 - Ws[1])$ :	
$dx1 = dx1 - 1$	
$x = x \cup dx1$	
$y = y \cup ((i-1)*ts + ts/2)$	
<b>while</b> $i \text{ MOD } 2 == 0$ <b>and</b> $dx2 <= x0$ :	
$dx2 = dx2 + 1$	
$x = x \cup dx2$	

---

---

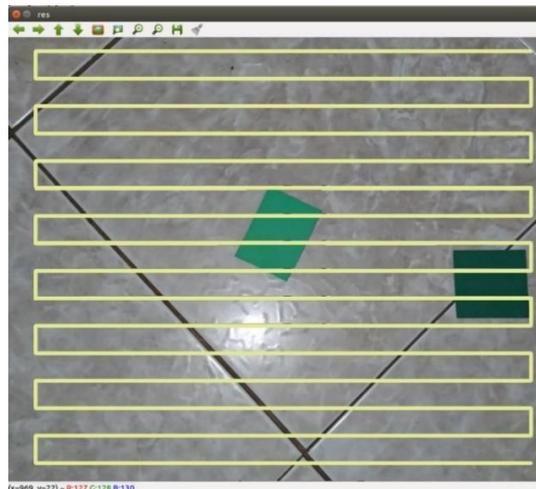
```

    y= y U ((i-1)*ts + ts/2)
dx1 = x0
dx2 = x0 - Ws[1]
while i MOD 2  $\neq$  0 and ((i-1)*ts + ts/2) <= dy <= (i*ts + ts/2):
    dy=dy+1
    x = x U dx2
    y= y U ((i-1)*ts + ts/2+dy)
while i MOD 2 == 0 and ((i-1)*ts + ts/2) <= dy <= (i*ts + ts/2):
    dy=dy+1
    x = x U dx1
    y= y U ((i-1)*ts + ts/2+dy)
return [x,y]

```

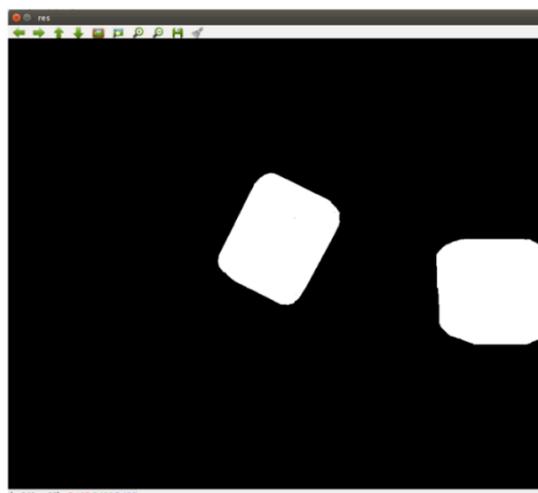
---

Es recomendable que el plano de trabajo tenga un alto múltiplo del ancho del robot para que el reglón final sea de la misma dimensión de los anteriores y se pueda recorrer completamente la superficie. El resultado del algoritmo de cobertura de superficies en zig-zag para un dominio convexo sin tomar en cuenta obstáculos se puede observar en la Figura 84.



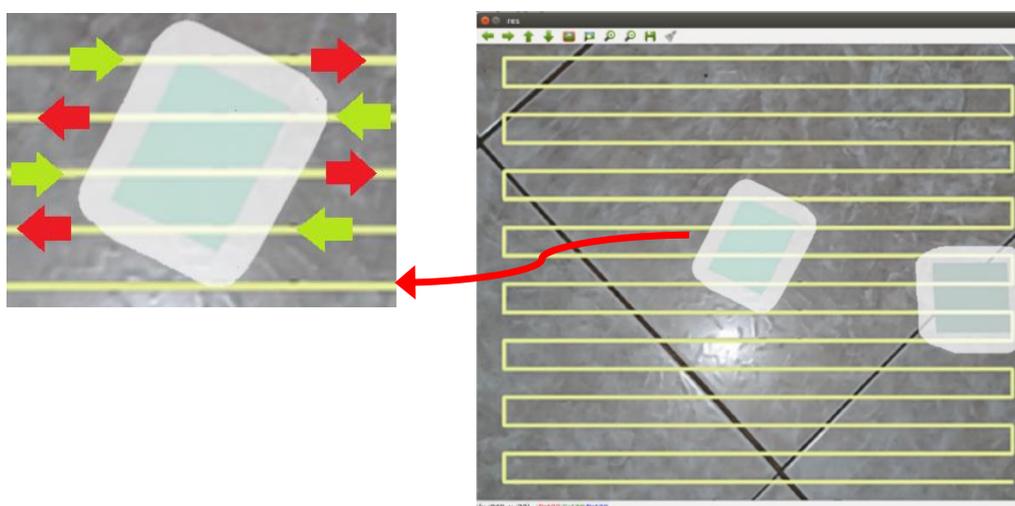
**Figura 84.** Resultado algoritmo de Boustrophedon.

Para solventar el problema de esquivar los obstáculos que se encuentran en el área de trabajo se empieza utilizando el algoritmo descrito en la sección 4.4 obteniendo la máscara de detección de obstáculos basada en color como se visualiza en la Figura 85.



**Figura 85.** Máscara basada en color.

Con el objetivo de visualizar de una mejor manera el procedimiento a realizar se ha sobrepuesto la máscara encontrada con la trayectoria planificada sin obstáculos presentado el resultado en la Figura 86. En cada porción de la planificación de Boustrophedon que se cruce con una región prohibida, se identifican los puntos de entrada y salida como se observa en el recuadro de la derecha donde las flechas verdes simbolizan el ingreso a una zona prohibida y las rojas la salida de la misma.



**Figura 86.** Superposición de máscara y planificación de Boustrophedon.

Una vez se ha identificado las parejas de puntos de entrada y salida que cruzan los obstáculos se procede a escoger la ruta más corta a lo largo del borde la zona prohibida que conecta esos dos puntos y se la sustituye por la línea recta del Boustrophedon. Por ejemplo, en la Figura 87 se observa una pareja de puntos con dos posibles rutas a seguir para evadir el obstáculo, visualmente se puede determinar que el camino 2 es el más corto para unir los dos puntos, este proceso se lo realiza para todas las parejas de puntos encontradas.

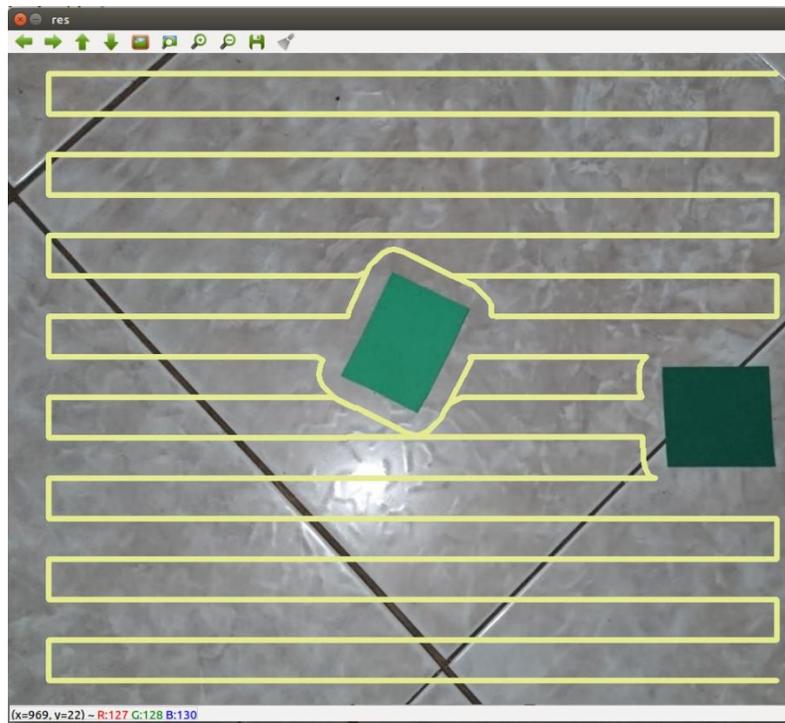


**Figura 87.** Ejemplo del algoritmo de evasión de obstáculos.

A continuación, se presentan los pasos a seguir para obtener el algoritmo de Boustrophedon modificado para evadir obstáculos.

- Encontrar la máscara basada en color del área de trabajo.
- Realizar la planificación de Boustrophedon normal.
- Encontrar puntos de intersección con su entrada y salida entre los bordes de la máscara y el Boustrophedon.
- Determinar la ruta más corta a lo largo del borde de la máscara de objetos para unir cada pareja de puntos.
- Cambiar la línea recta del Boustrophedon normal por las rutas más cortas seleccionadas de cada zona prohibida.

En resultado de aplicar el Boustrophedon modificado se observa en la Figura 88.



**Figura 88.** Resultado del algoritmo Boustrophedon modificado para evadir obstáculos.

#### **4.6. Sistema de Control del Seguimiento de la Trayectoria del Robot Móvil**

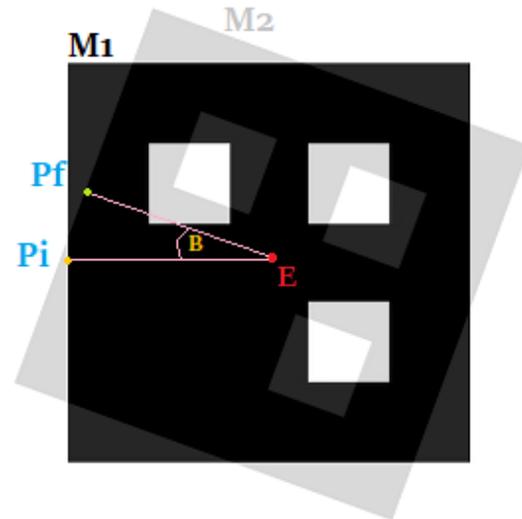
Con el objetivo de seguir con el robot móvil la trayectoria planificada en la sección 4.5 sin ningún sensor a bordo de la plataforma se propone realizar un control servovisual que permita mantener la ruta establecida, para lo cual esta sección se describen los algoritmos necesarios para el sistema.

##### **4.6.1. Estimación del modelo de movimiento del robot.**

Para estimar el movimiento angular del robot móvil se registra un video durante el vuelo del drone, mismo que se ubicará en una sola posición, durante la grabación se envía un tren de pulsos con velocidades permisibles a cada rueda del robot móvil en la escala de  $[-1$  a  $1]$  pero con pulsos intercalados entre las dos señales para poder lograr el giro horario y antihorario en su propio eje.

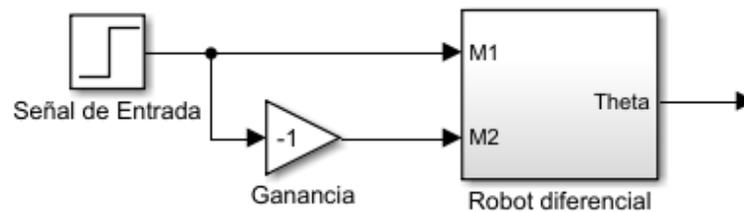
Una vez se tenga el video se analizará los fotogramas consecutivos calculando el ángulo  $\beta$  que se ha girado el robot como se representa en la Figura 89.

- **M1:** Marcador posición inicial frame 1
- **M2:** Marcador posición final frame 2
- **Pi:** Punto de inicio
- **Pf:** Punto final
- **B:** Ángulo generado entre las posiciones del marcador
- **E:** Punto eje de giro



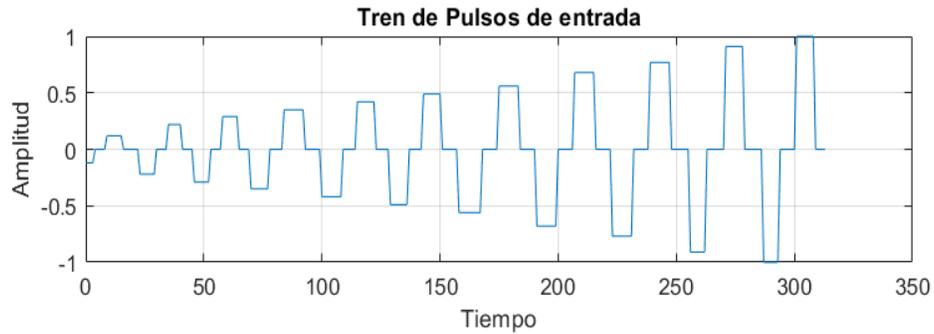
**Figura 89.** Ejemplo de variación del ángulo de giro del robot.

La señal de entrada que se envía al robot móvil para la identificación de la planta se la utilizó para la puesta en marcha de los dos motores, pero intercalando los signos de la misma mediante el diagrama de lazo abierto presentado en la Figura 90, consiguiendo así que el robot gire en su propio eje.



**Figura 90.** Diagrama de prueba para el modelo del robot.

La señal de prueba de la planta se presenta en la Figura 91.



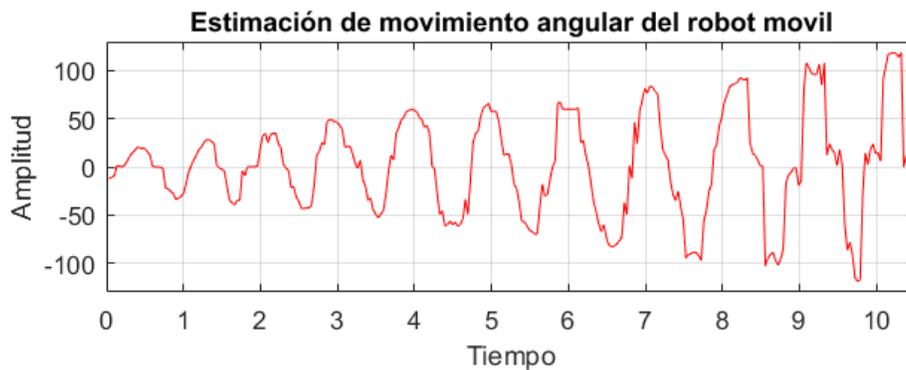
**Figura 91.** Tren de pulsos de entrada para el control del ángulo

El ángulo formado entre los puntos  $P_i$ ,  $P_f$  y  $E$  de la Figura 89 se calcula con la ecuación (22) cuando el movimiento es horario, caso contrario se ocupará la ecuación (23), si el argumento de estas dos ecuaciones es cero significa que los vectores son paralelos es decir que no hubo variación de movimiento por lo que se condiciona el resultado de  $\beta=0$  para este caso.

$$\beta = \cos^{-1} \left( \frac{\overline{EP_f} \cdot \overline{EP_l}}{|\overline{EP_f}| * |\overline{EP_l}|} \right) \quad (22)$$

$$\beta = -\cos^{-1} \left( \frac{\overline{EP_f} \cdot \overline{EP_l}}{|\overline{EP_f}| * |\overline{EP_l}|} \right) \quad (23)$$

Una vez realizado el proceso entre todos los fotogramas consecutivos para determinar la variación de ángulo correspondiente se guarda los valores en cada instante de tiempo obteniendo la señal de la Figura 92.



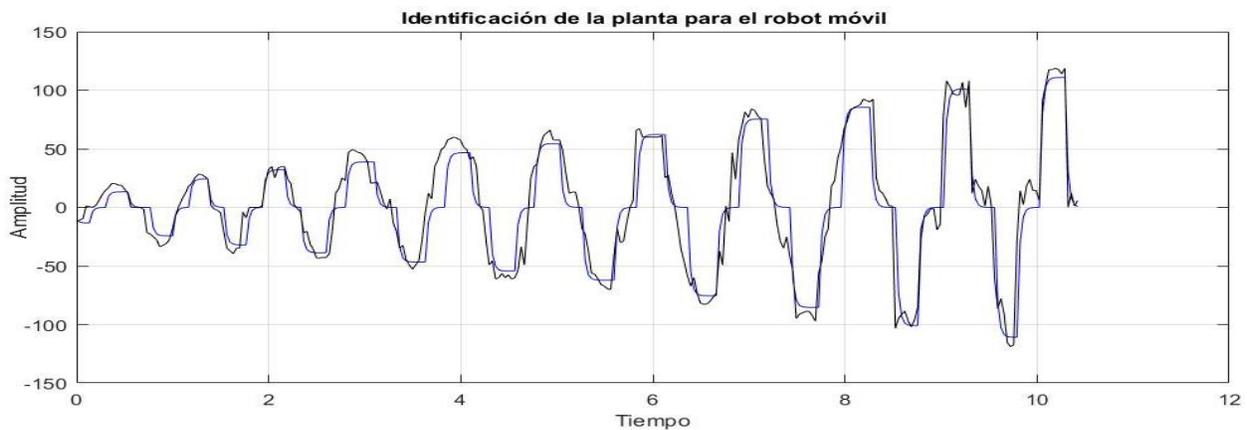
**Figura 92.** Comportamiento del movimiento del robot móvil

La estimación de la planta se realizó mediante la herramienta de Matlab “ident”, se asignó como parámetro de entrada la señal de prueba de la Figura 91 y en la salida los desplazamientos de ángulo obtenidos, el tiempo de muestreo se tomó de acuerdo con la frecuencia de envío de imágenes de la cámara del drone la cual es de 30Hz y se obtuvo la función de transferencia de la ecuación (24).

$$G(s) = \frac{K_p}{1 + \tau_p s} = \frac{A}{B + s} \quad (24)$$

Donde el valor de la ganancia de  $K_p = 110.96$ , y el tiempo de  $T_p = 0.025547$ .

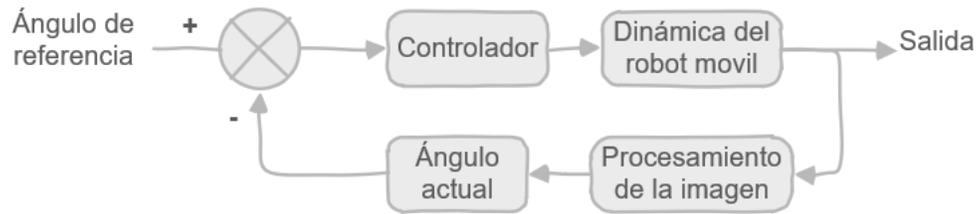
A continuación, en la Figura 93 se muestran una comparación del modelo estimado con los datos medidos de la planta.



**Figura 93.** Identificación de la planta del robot móvil

#### 4.6.2. Diseño del controlador

Para el posicionamiento angular del robot móvil se implementó un controlador que permita direccionar de forma adecuada la plataforma hacia los puntos de la trayectoria planificada y seguir esta correctamente, en la Figura 94 se observa el esquema de control establecido para el sistema.



**Figura 94.** Esquema de control para el posicionamiento angular del robot móvil

Mediante el método de ubicación de polos se procede a diseñar un controlador para el modelo obtenido de la planta, se hace uso de la forma general de un controlador PID presentado en la ecuación (25).

$$C(s) = \frac{K_d s^2 + K_c s + K_i}{s} \quad (25)$$

A partir del lazo de control mostrado en la Figura 72 se obtiene la función de transferencia en el lazo cerrado representado por las ecuaciones (26) y (27):

$$H(s) = \frac{G(s) * C(s)}{1 + G(s) * C(s)} \quad (26)$$

$$H(s) = \frac{A(K_d s^2 + K_c s + K_i)}{s^2(1 + AK_d) + s(B + AK_c) + AK_i} \quad (27)$$

Como parámetros de diseño se propone un tiempo de establecimiento menor a 2 segundos y una sobreelongación del 0%, posicionando un polo en  $p_s = -0.5$ . Los polinomios característicos correspondientes son:

$$\Delta_1(s) = s^2(1 + AK_d) + s(AK_c) + AK_i \quad (28)$$

$$\Delta_2(s) = (s + p_s)^2 = p_0 s^2 + p_1 s + p_2 \quad (29)$$

La solución matricial del polinomio se resuelve como PC=D donde:

$$\begin{aligned}
 s^2: & 1 + AK_d = p_0 \\
 s^1: & B + AK_c = p_1 \\
 s^0 & AK_i = p_2
 \end{aligned} \tag{30}$$

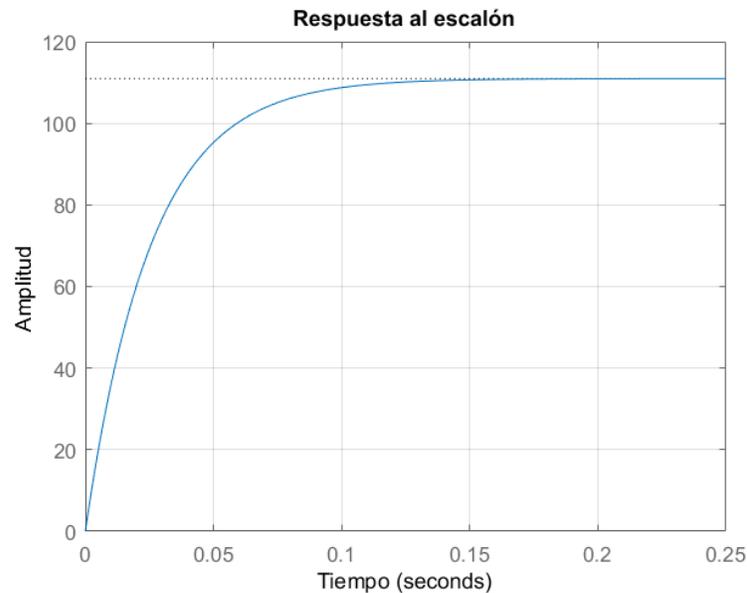
Las matrices P, C y D son:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{bmatrix} \cdot \begin{bmatrix} K_d \\ K_c \\ K_i \end{bmatrix} = \begin{bmatrix} p_0 - 1 \\ p_1 - B \\ p_2 \end{bmatrix} \tag{31}$$

Se reemplaza los valores con el polo  $p_s = -0.5$  y los valores  $A = 57.43$  y  $B = 71.43$  de la planta desde la ecuación (24), obteniendo las ganancias respectivas:

$$K_c = 0.278 \quad K_i = 1.939 \quad K_d = 0.0004$$

En base a la respuesta al impulso unitario en la Figura 95 se comprueba el desempeño del controlador de acuerdo a los parámetros impuestos.



**Figura 95.** Respuesta al impulso unitario del controlador de giro.

#### 4.6.3. Cálculo de la dirección y ángulo de giro del robot.

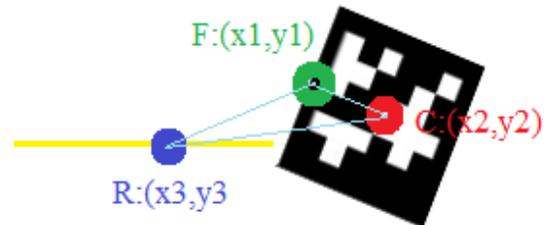
Para la implementación del controlador se necesita encontrar el ángulo al cual el robot móvil debe orientarse para esto se identificaron tres puntos, dos de estos se obtienen del frente y centro

del robot representados en la Figura 96 como F y C respectivamente, el tercer punto viene dado por la ruta que se debe seguir, formando el triángulo escaleno que se observa en la Figura 96.

**F:** Punto frente del robot móvil

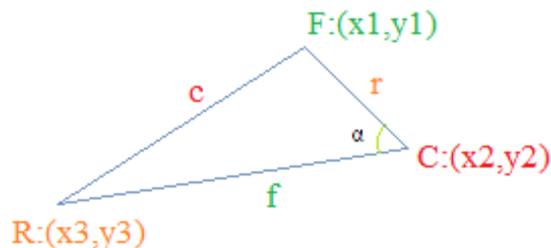
**C:** Punto centro del robot móvil

**R:** Punto de la ruta donde debe dirigirse el robot.



**Figura 96.** Coordenadas para el seguimiento de la trayectoria para el control servovisual

Aislando el triángulo escaleno mostrado en la Figura 97 para poder trabajar de una mejor manera se procede a calcular el ángulo  $\alpha$  que deberá girar robot para posicionarse en la dirección del punto que pertenece a la ruta planificada, independientemente del valor que tenga el ángulo calculado se debe tener en cuenta que el giro puede ser horario o antihorario para posicionarse.



**Figura 97.** Triángulo escaleno para el cálculo del ángulo  $\alpha$  de giro del robot móvil

Mediante la ley de cosenos se calcula el ángulo  $\alpha$ , obteniendo la ecuación (32).

$$\alpha = \arccos \left( \frac{f^2 + r^2 - c^2}{2 \cdot f \cdot r} \right) \quad (32)$$

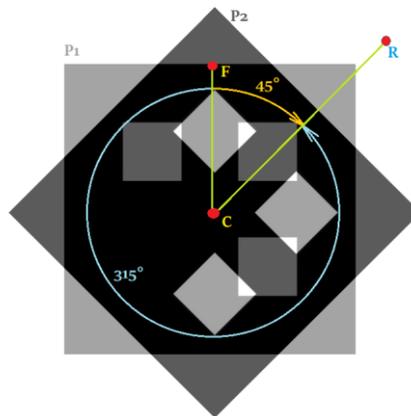
Donde f, r y c representan las distancias entre los puntos F, R y C, mismas que se calculan mediante las ecuaciones (33),(34) y (35):

$$f = \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2} \quad (33)$$

$$r = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (34)$$

$$c = \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2} \quad (35)$$

El cálculo del ángulo con la ecuación (32) presenta una ambigüedad, por ejemplo, el resultado de  $\arccos(\sqrt{2}/2)$  es  $45^\circ$ , pero también puede ser igual a  $315^\circ$ , esto sucede porque el coseno es una función par es decir el resultado del primer y cuarto cuadrante son iguales, lo mismo sucede para el segundo y tercer cuadrante, esto es un problema debido a que el robot va a tener dos rutas a seguir, en el ejemplo de la Figura 98 se requiere posicionar el frente del robot “F” hacia el punto de la trayectoria “R”, moviéndose de la posición 1 “P1” a la posición 2 “P2”, como se observa hay dos posibilidades para orientar el vector  $\overrightarrow{FC}$  al vector  $\overrightarrow{RC}$  girar  $45^\circ$  en sentido horario o  $315^\circ$  en sentido antihorario, en este punto el robot podrá realizar el objetivo en cualquier caso pero no es eficiente ya que puede tomar la ruta más larga.



**Figura 98.** Direccionamiento del robot.

Para determinar qué sentido de giro debe realizar el robot para orientarse, se procede a calcular el ángulo  $\alpha$  que equivalen a la ecuación (36), por lo que se presenta un problema semejante a la

función de la tangente ya que se obtienen los mismos resultados en el primer y tercer cuadrante, al igual que con el segundo y cuarto cuadrante del círculo trigonométrico. Combinando los resultados de las ecuaciones (32) y (36) se puede obtener precisamente la dirección a la que el robot debe girar sin ambigüedades.

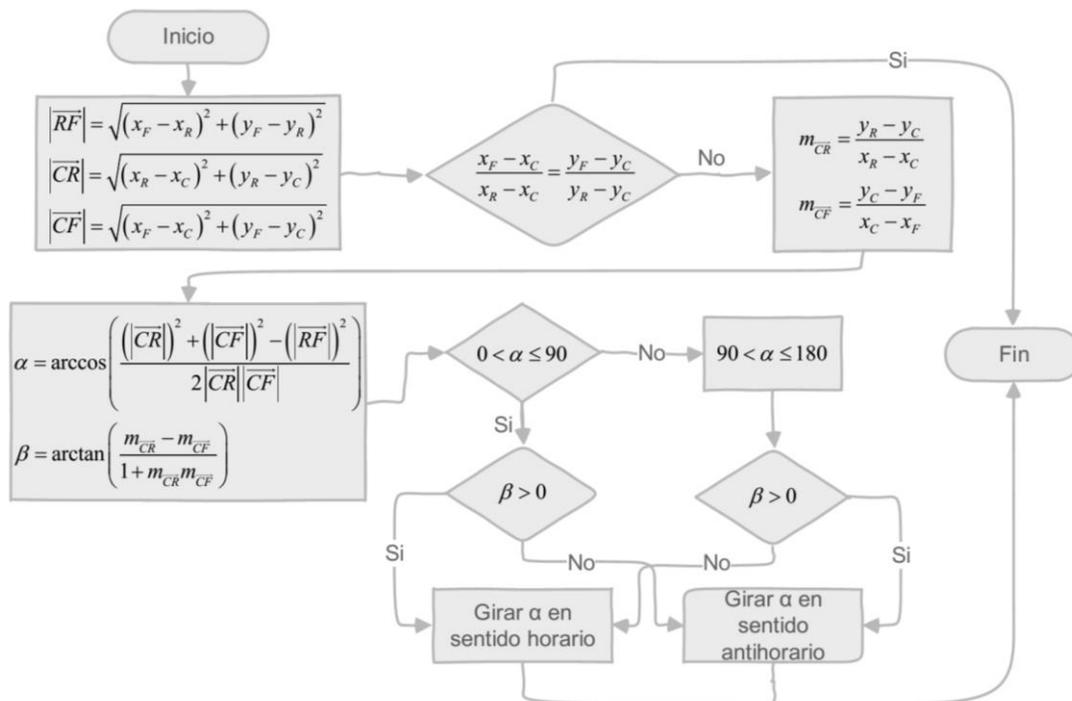
$$\beta = \arctan \left( \frac{m_f - m_r}{1 + m_r \cdot m_f} \right) \quad (36)$$

Las pendientes  $m_r$  y  $m_f$  de la ecuación (36) se determinan mediante las ecuaciones (37) y (38):

$$m_f = \frac{y_3 - y_2}{x_3 - x_2} \quad (37)$$

$$m_r = \frac{y_2 - y_1}{x_2 - x_1} \quad (38)$$

Para un mejor entendimiento del proceso realizado se presenta el diagrama de la Figura 99 para el cálculo del ángulo y determinación del sentido del giro del robot.



**Figura 99.** Diagrama de flujo del ángulo de rotación del robot móvil

#### 4.6.4. Implementación del controlador para posicionamiento angular del robot.

Con los métodos descritos en las secciones 4.6.2 y 4.6.3 se procede a la implementación del controlador de giro del robot con el algoritmo que se describe a continuación:

---

**Algoritmo 1.2. Algoritmo del controlador implementado**

---

```

if  $t = 0$ 
     $Error\_anterior = Error$ 
if  $t > 0$ 
     $ErrorI = Error + Error\_anterior$ 
     $ErrorD = Error - Error\_anterior$ 
     $TermI = ErrorI * Ki * T_s$ 
     $out = Error * K_p + TermI + ErrorD * K_d$ 
     $Error\_anterior = Error$ 
if  $\theta > 0$ 
     $\theta = \frac{out(v_{max}-v_{min})}{K_p Error} + v_{min}$ 
else
     $\theta = -\frac{out(v_{max}-v_{min})}{K_p Error} + v_{min}$ 
end

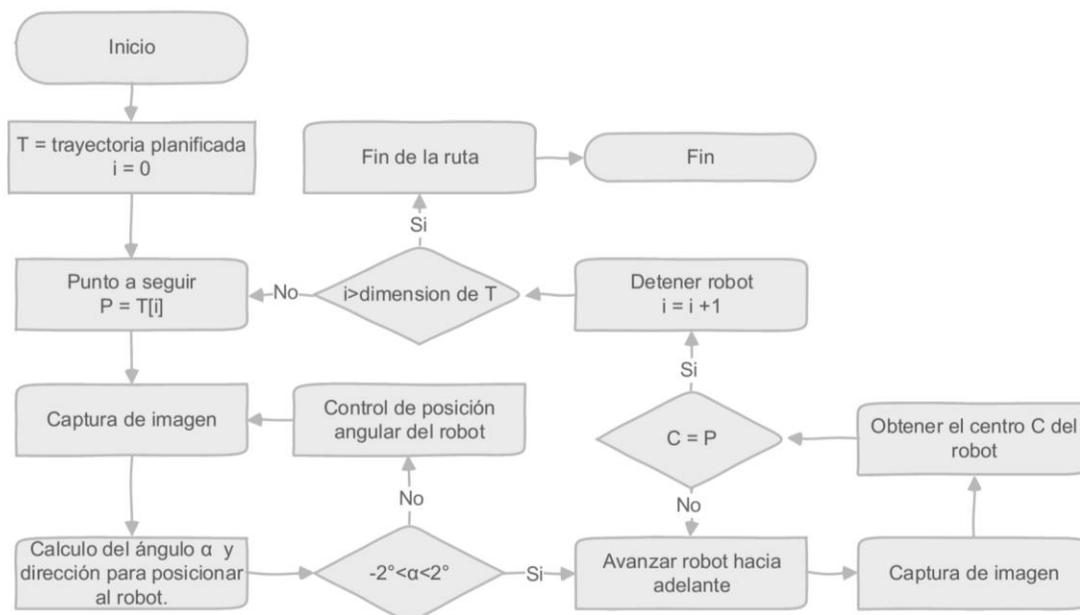
```

---

El “set point” para el controlador es cero ya que es necesario posicionarse en dirección a los puntos de la trayectoria planificada, el error de posición está dado por el cálculo del ángulo con el algoritmo de la sección 4.6.3.

#### 4.6.5. Seguimiento de trayectoria.

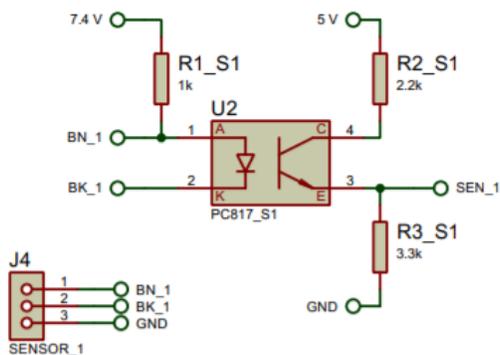
Con el objetivo de seguir la ruta planificada primero se hace uso del controlador de giro para ubicar al robot en la dirección correcta, si este se ha alineado satisfactoriamente con el punto de ruta se procede a enviar el comando hacia adelante para que el robot avance. Una vez el centro del robot se ubique sobre el punto de la trayectoria al cual se orientó, se calculará el nuevo ángulo de giro para el próximo punto a seguir, este bucle se realizará hasta haber llegado al último punto de la trayectoria. El diagrama de flujo referente al seguimiento de la ruta se lo observa en la Figura 100.



**Figura 100.** Diagrama de flujo del control de seguimiento de trayectoria del robot móvil.

#### 4.7. Sistema de Detección de Minas Antipersonales.

Para la detección de las minas antipersonales se utilizó los sensores de proximidad inductivos IJ8A3-2-z/bx con salida tipo interruptor, debido a que la tarjeta de control Node Mcu posee varias entradas digitales y solo una analógica. Estos sensores son capaces de detectar ciertos metales, se recurrió a un optoacoplador para adaptar la salida del sensor de 7.4V a 3.3V debido a que la tarjeta de control trabaja a este nivel de voltaje, el diseño del circuito se presenta en la Figura 101.



**Figura 101.** Circuito para acoplar el sensor de proximidad inductivo

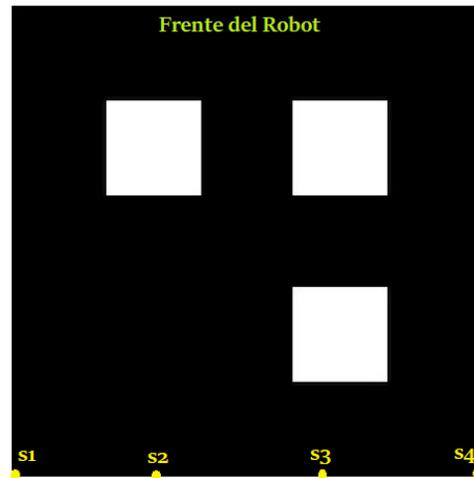
Los sensores estarán ubicados en la parte posterior del robot como se muestra en la Figura 102.

**s1:** Posición sensor 1

**s2:** Posición sensor 2

**s3:** Posición sensor 3

**s4:** Posición sensor 4



**Figura 102.** Posiciones de los sensores en el marcador del robot.

Cuando cualquiera de los sensores se active, es decir se detectó un metal se envían a través de un tópico Booleano los estados de todos ellos a la estación en tierra, donde se establece la posición de cada uno y solo se guardarán los que están activos. Las posiciones de los sensores en el marcador se obtienen mediante el algoritmo de la sección 4.1 que determina las ubicaciones de los puntos “s1” y “s4”, el cálculo de las coordenadas de “s2” y “s3” se estima con las ecuaciones (39) y (40).

$$x_s = \frac{x_{s1} + x_{s4} * r}{1 + r} \quad (39)$$

$$y_s = \frac{y_{s1} + y_{s4} * r}{1 + r} \quad (40)$$

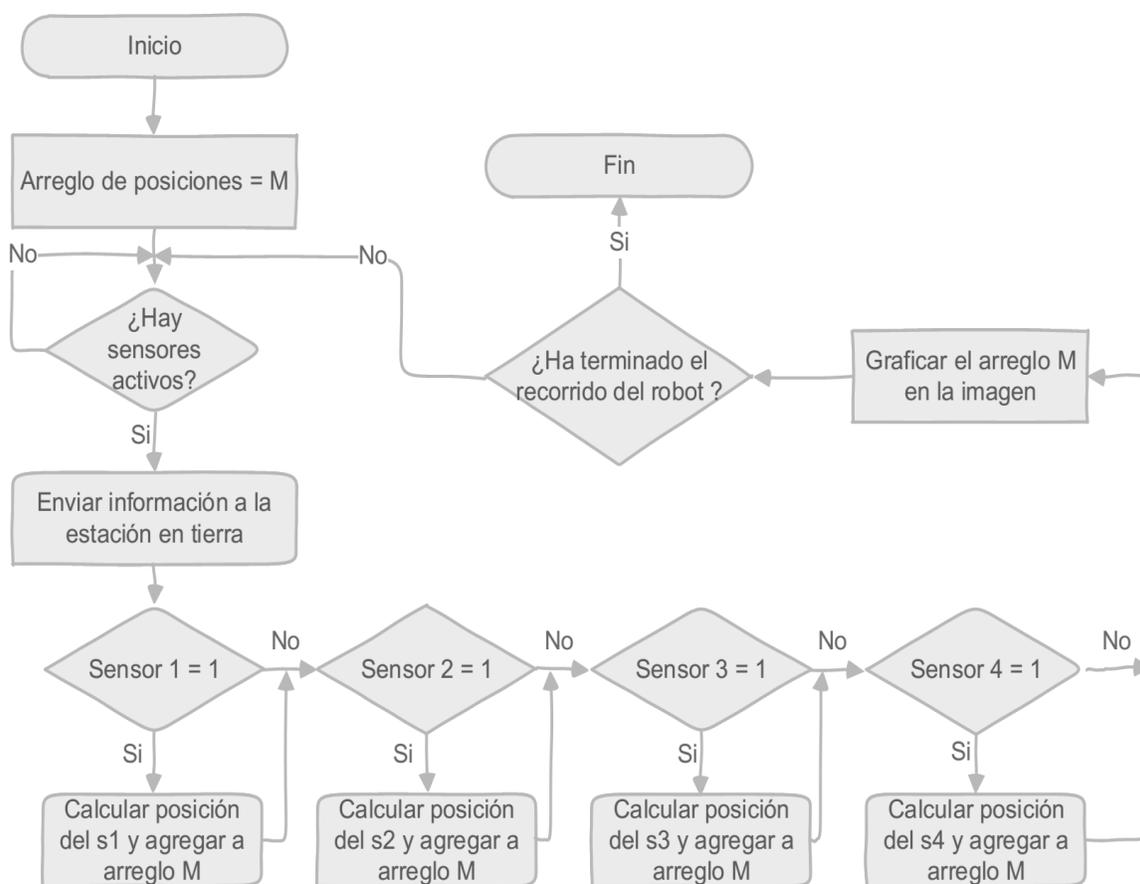
El valor de la razón “r” para el cálculo de las coordenadas de s2 se lo realiza con la ecuación 41.

$$r = \frac{|\overrightarrow{s1s2}|}{|\overrightarrow{s2s4}|} = \frac{\frac{1}{3}}{\frac{2}{3}} = 0.5 \quad (41)$$

En el caso de  $s_3$  el valor de la razón se define por la ecuación  $\zeta$ .

$$r = \frac{|s_1 s_3|}{|s_3 s_4|} = \frac{\frac{2}{3}}{\frac{1}{3}} = 2 \quad (42)$$

Una vez se tengan las coordenadas de los sensores que se activaron se procede a graficar en la imagen del plano de trabajo, el diagrama de flujo de la Figura 103 describe el proceso.



**Figura 103.** Diagrama de flujo del proceso para detectar minas.

## CAPÍTULO V

### EVALUACIÓN, PRUEBAS Y ANÁLISIS DE RESULTADOS

En el presente capítulo se muestran los resultados de la experimentación que se desarrolló para el avance del proyecto en alguna de sus etapas, así como también las pruebas de funcionamiento a las que se sometió el sistema para evaluar su desempeño.

#### 5.1. Experimentación y resultados de Landmarks

##### 5.1.1. Lectura de marcadores

##### 5.1.1.1. Códigos QR

La herramienta zbar de Python es una biblioteca de lectura de códigos de barras y códigos QR. Las imágenes de entrada deben ser matrices 2D de tipo numpy, es decir imágenes en escala de grises, su utilización es fácil no requiere de mucha programación ni memoria para su ejecución, el resultado obtenido se observa en la Figura 104 donde se puede observar enmarcados los códigos del 0 al 4 en color azul y su respectivo identificador en el centro del mismo.



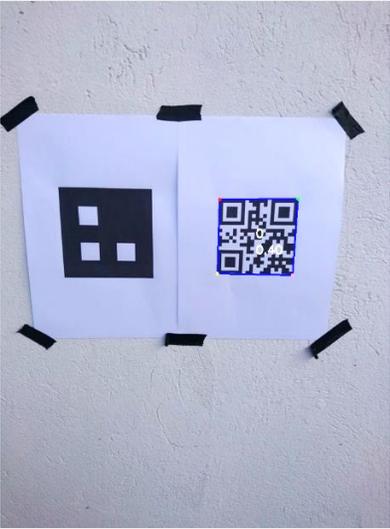
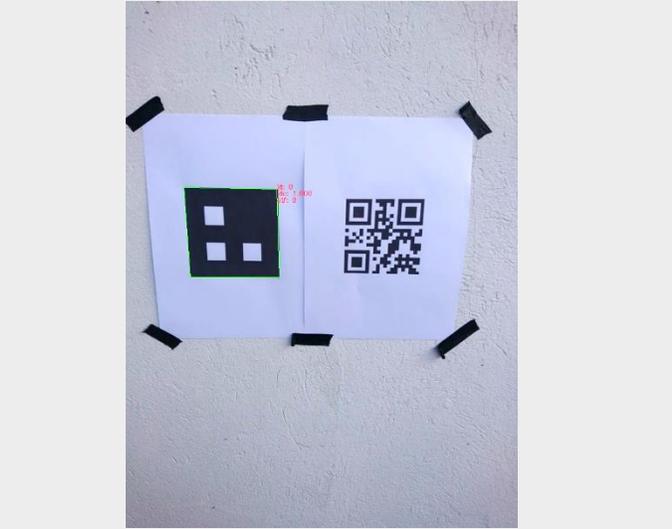
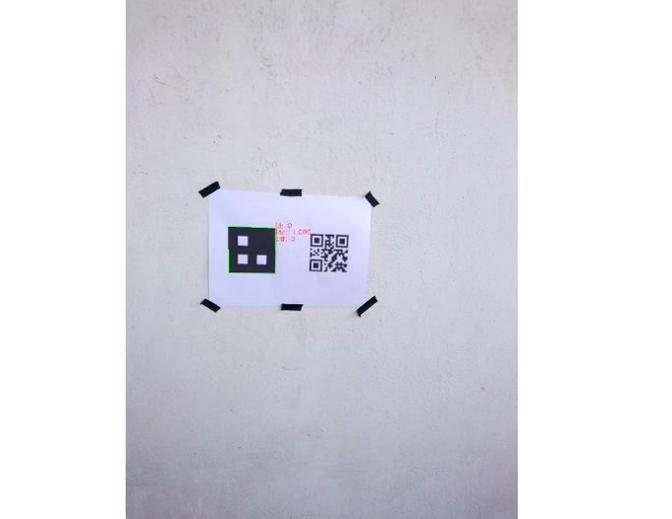
*Figura 104.* Reconocimiento de códigos QR.

### 5.1.1.2. Marcadores ARUCO vs Códigos QR

Con el objetivo de determinar cuál método es mejor en esta sección se analizará la respuesta de los algoritmos expuestos en las secciones 4.1y 5.1.1.1 la con imágenes capturadas a diferentes distancias, en la Tabla 11 se exponen los resultados.

**Tabla 11**

*Comparación de reconocimiento de marcadores Aruco y Códigos QR.*

Distancia	Reconocimiento de códigos QR	Reconocimiento de marcadores Aruco.
0.5m		
1m		

CONTINÚA 



**3m**

No reconocido

**3.5m**

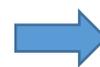
No reconocido

**4m**

No reconocido



CONTINÚA



**4.5m**

No reconocido

**5m**

No reconocido

**5.5m**

No reconocido



CONTINÚA



**6m**

No reconocido



**6.5m**

No reconocido



**7m**

No reconocido



Como se puede observar en la Tabla 12 el algoritmo para reconocimiento de marcadores Aruco desarrollado en la sección 7.1 es más efectivo y robusto en comparación con la librería zbar de Python, con Aruco se reconoció marcadores hasta 7m de distancia de forma estable y para distancias mayores el algoritmo es intermitente mientras que con zbar solo se tiene un alcance de hasta 1.5m, por esta razón para el presente proyecto se decidió utilizar el reconocimiento de marcadores Aruco debido a que el drone estará ubicado a una altura de 2 m a 3.5 m.

**Tabla 12**

*Resumen del reconocimiento de marcadores Aruco y Códigos QR.*

Distancia [m]	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7
<b>Detección Tag Aruco</b>	Sí	Sí												
<b>Detección Código QR</b>	Sí	Sí	Sí	No	No	No								

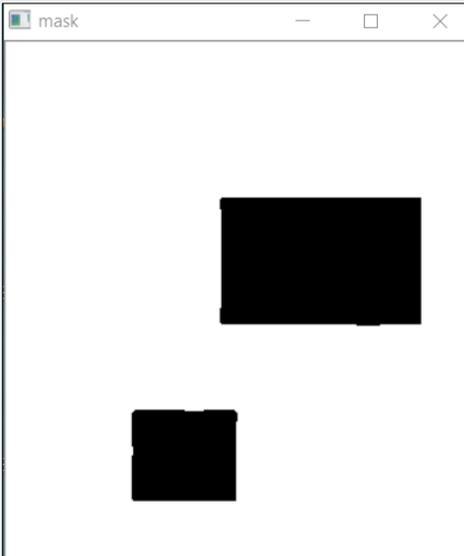
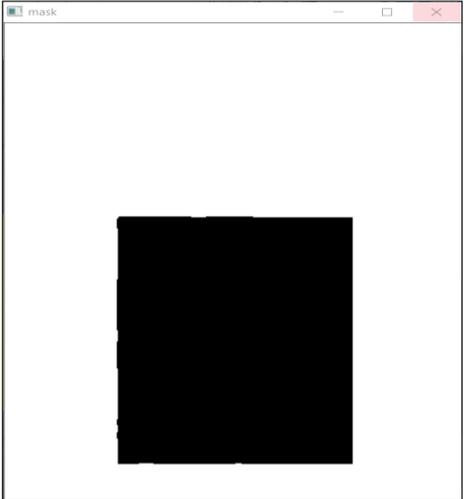
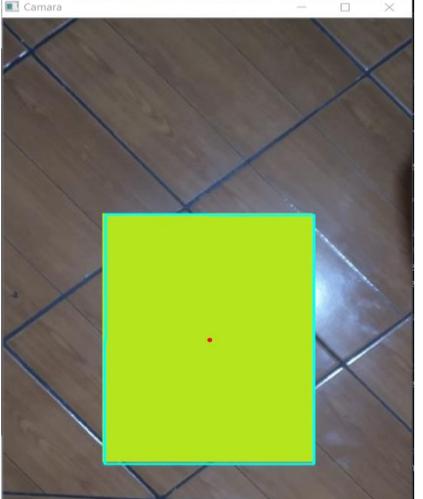
### 5.1.2. Pruebas para detección de obstáculos

El espacio de trabajo puede o no tener la presencia de obstáculos los cuales deben ser tomados en cuenta en la planificación de trayectoria, cabe recalcar que en el presente proyecto solo se trabajará con obstáculos no dinámicos. Para determinar de una manera eficiente la planificación de trayectoria y que el robot no presente inconvenientes en el seguimiento de ruta se evalúa la capacidad de reconocimiento de objetos con dos algoritmos presentados a continuación.

#### 5.1.2.1. Detección basada en color

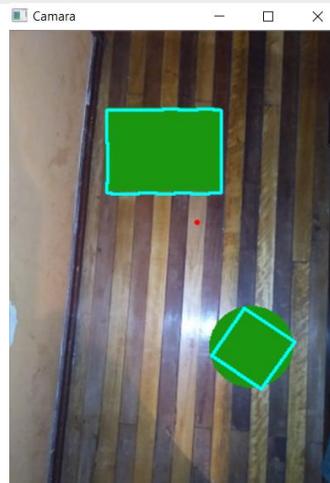
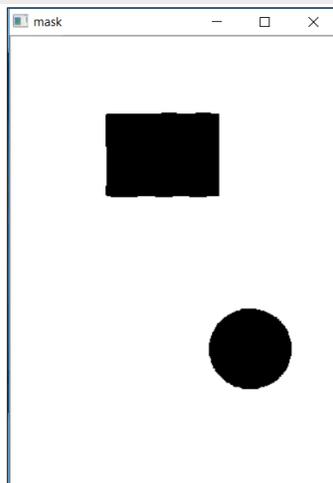
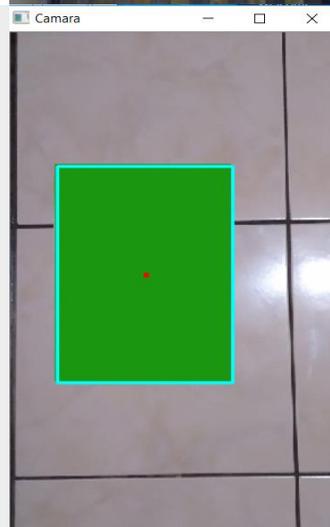
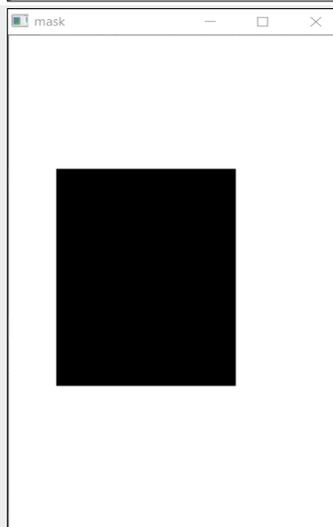
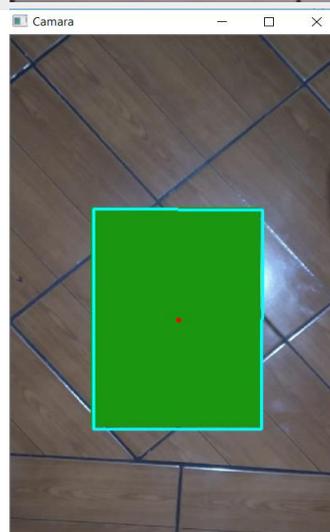
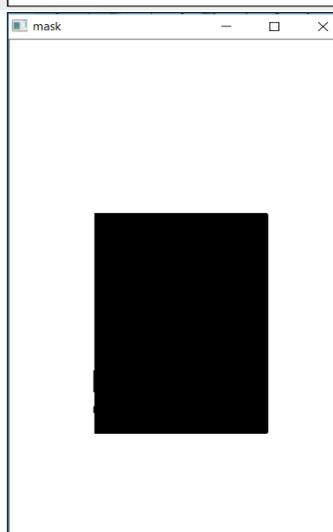
Anteriormente en la sección 4.4 se explicó cómo obtener una máscara basada en color, con el método `inRange` de OpenCv que se usa para filtrar una gama de colores en el espacio HSV en un rango específico, para determinar cuan efectivo es el algoritmo frente a varios tipos de suelos, colores y formas de objetos se presenta los resultados de la Tabla 13.

**Tabla 13**  
*Pruebas experimentales para detectar obstáculos.*

Código HSV en la gama de verdes [181,230,29]		
Escenario	Máscara	Obstáculo Detectado
Piso de baldosa blanca		
Piso de parquet		
Piso de baldosa café		

CONTINÚA



**Código HSV en la gama de verdes [26,150,16]****Piso de parquet****Piso de baldosa blanca****Piso de baldosa café**

CONTINÚA

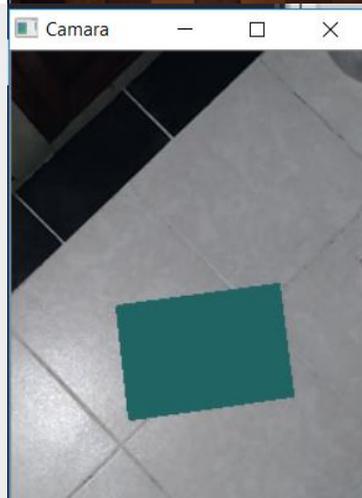


**Código HSV en la gama de verdes [32,100,99]****Piso de parquet**

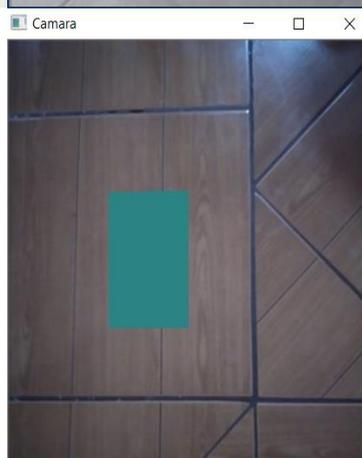
No detecta el obstáculo

**Piso de baldosa blanca**

No detecta el obstáculo

**Piso de baldosa café**

No detecta el obstáculo



Para determinar qué tan efectiva es la selectividad del método para escoger los pixeles con umbral correcto en los límites establecidos se creó 3 pruebas, 2 de ellas con umbrales que se encuentran dentro

del rango y uno que no pertenece. Como se observa en la Tabla 14 el método en los dos primeros umbrales que son correctos reconoció los objetos obteniendo una máscara que describen los bordes de los objetos con todos sus detalles, mientras que el último umbral analizado se observa que no se reconoció ningún objeto esto da a conocer que el método es bastante robusto.

**Tabla 14**

*Resumen de detección de obstáculos mediante el uso del color.*

Escenario	Detección de obstáculos
<b>Gama de verde [181,230,29]</b>	
Piso de parquet	Sí
Piso de baldosa blanca	Sí
Piso de baldosa café	Sí
<b>Gama de verde [26,150,16]</b>	
Piso de parquet	Sí
Piso de baldosa blanca	Sí
Piso de baldosa café	Sí
<b>Gama de verde [32,100,99]</b>	
Piso de parquet	No
Piso de baldosa blanca	No
Piso de baldosa café	No

### 5.1.2.2. Detección basada en morfología

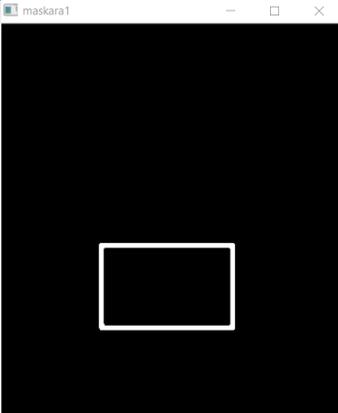
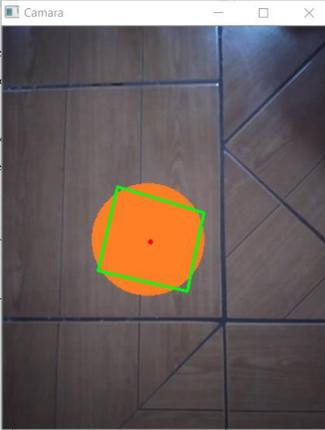
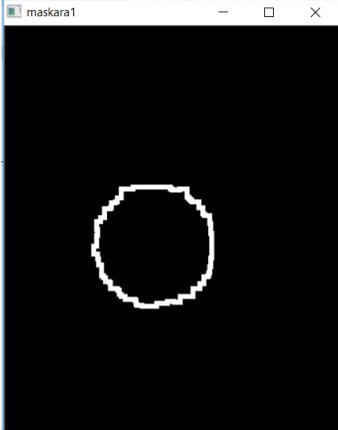
Para determinar la morfología de los objetos en el plano de trabajo se hace uso de la función “findContours” de OpenCV que permite encontrar los  $n$  contornos externos e internos de una imagen binaria, otra característica es que agrupa los contornos por jerarquías tomando al contorno más externo como el de mayor rango. En modo de uso es muy simple solo es necesario pasar como parámetro la imagen en blanco y negro y especificar el tipo de aproximación que se desea realizar las cuales se describen a continuación:

- CV\_CHAIN\_APPROX\_NONE: Guarda todos los puntos de los contornos.
- CV\_CHAIN\_APPROX\_SIMPLE: Considera segmentos horizontales, verticales y diagonales.

Con el fin de mejorar la aproximación de los bordes de los objetos se ocupa el algoritmo de la sección 5.1.2.1 y luego la función “findContours” para tres objetos diferentes, los resultados se presentan en la Tabla 15.

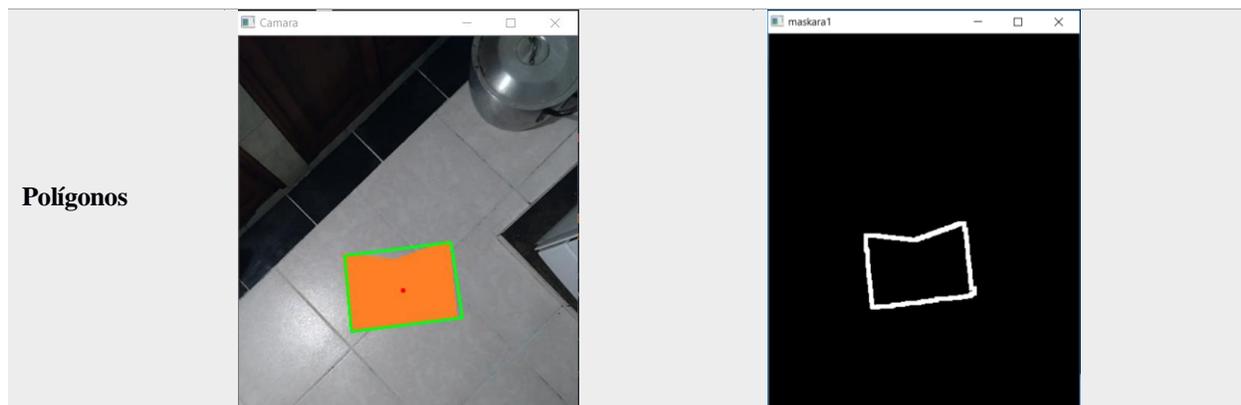
**Tabla 15**

*Pruebas experimentales para detectar obstáculos mediante la morfología.*

Forma	Objeto	Detección de borde
Cuadrilátero		
Círculo		

CONTINÚA





Como se puede observar en las máscaras obtenidas de cada imagen de la Tabla 15 la aproximación del borde es buena pero no refleja todos los rasgos del objeto.

### 5.1.2.3. Detección basada en morfología vs color

Con los resultados obtenidos en las secciones 5.1.2.1 y 5.1.2.2 se establece que el método de detección de obstáculos basado en color resulta muy estable y confiable ya que refleja la mayoría de los rasgos de los objetos y se puede discriminar aquellos que no se deseen mediante los niveles de umbral del objeto a detectar, mientras que al obtener la morfología del objeto mediante una aproximación de borde no refleja con tanta exactitud los rasgos del objeto, es por esta razón que se utilizó el algoritmo basado en color en el presente proyecto.

## 5.2. Experimentación y resultados del sistema

Para valorar el trabajo, se hizo un análisis experimental de pruebas para distintos escenarios con diversas condiciones para obtener resultados en el entorno por el cual se desenvolverá el sistema desarrollado.

Las condiciones más importantes al implementar algoritmos de planificación de trayectoria basados en visión son: la altura de la cámara al plano de trabajo, iluminación y tipo de suelo. Las pruebas de funcionamiento se desarrollaron en 2 escenarios mismos que están ubicados en el Coliseo

Gral. Miguel Iturralde de la ESPE, por lo que se utilizó un espacio cerrado ubicado dentro de las instalaciones de la universidad, en la Tabla 16 se muestran los factores de interés que influyen para ambos escenarios.

**Tabla 16**

*Características de interés en los escenarios de pruebas experimentales.*

Lugar de pruebas	Grado de iluminación	Tipo de iluminación	Intervención de Agentes externos	Color del suelo	Material y tipo de suelo.
<b>Coliseo Gral. Miguel Iturralde (cancha)</b>	Alta Mediana Baja	Artificial	Nula	Café claro	Parquet-Liso
<b>Coliseo Gral. Miguel Iturralde (escenario)</b>	Alta Mediana Baja	Artificial	Nula	Café oscuro	Parquet-Liso

Antes de iniciar con las pruebas de funcionamiento se presenta la secuencia de acciones del sistema en la Tabla 17 que será llevada a cabo en cada escenario de prueba.

**Tabla 17**

*Secuencia de acciones para cada escenario de pruebas experimentales.*

<b>1. Despegue</b>	Despegar el drone y ubicarlo a la altura correcta de forma automática.
<b>2. Posicionamiento</b>	Control automático del drone para situarse en el centro del área minada.
<b>3. Obtención del plano de trabajo</b>	Reconocimiento de los marcadores ubicados en las esquinas del área minada para obtener el plano de trabajo.
<b>4. Planificación de trayectoria</b>	Detectar los obstáculos que estén presentes en el área minada y calcular la trayectoria que debe seguir el robot móvil.
<b>5. Control de seguimiento de trayectoria y detección de minas.</b>	Control servo visual para el seguimiento de la trayectoria del robot móvil de la planificación realizada y detección de minas con sensores a bordo del robot.

### 5.2.1. Pruebas para la detección de obstáculos

Con el fin de evaluar el desempeño del algoritmo de la sección 4.4 para detección de obstáculos basado en la filtración de umbrales de color, se colocaron 4 obstáculos(diferentes) en los dos escenarios establecidos con 3 diferentes niveles de iluminación en cada uno para así determinar el porcentaje de efectividad del mismo. El porcentaje de cada prueba se calcula con la ecuación (43).

$$Det_{obs} = \frac{\# \text{ Obstáculos detectados}}{\# \text{ Obstáculos totales}} \cdot 100\% \quad (43)$$

A continuación, se presenta en la Tabla 18 el resumen de los resultados de las pruebas en cada escenario con sus respectivos niveles de iluminación.

**Tabla 18**

*Resultados de la detección de obstáculos en diversos escenarios de iluminación*

Lugar de pruebas	Nivel de Iluminación	Número de Obstáculos		Detección de obstáculos (%)
		Reales	Detectados	
<b>Escenario Parquet oscuro</b>	33%-580lux	4	1	25%
	66%-655 lux	4	4	100%
	100%-729lux	4	4	100%
<b>Cancha Parquet claro</b>	33%-580lux	4	2	50%
	66%-655 lux	4	4	100%
	100%-729lux	4	4	100%
<b>%Promedio</b>				79.16%

Como se puede evidenciar en la Tabla 18 no es recomendable trabajar con niveles de iluminación bajo de los 580 luxes debido al problema que ocasionan para reconocer satisfactoriamente los obstáculos. El porcentaje promedio de reconocimiento de obstáculos basada en color termina siendo aceptable con un 79.16% efectividad pese a la baja certeza de las pruebas con el 33% de iluminación.

### 5.2.2. Pruebas de iluminación y altura en el plano de trabajo

En esta sección se evalúa la incidencia de la iluminación y altura sobre el plano de trabajo, en el tiempo y distancia del recorrido del robot, además de los obstáculos detectados. La prueba se realizó para un solo escenario con dimensiones de 1.2m x 1.2m, los resultados se presentan en la Tabla 19.

**Tabla 19**  
*Pruebas de iluminación y altura.*

Altura[m]	Iluminación											
	33%				66%				100%			
	Tiempo [s]	Distancia [m]	Obstáculos reales	Obstáculos detectados	Tiempo [s]	Distancia [m]	Obstáculos reales	Obstáculos detectados	Tiempo [s]	Distancia [m]	Obstáculos reales	Obstáculos detectados
2	158	12.22	3	1	205	13.16	3	3	187	13.08	3	3
2,5	167	12.28	3	1	213	13.25	3	3	196	13.17	3	3
3	184	12.35	3	1	219	13.37	3	3	208	13.22	3	3

Se ratifica con los resultados de la Tabla 19 que la detección de obstáculos con iluminación del 33% no es efectiva ya que no reconoció 2 obstáculos para todos los niveles de altura, mientras que con un nivel de iluminación del 66% y 100% se reconoce por completo todos los obstáculos. El nivel de altura y el porcentaje de iluminación repercuten en el tiempo y distancia que se demora el sistema en culminar el recorrido del área, como se puede observar en las mediciones del 33% de iluminación la distancia recorrida es menor debido a que no se detectaron dos obstáculos, sin embargo el tiempo es muy cercano a las mediciones al 100% esto se debe a que en algunos momentos no se reconoce el marcador del robot y el sistema se detiene hasta que este sea reconocido, debido a esto el tiempo de recorrido al 66% es mayor que el tiempo al 100% con distancias similares de recorrido.

Haciendo un resumen de la Tabla 19 para determinar el tiempo y distancia promedio de recorrido del robot en un área de trabajo de 1.2m x 1.2m se presenta los resultados en la Tabla 20.

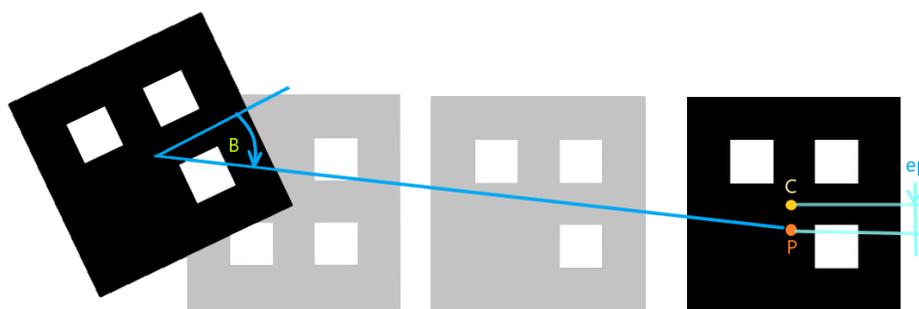
**Tabla 20.**

*Resumen de tiempos y promedios.*

Altura[m]	Tiempo[s]			Promedio [s]	Distancia recorrida[m]			Promedio [m]
	% Iluminación				% Iluminación			
	33%	66%	100%		33%	66%	100%	
<b>2</b>	158	205	187	183	12.22	13.16	13.08	12.82
<b>2,5</b>	167	213	196	192	12.28	13.25	13.17	12.9
<b>3</b>	184	219	208	203	12.35	13.37	13.22	12.98
<b>Promedios</b>	169	212	197	<b>192</b>	12.28	13.26	13.16	<b>12.9</b>

### 5.2.3. Pruebas de planificación y seguimiento de trayectoria

En esta sección se realizan 10 secuencias con el robot para determinar el error cuadrático medio (RMSE) del control de seguimiento de trayectoria, las pruebas serán realizadas ubicando al robot de cualquier punto del plano y se enviara la señal de inicio para que el robot se dirija a un punto en común en todos los ensayos una vez este llegue a su destino se detendrá y se determinará la distancia “ep” entre el punto “C” (Centro del robot) y el punto “P”( Destino), como se observa en la Figura 105.



**Figura 105.** Secuencia para determinar el error del control servovisual.

Una vez se ha acumulado los valores de las distancias en las 10 pruebas se procede a calcular el valor RMSE presentado en el resumen de la Tabla 21.

**Tabla 21**  
*RMSE objetivo ideal vs real.*

Prueba	Error (píxeles)
1	8
2	9
3	5
4	7
5	3
6	4
7	5
8	6
9	4
10	10
<b>RMSE</b>	6.49

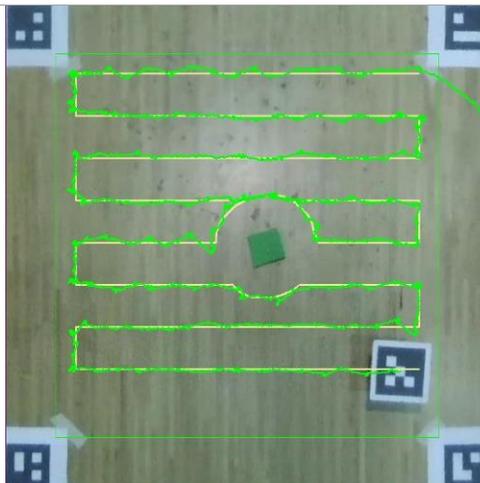
Como respaldo para verificar de una manera visual que el error en el seguimiento de trayectoria es relativamente pequeño se realiza 3 pruebas de cobertura total de superficies con varios materiales, número y formas de objetos, como se puede observar en la Tabla 22 en línea verde se ha graficado la trayectoria recorrida y en amarillo la ideal presentando gran similitud entre ellas.

**Tabla 22**  
*Pruebas de seguimiento y planificación de trayectorias*

Datos de la prueba		Resultado de la prueba
<b>Prueba 1</b>		
<b>Lugar</b>	Coliseo Gral. Miguel Iturralde. (Cancha)	
<b>Velocidad</b>	0.02[m/s]	
<b>Tiempo</b>	433[s]	



**Distancia** 9.6[m]



### Prueba 2

**Lugar** Coliseo Gral. Miguel Iturralde.  
(Escenario)

**Velocidad** 0.02[m/s]

**Tiempo** 587[s]

**Distancia** 13.1[m]



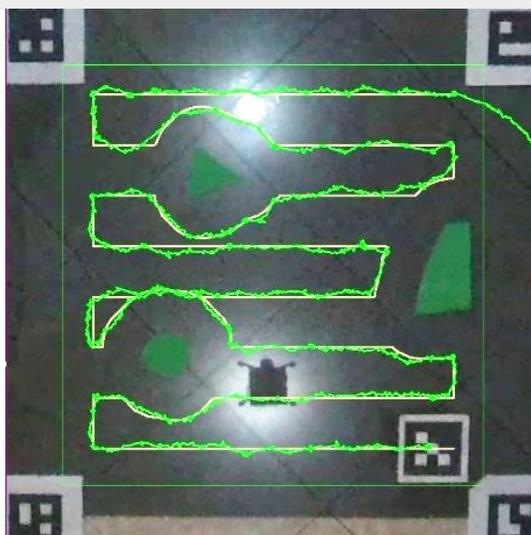
### Prueba 3

**Lugar** Domicilio piso de baldosa

**Velocidad** 0.02[m/s]

**Tiempo** 468[s]

**Distancia** 11.6[m]



#### 5.2.4. Pruebas de detección de minas antipersonales.

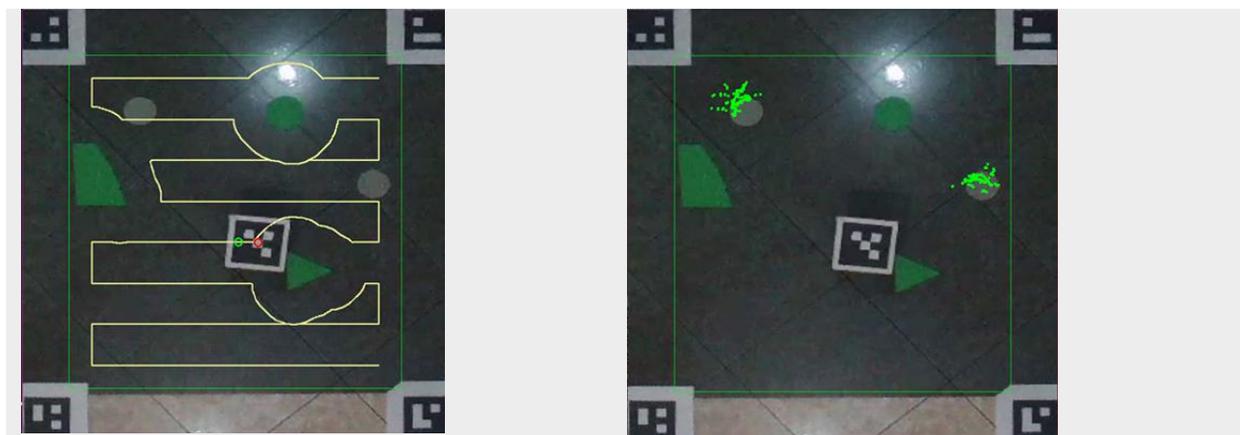
Para evaluar el porcentaje de aciertos del sistema de la sección 4.7 destinado a detectar las minas antipersonales existentes en un área, se procede a registrar 3 pruebas sobre un plano de 1.2m x 1.2m, las minas se han representado con latas metálicas redondas colocadas al azar en cualquier ubicación dentro del plano. Los obstáculos se han colocado de manera aleatoria para obtener trayectorias diferentes que deberá seguir el robot. Hay que tomar en cuenta que los sensores se han ubicado a una distancia no mayor a 3mm del suelo con el objetivo de que la detección sea efectiva. Las marcas de color verde que se registren en la ventana indicarán las zonas en las que se encontró metales que se tomaran como presuntas minas antipersonales. El cálculo del porcentaje de efectividad se lo realiza con la ecuación (44).

$$Det_{obs} = \frac{\# \text{ minas detectados}}{\# \text{ minas totales}} \cdot 100\% \quad (44)$$

A continuación, se presentan los resultados de las pruebas en la Tabla 23.

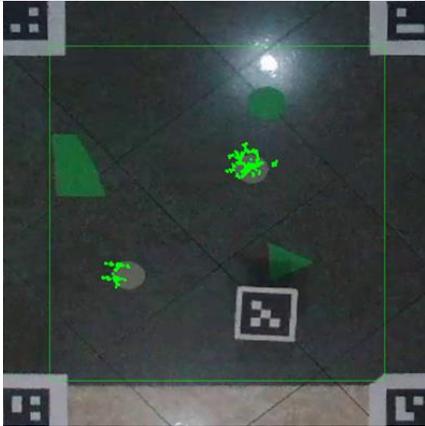
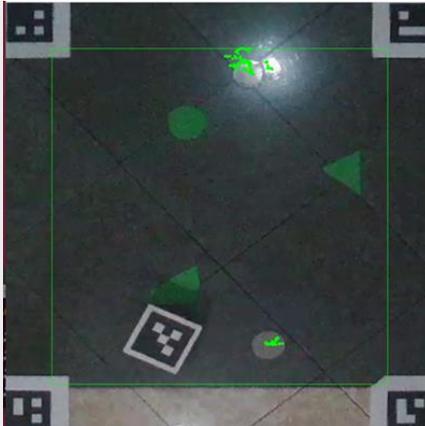
**Tabla 23**  
*Resultados de la detección de minas.*

Plano de trabajo	Rastro de minas	
N° prueba: 1	Número de Minas	Detección de minas (%)
Escenario: Piso de baldosa	Reales: 2	Detectados: 2 100%



CONTINÚA

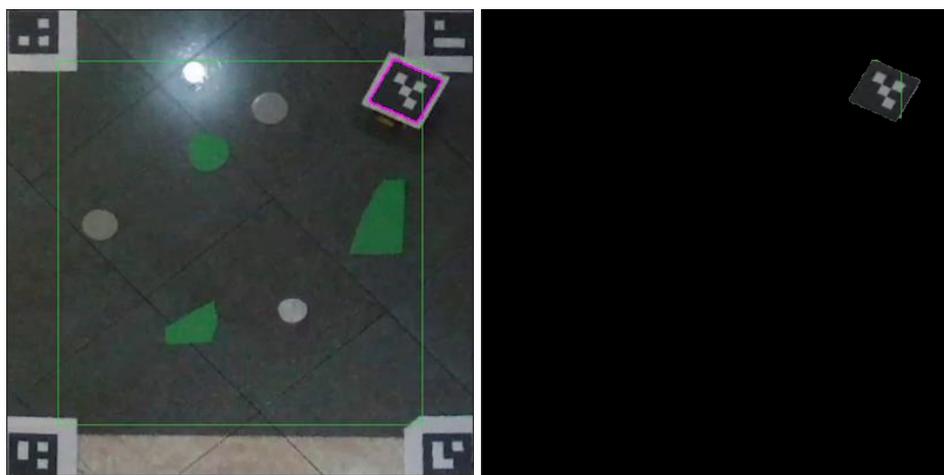


N° prueba: 2	Número de Minas		Detección de minas (%)
<b>Escenario:</b> Piso de baldosa	<b>Reales:</b> 2	<b>Detectados:</b> 2	100%
			
N° prueba: 3	Número de Minas		Detección de minas (%)
<b>Escenario:</b> Piso de baldosa	<b>Reales:</b> 2	<b>Detectados:</b> 2	100%
			

Como se puede observar en la Tabla 23, se han detectados en todas las pruebas cada una de las planchas metálicas con rastros de color verde, cabe destacar que los puntos que se marcan en los pixeles que están fuera del objeto se debe al cambio de altura que existe entre el marcador del robot y el suelo, como se observa no existen falsos positivos en la detección ni tampoco el cambio de ubicación influyo para que alguna plancha no sea detectada.

### 5.2.5. Pruebas de barrido de superficie

Para comprobar que el barrido del robot en zona minada es óptimo se realizaron 3 escenarios con diferentes obstáculos y posiciones, en cada prueba se video grabó el plano de trabajo con la ayuda de la herramienta de OpenCV “VideoWriter”, posteriormente se procedió a cargar el video al software de posproducción de imágenes Adobe After Effects, que permite seguir el movimiento de objetos en cada frame del video con la herramienta “Pincel tipo rotoscopia y Perfeccionar mate”, mediante esta se crea el patrón inicial marcado de color rosa que separa el objeto de su fondo en cada frame como se observa en la Figura 106.



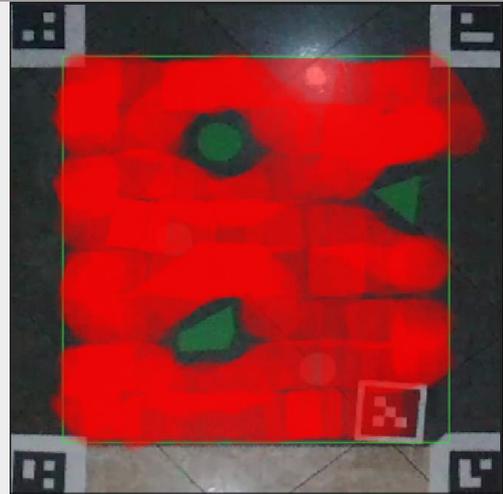
**Figura 106.** Separación del marcador del fondo de la imagen.

Una vez se ha separado el marcador del fondo en cada frame del video que describe el movimiento del robot, se aplica el efecto “eco” para visualizar todos los recortes y se cambia de color a rojo cada zona recorrida por el robot con el efecto “cambio de color”, el resultado se puede visualizar en las pruebas de la Tabla 24, al analizar cada una de ellas se observa que existen zonas con un color rojo intenso y otras muy opacas, las intensas se interpretan como zonas donde el robot se detuvo por alguna circunstancia o fueron recorridas por más de una ocasión de acuerdo a la planificación realizada, esto se puede notar recurrente en esquinas donde el robot tenga que girar a  $90^\circ$  o  $180^\circ$ , además de unas

pequeñas zonas en el solapamiento de barrido de los reglones. Las zonas en tono rojo oscuro demuestran situaciones en las que el robot recorrió fluidamente la trayectoria.

**Tabla 24**

*Pruebas de barrido de superficie.*

Prueba	Fotograma final video original	Fotograma final con After Effects
1		
2		

CONTINÚA



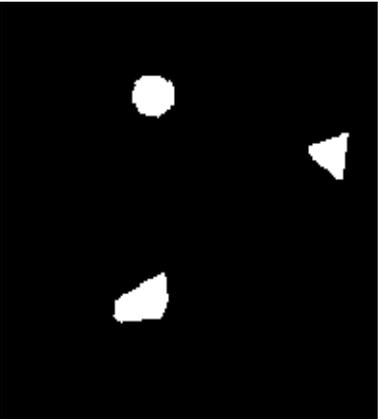


Una vez se ha graficado el barrido realizado por el robot se requiere verificar el porcentaje de efectividad en la cobertura del área de trabajo, partiendo desde el algoritmo de la sección 4.4 se obtiene la máscara de la cual se rescata el número de píxeles disponibles para que el robot pueda desplazarse, luego se prepara una máscara que limite la gama de colores rojo de las pruebas de la Tabla 24, mismas que se utilizan para determinar el número de píxeles que recorrió el robot. El porcentaje de efectividad se calcula con la ecuación (45).

$$\%E_{cob} = \frac{\# \text{ de píxeles recorridos}}{\# \text{ de píxeles disponibles}} \cdot 100\% \quad (45)$$

Los resultados del procedimiento se pueden observar en la Tabla 25, Tabla 26 y Tabla 27.

**Tabla 25***Prueba 1 de efectividad de cobertura de superficie.*

	Resultado After Effects	Resultado máscaras de color	
Resultados finales			
	Espacio disponible (negro)	Espacio recorrido (blanco)	
Resultados parciales			
Pixeles	37945	34301	Porcentaje de efectividad 90.39%

**Tabla 26**  
*Prueba 2 de efectividad de cobertura de superficie.*

	Resultado After Effects	Resultado máscaras de color	
Resultados finales			
	Espacio disponible (negro)	Espacio recorrido (blanco)	
Resultados parciales			
Pixeles	36870	34205	Porcentaje de efectividad 92.77%

**Tabla 27**  
*Prueba 3 de efectividad de cobertura de superficie.*

	Resultado After Effects	Resultado máscaras de color	
Resultados finales			
	Espacio disponible (negro)	Espacio recorrido (blanco)	
Resultados parciales			
Pixeles	36283	32506	Porcentaje de efectividad 89.6%

De los resultados de la Tabla 25, Tabla 26 y Tabla 27 se puede determinar que en promedio se obtiene un 91% de cobertura de la superficie libre de obstáculos de un área determinada, pero hay que tomar en cuenta que del 9% de error también incluyen zonas en las que el robot por sus dimensiones no puede alcanzar esas ubicaciones, es el caso de dos ejemplos de las pruebas realizadas en los que se observa que existen dos zonas de conflicto como se indica en la Figura 107, debido a esto el porcentaje de error puede ser aún menor.



*Figura 107.* Zonas de conflicto para la cobertura de superficie.

## CAPÍTULO VI

### CONCLUSIONES Y RECOMENDACIONES

#### 6.1. Conclusiones.

- Se estimó que el reconocimiento de patrones para el seguimiento de la plataforma robótica y la delimitación del área de trabajo, es mucho más robusta y efectiva con el algoritmo desarrollado para marcadores Aruco de la sección 4.1, en comparación con la librería de Python “Zbar” para reconocer códigos QR, ya que mediante las pruebas realizadas los códigos con esta librería solo se detectan hasta una distancia de 1.5m, mientras que con los marcadores Aruco se logró llegar a una distancia de 7m de forma estable.
- El uso de un controlador PI solventó el posicionamiento del dron en el centro del área de trabajo, mediante el rastreo del desplazamiento proporcional al error de posición del dron con respecto al centro del área de trabajo.
- La transformación a perspectiva cenital del área de trabajo, aporta estabilidad en el video, sin importar las perturbaciones externas que sufra el dron, mediante esto se facilita el control servo-visual de la plataforma robótica para el seguimiento de trayectoria, obteniendo un error del 6.5% con 10 puntos de prueba.
- El uso del algoritmo de Boustrophedon solventó la cobertura total de superficies, mediante el robot móvil de configuración diferencial se obtuvo una cobertura del 91% de la zona libre para transitar en el área de trabajo, sin embargo del 9% restante se determina que existen zonas a las que el robot no puede llegar por sus dimensiones, y otras muy pequeñas que se dejan como protección alrededor de los obstáculos para que el robot no colisione con los mismos, debido a

estas dos razones el porcentaje de superficie sin recorrer se puede minimizar si se reducen las dimensiones del robot.

- El uso de un controlador PID solucionó la orientación de la plataforma móvil respecto a los puntos de la trayectoria planificada para el control servo-visual, para determinar el ángulo de rotación es necesario establecer un triángulo escaleno formado entre el centro y frente del robot con el punto a seguir de la trayectoria.
- Para determinar el valor del ángulo y el segundo para encontrar el sentido de giro ya sea horario o antihorario que debe realizar el robot se utilizó el teorema de la ley de cosenos y la pendiente de la recta.
- Con base en las pruebas realizadas, la iluminación y altura del dron al plano de trabajo se debe tomar en cuenta ciertos factores como es la iluminación, ya que el fallo de luz afecta al reconocimiento de colores (objetos) que se encuentren en la zona de trabajo, y la altura influye en la detección de los marcadores, por ende, estos dos factores repercuten en el tiempo y distancia de recorrido del robot.
- De las pruebas realizadas se puede determinar que el color de la superficie de trabajo no influye en el desempeño del algoritmo para detectar obstáculos de color.
- De acuerdo con las pruebas de detección de objetos metálicos existentes en la superficie de trabajo, se concluyó que la combinación de la efectividad de cobertura de superficie con el algoritmo de Boustrophedon y el mecanismo implementado en el robot con sensores inductivos, tienen una certeza del 100% de encontrar los objetos.

## **6.2. Recomendaciones.**

- Para una mejor detección de objetos metálicos en la superficie de trabajo es recomendable calibrar los sensores inductivos a una distancia no mayor a 3mm del suelo, ya que este es el rango de detección de los sensores seleccionados para el proyecto.
- Para alivianar la estructura del robot se recomienda utilizar piezas de acrílico o impresiones 3D en PLA.
- Efectuar las pruebas en una superficie libre de imperfecciones o suciedad ya que cualquiera de estos factores puede impedir el correcto desenvolvimiento del robot móvil en su trayectoria.
- Se recomienda tener las precauciones necesarias al momento de volar el drone, especialmente si las pruebas se realizan en interiores, debido a que exista la posibilidad de que se pierda la comunicación por interferencias u otros imprevistos, y pueden causar colisiones con el medioambiente, provocando daños en la estructura del drone.

## **6.3. Trabajos Futuros.**

- Se plantea como trabajo futuro la implementación de varias plataformas terrestres para desarrollar una arquitectura de control distribuida, ya que estos pueden actuar de forma independiente sin necesidad que reciban órdenes de las otras plataformas, lo que ayudaría a recorrer el entorno de trabajo mucho más rápido y que cada robot implementado se encargue de un área específica, de esta manera se aprovecharía la autonomía del UAV y los robots terrestres.

## REFERENCIAS BIBLIOGRÁFICAS

- Adobe. (2019). *Adobe After Effects*. Obtenido de Adobe After Effects: [https://helpx.adobe.com/es/pdf/after\\_effects\\_reference.pdf](https://helpx.adobe.com/es/pdf/after_effects_reference.pdf)
- AESA. (2018). FAQ Drones: Normativa Drones.
- Agudelo, E., & Garces, P. (2006). *Diseño e implementación de un robot móvil autónomo con detección de color*. Obtenido de Universidad de San Buenaventura: <http://biblioteca.usbbog.edu.co:8080/Biblioteca/BDigital/37903.pdf>
- Aguilar, W. G., & Angulo, C. (2014). Real-time video stabilization without phantom movements for micro aerial vehicles. *EURASIP Journal on Image and Video Processing*, 1, 1-13.
- Aguilar, W. G., & Angulo, C. (2014). Robust video stabilization based on motion intention for low-cost micro aerial vehicles. *11th International Multi-Conference on Systems, Signals & Devices (SSD)*. Barcelona, Spain.
- Aguilar, W. G., & Angulo, C. (2016). Real-Time Model-Based Video Stabilization for Microaerial Vehicles. *Neural Processing Letters*, 43(2), 459-477.
- Aguilar, W. G., & Morales, S. (2016). 3D Environment Mapping Using the Kinect V2 and Path Planning Based on RRT Algorithms. *Electronics*, 5(4), 70.
- Aguilar, W. G., Abad, V., Ruiz, H., Aguilar, J., & Aguilar-Castillo, F. (2017). RRT-Based Path Planning for Virtual Bronchoscopy Simulator. En *Lecture Notes in Computer Science* (págs. 155-165).
- Aguilar, W. G., Álvarez, L., Grijalva, S., & Rojas, I. (2019). Monocular Vision-Based Dynamic Moving Obstacles Detection and Avoidance. En *Lecture Notes in Computer Science* (págs. 386-398). Springer.

- Aguilar, W. G., Angulo, C., & Costa-Castello, R. (2017). Autonomous Navigation Control for Quadrotors in Trajectories Tracking. En *Lecture Notes in Computer Science* (págs. 287-297).
- Aguilar, W. G., Angulo, C., & Pardo, J. A. (2017). Motion intention optimization for multirotor robust video stabilization. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.
- Aguilar, W. G., Caballeros, J., Segarra, D., & Castro, P. (2018). RRT Path Planning and Morphological Segmentation Based Navigation for a Tetrapod Robot. En *Lecture Notes in Computer Science* (págs. 273-280). Springer.
- Aguilar, W. G., Casaliglla, V., & Pólit, J. (2017). Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles. *Electronics*, 6(1), 10.
- Aguilar, W. G., Casaliglla, V., & Pólit, J. (2017). Obstacle Avoidance for Low-Cost UAVs. *IEEE 11th International Conference on Semantic Computing (ICSC)*. San Diego.
- Aguilar, W. G., Casaliglla, V., Pólit, J., Abad, V., & Ruiz, H. (2017). Obstacle Avoidance for Flight Safety on Unmanned Aerial Vehicles. En *Lecture Notes in Computer Science* (págs. 575-584).
- Aguilar, W. G., Castro, P., Caballeros, J., & Segarra, D. (2018). Homography and Morphological Detection-Based Virtual Shooting Range. En *Lecture Notes in Computer Science* (págs. 267-272). Springer.
- Aguilar, W. G., Cobeña, B., Rodriguez, G., Salcedo, V. S., & Collaguazo, B. (2018). SVM and RGB-D Sensor Based Gesture Recognition for UAV Control. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (págs. 713-719). Springer.

- Aguilar, W. G., Luna, M. A., Moya, J. F., Abad, V., Parra, H., & Ruiz, H. (2017). Pedestrian Detection for UAVs Using Cascade Classifiers with Meanshift. *IEEE 11th International Conference on Semantic Computing (ICSC)*. San Diego.
- Aguilar, W. G., Luna, M. A., Moya, J. F., Abad, V., Ruiz, H., Parra, H., & Angulo, C. (2017). Pedestrian Detection for UAVs Using Cascade Classifiers and Saliency Maps. En *Lecture Notes in Computer Science* (págs. 563-574).
- Aguilar, W. G., Luna, M. A., Moya, J. F., Abad, V., Ruiz, H., Parra, H., & Lopez, W. (2017). Cascade Classifiers and Saliency Maps Based People Detection. En *Lecture Notes in Computer Science* (págs. 501-510).
- Aguilar, W. G., Luna, M. A., Moya, J. F., Luna, M. P., Abad, V., Ruiz, H., & Parra, H. (2017). Real-Time Detection and Simulation of Abnormal Crowd Behavior. En *Lecture Notes in Computer Science* (págs. 420-428).
- Aguilar, W. G., Luna, M. A., Ruiz, H., Moya, J. F., Luna, M. P., Abad, V., & Parra, H. (2017). Statistical Abnormal Crowd Behavior Detection and Simulation for Real-Time Applications. En *Lecture Notes in Computer Science* (págs. 671-682).
- Aguilar, W. G., Manosalvas, J. F., Guillén, J. A., & Collaguazo, B. (2018). Robust Motion Estimation Based on Multiple Monocular Camera for Indoor Autonomous Navigation of Micro Aerial Vehicle. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (págs. 547-561). Springer.
- Aguilar, W. G., Morales, S., Ruiz, H., & Abad, V. (2017). RRT\* GL Based Optimal Path Planning for Real-Time Navigation of UAVs. En *Lecture Notes in Computer Science* (págs. 585-595).
- Aguilar, W. G., Morales, S., Ruiz, H., & Abad, V. (2017). RRT\* GL Based Path Planning for Virtual Aerial Navigation. En *Lecture Notes in Computer Science* (págs. 176-184).

- Aguilar, W. G., Quisaguano, F., Alvarez, L., Pardo, J., & Proaño, Z. (2018). Monocular Depth Perception on a Micro-UAV Using Convolutional Neuronal Networks. En *Lecture Notes in Computer Science* (págs. 392-397). Springer.
- Aguilar, W. G., Quisaguano, F., Rodríguez, G., Alvarez, L., Limaico, A., & Sandoval, D. (2018). Convolutional Neuronal Networks Based Monocular Object Detection and Depth Perception for Micro UAVs. En *Lecture Notes in Computer Science* (págs. 401-410). Springer.
- Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). On-Board Visual SLAM on a UGV Using a RGB-D Camera. En *Lecture Notes in Computer Science* (págs. 298-308).
- Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). Real-Time 3D Modeling with a RGB-D Camera and On-Board Processing. En *Lecture Notes in Computer Science* (págs. 410-419).
- Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). Visual SLAM with a RGB-D Camera on a Quadrotor UAV Using on-Board Processing. En *Lecture Notes in Computer Science* (págs. 596-606).
- Aguilar, W. G., Salcedo, V. S., Sandoval, D. S., & Cobeña, B. (2017). Developing of a Video-Based Model for UAV Autonomous Navigation. En *Communications in Computer and Information Science* (págs. 94-105).
- Aguilar, W. G., Sandoval, D., Caballeros, J., Alvarez, L., Limaico, A., Rodríguez, G., & Quisaguano, F. (2018). Graph Based RRT Optimization for Autonomous Mobile Robots. En *Lecture Notes in Computer Science* (págs. 12-21). Springer.

- Aguilar, W. G., Sandoval, S., Limaico, A., Villegas-Pico, M., & Asimbaya, I. (2019). Path Planning Based Navigation Using LIDAR for an Ackerman Unmanned Ground Vehicle. En *Lecture Notes in Computer Science* (págs. 399-410). Springer.
- Aguilar, W., & Salcedo, V. S. (2018). Aterrizaje automático de un vehículo aéreo no tripulado basandose en seguimiento de puntos de interés para superficies móviles.
- Albus, J. (2015). A Control Architecture for Cooperative Intelligent Robots. 7-8.
- Alejando, & Vicente. (2009). *Sistemas de Visión Artificial*.
- Almeida, I., & Ochoa, J. (Abril de 2013). *Diseño y Construcción de un robot explorador de terreno*.  
Obtenido de Universidad Politécnica Salesiana:  
<https://dspace.ups.edu.ec/bitstream/123456789/4498/1/UPS-GT000403.pdf>
- Alvarez, L. (2018). *Desarrollo de un sistema vinculado a un Micro-UAV para la detección y evasión de objetos dinámicos a partir de imágenes monoculares*. Obtenido de Universidad de las Fuerzas Armadas ESPE: <https://repositorio.espe.edu.ec/bitstream/21000/15531/1/T-ESPE-040604.pdf>
- Amaguaña, F., Collaguazo, B., Tituaña, J., & Aguilar, W. G. (2018). Simulation System Based on Augmented Reality for Optimization of Training Tactics on Military Operations. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (págs. 394-403). Springer.
- Andrea, C. C., Byron, J. Q., Jorge, P. I., Inti, T. C., & Aguilar, W. G. (2018). Geolocation and Counting of People with Aerial Thermal Imaging for Rescue Purposes. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (págs. 171-182). Springer.
- Aristondo Etxeberria, J. (2010). Algoritmo de reconocimiento de forma y color para una plataforma robotica.

- Askix. (2019). *Askix*. Obtenido de [https://www.askix.com/enjambre-de-robots-asamblea-y-cooperativa-de-transporte\\_2.html](https://www.askix.com/enjambre-de-robots-asamblea-y-cooperativa-de-transporte_2.html)
- Bambino. (2008). Una introducción a los robots móviles.
- Barrientos, V., García, J., & Silva, R. (2007). Robots Móviles: Evolución y Estado del Arte. *Revistas Científicas de América Latina, el Caribe, España y Portugal*, 12-17.
- Basantes, J., Godoy, L., Carvajal, T., Castro, R., Toulkeridis, T., Fuertes, W., . . . Addison, A. (2018). Capture and processing of geospatial data with laser scanner system for 3D modeling and virtual reality of Amazonian Caves. *IEEE Ecuador Technical Chapters Meeting (ETCM)*. Samborondón, Ecuador.
- Benito Carrasco, J. (2015). *Integración de un UAV (vehículo aéreo no tripulado) en la plataforma robótica ARGOS*. Obtenido de Universidad Autónoma de Madrid : <http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20150407JuanAlbertoBenitoCarrasco.pdf>
- Bermúdez, G. (2002). Robots Móviles: Teoría, Aplicaciones y Experiencias. *Universidad Distrital Francisco José de Caldas*, 16-19.
- Bohlim, R., & Kavraki, L. (2000). Path planning using Lazy PRM. *Proceedings-IEEE International Conference on Robotics and Automation*, 521-528.
- Bohórquez, R., Mendoza, L., & Cortés, C. (2013). Sistema de inspección y vigilancia utilizando un robot aéreo guiado mediante visión artificial. *Iteckn*, 190-198.
- Boucher, P. (2014). Domesticating the Drone: The Demilitarisation of Unmanned Aircraft for Civil Markets. *Science and Engineering Ethics*, 1393-1412.
- Bravo, F., Patiño, D., Melo, K., & Parra, C. (2011). Switching control and modeling of mobile robots formation. *IEEE: IX Latin American Robotics Symposium and IEEE Colombian Conference on Automatic Control, 2011 IEEE*, 1-6.

- Buniyamin, A., Sariff, N., Wan, A., & Mohamad, Z. (2011). Robot global path planning overview and a variation of ant colony system algorithm. *Int. J. Math. Comput. Simul.*, 9-16.
- Burgard, W., Fox, D., & Simmons, R. (2000). Collaborative multi-robot exploration. *Proceedings - IEEE International Conference on Robotics and Automation*, 476-481.
- Calderón, M., & Maldonado, D. (2014). Control por visión de un cuadricóptero utilizando ROS. 46-52.
- Campoverde, E. (2017). *Desarrollo de dos robots para realizar trabajo cooperativo*. Obtenido de Universidad de las Fuerzas Armadas ESPE: <http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/12835/T-ESPE-053726.pdf?sequence=1&isAllowed=y>
- Chantemargue, F., Courant, M., & Hirsbrunner, B. (2002). *Autonomous Agents and Collective Robotics*. Obtenido de University of Fribourg Computer Science Department: [https://www.researchgate.net/publication/2758642\\_Autonomous\\_Agents\\_and\\_Collective\\_Robotics](https://www.researchgate.net/publication/2758642_Autonomous_Agents_and_Collective_Robotics)
- Chaumette, F., Rives, P., & Espiau, B. (2014). Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. *Proceedings. 1991 IEEE Int. Conf. Robot. Automation*, 2248-2253.
- Coliaux, D., & Hanappe, P. (2017). LettuceThink: A open and versatile robotic platform for weeding and crop monitoring on microfarms. *EFITA WCCA 2017 Conference*. France.
- Cuautle, J., Berra, E., & Pérez, M. (Diciembre de 2018). *Róbotica cooperativa para el transporte de objetos: revisión histórica de estrategias*. Obtenido de Universidad del Valle de Puebla: [https://www.researchgate.net/publication/329428308\\_Robotica\\_cooperativa\\_para\\_el\\_transporte\\_de\\_objetos\\_revisión\\_histórica\\_de\\_estrategias](https://www.researchgate.net/publication/329428308_Robotica_cooperativa_para_el_transporte_de_objetos_revisión_histórica_de_estrategias)

- Cuero, A., García, L., Carrio, Marco, & Sánchez, A. (2016). Clasificación de las aeronaves, análisis de criterios. 76-85.
- Daniela, J. (2017). El Desminado Humanitario En El Ecuador: La Convención De Ottawa Sobre Minas Antipersonal (1998-2014). 1-147.
- Delgado Hernández, M. Á. (2016). Visión artificial aplicada a la robótica. *Universidad de la Laguna*.
- Díaz, Y., & Becerra, R. (2014). The programming language Python. *Centro de Información y Gestión Tecnológica de Santiago de Cuba.*, 1-13.
- Electronics Katrangi Trading*. (08 de 09 de 2019). Obtenido de <http://www.ekt2.com/>
- Escalante Torrado, J. O., & Cáceres Jiménez, J. J. (2016). Ortomosaicos y modelos digitales de elevación generados a partir de imagenes tomadas con sistemas UAV.
- European Robotics Week, S. (29 de 11 de 2018). *Robots móviles: Semana Europea de la Robótica*. Obtenido de [https://www.eu-robotics.net/robotics\\_week/newsroom/press/erw2018-next-generation-robot-press-release.html?changelang=4](https://www.eu-robotics.net/robotics_week/newsroom/press/erw2018-next-generation-robot-press-release.html?changelang=4)
- Fernández, A., & Lezama, E. (2005). Sensores Magnéticos e Inductivos. 1-168.
- Fuertes, M., Lucena, M., Pérez, N., & Valdivia, J. (2016). *Extracción de Características en un esquema de recuperación de imágenes*. Obtenido de Procesamiento Distribuido y Paralelo. Tratamiento de Señales: <https://pdfs.semanticscholar.org/d2d3/5f2541a5cd0cffbd5baa060a91a0ab869e92.pdf>
- Galarza, J., Pérez, E., Serrano, E., Tapia, A., & Aguilar, W. G. (2018). Pose Estimation Based on Monocular Visual Odometry and Lane Detection for Intelligent Vehicles. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (págs. 562-566). Springer.

- Galceran, E., & Carreras, M. (2013). A survey on coverage path planning for robotics. *Robotics and Autonomous*, 1258-1276.
- García Gómez, G. J., Pomares Baeza, J., & Torres Medina, F. (2010). Control Visual-Fuerza autocalibrado para seguimiento de trayectorias en tareas cooperativas robóticas.
- García, A. (2013). ROS: Robot Operating System. *Electrónica Industrial*, 15-23.
- Garrido, S., Muñoz, R., & Madrid, F. (2014). Automatic generation and detection of highly reliable fiducial markers. *Pattern Recognition*, 2280-2292.
- Gasparetto, A., Boscaroli, P., Lanzutti, A., & Vidoni, R. (2015). Path Planning and Trajectory Planning Algorithms: A General Overview. *Motion and Operation Planning of Robotic System*, 426-431.
- GeekFactory. (08 de 09 de 2019). Obtenido de <https://www.geekfactory.mx/tienda/motores-y-controladores/controlador-de-motor-dual-tb6612fng/>
- Giusti, A., Guzzi, J., Ciresan, D., He, F.-L., Rodríguez, J., Fontana, F., . . . Gambardella, L. (2016). Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots. *IEEE Robotics and Automation Letters*, 661-667.
- Gómez, G. (2011). Visión computacional.
- González Marcos, A., & Pisón Ascacibar, F. J. (2006). *Técnicas y algoritmos de visión artificial*. La Rioja.
- González Torres, C. F. (2015). Plataforma para robótica cooperativa soportada por teléfonos inteligentes Android.
- González, W. (2017). *Diseño y Construcción de un vehículo aéreo no tripulado (UAV) del tipo drone cuadricóptero de carreras*. Obtenido de Institución Universitaria Politécnico Grancolombiano:

<http://alejandria.poligran.edu.co/bitstream/handle/10823/1032/EL%20DRONE%20OK.PDF?sequence=1&isAllowed=y>

- González, C., Aguilar, P., & Calvo, P. (2014). De los UAV a los RPAS. *Perfiles. IDS*.
- González, C., Herranz, F., & Calvo, P. (2014). De los UAV a los RPAS. *Perfiles IDS*, 1-124.
- Gracia, L. I. (2010). Modelado Cinemático y Control de Robots Móviles con Ruedas. *Tesis Doctoral: Departamento de Ingeniería de Sistemas y Automática. Universidad Politécnica de Valencia*, 39-43.
- Grijalva, S., & Aguilar, W. G. (2019). Landmark-Based Virtual Path Estimation for Assisted UAV FPV Tele-Operation with Augmented Reality. En *Lecture Notes in Computer Science* (págs. 688-700). Springer.
- Grijalva, S., Cueva, G., Ramírez, D., & Aguilar, W. G. (2019). HSVM-Based Human Activity Recognition Using Smartphones. En *Lecture Notes in Computer Science* (págs. 217-228). Springer.
- Gualdrón, O., Duque, O., & Chacón, M. (2013). Design of a face recognition system using the hybridisation pattern recognition technique, and vision IA, focused on Robotics safety and social interaction. *Repositorio Universidad de Paplona, Colombia*, 16-28.
- Guerrero Virramontes, A., Becerra Fermín, H. M., & Torres Muñoz, O. S. (2014). Implementacion de un control visual para robots móviles con ruedas.
- Hannuksela, J., Barnard, M., Sangi, P., & Heikkila, J. (2011). Camera-Based Motion Recognition for Mobile Interaction. *International Scholarly Research Notices*, 12-24.
- Heredia, D. (2018). Seguimiento de trayectorias para un cuadricóptero parrot basado en un controlador difuso. *Trabajo de Titulación: Electrónica, Automatización y Control*, 37-41.

- Hernández, L., Pedraza, J., & Velázquez, G. (2014). Diseño y desarrollo de un vehículo volador de cuatro hélices no tripulado de bajo costo. *Asociación Mexicana de Mecátronica A.C*, 23-36.
- Hutchinson, S., Hanger, G., & Corke, P. (1996). Tutorial on Visual Servo Control. . *IEEE Transactions on Robotics and Automation.*, 651-671.
- Iser, J. (Enero de 2003). *Estudio experimental de métodos de calibración y autocalibración de cámaras*. Obtenido de Universidad De Las Palmas de Gran Canaria: [https://www.researchgate.net/publication/38182582\\_Estudio\\_experimental\\_de\\_metodos\\_de\\_calibracion\\_y\\_autocalibracion\\_de\\_camaras](https://www.researchgate.net/publication/38182582_Estudio_experimental_de_metodos_de_calibracion_y_autocalibracion_de_camaras)
- Ismail, A., & Hassaballah, M. (2015). *Image Feature Detectors and Descriptors*. Obtenido de Studies in Computational Intelligence: <https://www.springer.com/series/7092>
- Jara-Olmedo, A., Medina-Pazmiño, W., Mesías, R., Araujo-Villaroel, B., Aguilar, W. G., & Pardo, J. A. (2018). Interface of Optimal Electro-Optical/Infrared for Unmanned Aerial Vehicles. En *Smart Innovation, Systems and Technologies* (págs. 372-380).
- Jara-Olmedo, A., Medina-Pazmiño, W., Tozer, T., Aguilar, W. G., & Pardo, J. A. (2018). E-services from Emergency Communication Network: Aerial Platform Evaluation. *International Conference on eDemocracy & eGovernment (ICEDEG)* (págs. 251-256). IEEE.
- Jiménez, C. (2017). Desarrollo de un sistema de control de un robot móvil. 14-16.
- Jimenez, G., Sevillano, J., & Vicente, S. (1999). A generalization of path following for mobile robots. *IEEE Int. Conference Robotics and Automation*, 7-12.
- Kendoul, F. (2012). Survey of Advances in Guidance , Navigation , and Control of Unmanned Rotorcraft Systems. 315-378.
- Krishna, N. (2011). The Flight Characteristics of Unmanned Aircraft Systems .

- Lascano, E., & Rosero, C. (2017). Plataforma de robótica móvil para experimentos a través de ROS: Desarrollo de la tarjeta de tiempo real. *Repositorio Digital: Universidad Técnica del Norte*, 31-46.
- Leal, D. (2018). Cooperación Cordada entre Robot Terrestre ( UGV ) y Robot Aéreo ( UAV ). 23-35.
- Liang Cao, Z., Huang, Y., & Hall, E. (1988). Region Filling Operations with Random Obstacle Avoidance for Mobile Robots. *Journal of Robotics Systems*, 87-102.
- Lopes, D., Duarte, P., & Araújo, H. (2012). Algoritmos de cobertura para frotas de robôs sujeitos a restrições. 12-19.
- López, L., Mata, E., & García, E. (2001). Estado del arte en robótica móvil autónoma distribuida. *Conciencia Tecnológica: Instituto Tecnológico de Aguascalientes*, 17-29. Obtenido de Con: <https://www.redalyc.org/pdf/944/94401701.pdf>
- Lyra, M., Ploussi, A., & Georgantzoglou, A. (2011). *MATLAB as a Tool in Nuclear Medicine Image Processing*. Obtenido de MATLAB - A Ubiquitous Tool for the Practical Engineer: [https://www.researchgate.net/publication/221918148\\_MATLAB\\_as\\_a\\_Tool\\_in\\_Nuclear\\_Medicine\\_Image\\_Processing](https://www.researchgate.net/publication/221918148_MATLAB_as_a_Tool_in_Nuclear_Medicine_Image_Processing)
- MacDonald, J., Lockwood, McFee, J., Altahuler, T., Carin, L., Harmon, R., . . . Weaver, R. (2003). Alternatives for Landmine detection.
- Marín, L., Vallés, M., & Valera, Á. (2014). Localización de robots móviles de recursos limitados basada en fusión sensorial por eventos. 127-129.
- Márquez, C., Silva, R., & Marcelino, M. (2014). Robots Móviles de Ruedas: Generalidades. *Instituto Politécnico Nacional, UPIICSA, Sección de Estudios de Posgrado e Investigación*, 13-16.
- Min, B., Shin, H., & Tahk, M. (2015). Control System Design for an Autonomous. *Energy*, 1-7.

- Molina, M., & Rodríguez, E. (2014). *Flotilla de Robots para trabajos en Robótica Cooperativa*. Obtenido de Universidad Militar Nueva Granada: <https://repository.unimilitar.edu.co/bitstream/handle/10654/12129/FLOTILLA%20DE%20ROBOTS%20PARA%20TRABAJOS%20EN%20ROBOTICA%20COOPERATIVA.pdf?sequence=1>
- Monajjemi, M. (2017). *Bebop Autonomy*. 21-23.
- Morales, L., & Paucar, C. (2017). Investigación e implementación de un sistema de seguridad fijo y móvil mediante un dron usando visión artificial para detección y seguimiento de personas en un ambiente externo específico. *Trabajo de Titulación Universidad de las Fuerzas Armadas ESPE*, 29-30.
- Nadales, C. (2009). *Control de un Quadrotor mediante la plataforma Arduino*. Obtenido de Escola Politècnica Superior de Castelldefels: <https://upcommons.upc.edu/bitstream/handle/2099.1/8047/memoria.pdf?sequence=8>
- Nielsen, M. (Abril de 2016). *Usos y aplicaciones de los drones en el futuro*. Obtenido de EQUIPOALTRAN.
- Nixon, M., & Aguado, A. (2012). *Images, Sampling and Frequency Domain Processing*. En M. Nixon, & A. Aguado, *Feature Extraction and Image Processing for Computer Vision*. Elsevier.
- Núñez Torres, M., León, C. F., & Cardenas, P. (2016). ROS sistema operativo para robótica, nociones y aplicaciones.
- Oksanen, & Visala. (2009). Coverage Path Planning Algorithms for Agricultural Field Machines. *Journal of Field Robotics*.
- OpenCV. (2017). *Detection of Aruco Markers*. Obtenido de [https://docs.opencv.org/trunk/d5/dae/tutorial\\_aruco\\_detection.html](https://docs.opencv.org/trunk/d5/dae/tutorial_aruco_detection.html)

- OpenCV-Python Tutorilas*. (21 de 09 de 2019). Obtenido de [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_contours/py\\_contour\\_features/py\\_contour\\_features.html?highlight=cv2%20approxpolydp](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contour_features/py_contour_features.html?highlight=cv2%20approxpolydp)
- Orbea, D., Moposita, J., Aguilar, W. G., Paredes, M., León, G., & Jara-Olmedo, A. (2017). Math Model of UAV Multi Rotor Prototype with Fixed Wing Aerodynamic Structure for a Flight Simulator. En *Lecture Notes in Computer Science* (págs. 199-211).
- Orbea, D., Moposita, J., Aguilar, W. G., Paredes, M., Reyes, R. P., & Montoya, L. (2017). Vertical take off and landing with fixed rotor. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.
- Ortigoza, R., & Sánchez, R. (2015). *Estado del Arte: Robots móviles*. Obtenido de <https://www.redalyc.org/pdf/4026/402640448003.pdf>
- Otto, A., Agatz, N., Campbell, J., Golden, B., & Pesch, E. (2018). Optimization approaches for civil applications of unmanned aerial vehicles (UAVs). *Networks*, 3-7. Obtenido de <https://onlinelibrary.wiley.com/doi/abs/10.1002/net.21818>
- Pappot, M., & Boer, R. (2015). The integration of drones in today's society. *Procedia Engineering*, 54-63.
- Pardo, J. A., Aguilar, W. G., & Toulkeridis, T. (2017). Wireless communication system for the transmission of thermal images from a UAV. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.
- Parrot. (2018). *Parrot: Bebop 2*.
- Payas, & Areny. (2001). Sensores y acondicionadores de señal.
- Pepperl-Fuchs. (2019). *Sensores inductivos*. Obtenido de Sensores inductivos: <https://www.pepperl-fuchs.com/global/es/index.htm>
- Perry, M. J. (2012). Pre-flight passenger safety briefings. *Aeroscience*. 17-20.

- Ponticelli Lima, R. C., & Gonzales de Santos, P. (2011). Sistema de exploración de terrenos con robots móviles: aplicación en tareas de detección y localización de minas antipersonas.
- Python. (2019). *Software de programación "Python"*. Obtenido de <https://www.python.org/>
- Quisaguano, F. (2018). *Desarrollo de un sistema simultáneo de seguimiento de personas y evasión de obstáculos no definidos usando estimación de profundidad para imágenes monoculares en un Micro-UAV*. Obtenido de Universidad de las Fuerzas Armadas ESPE: <https://repositorio.espe.edu.ec/bitstream/21000/15530/1/T-ESPE-040603.pdf>
- Reale, F., Patiño, C., & Arrieta, F. (2014). Estudio del estado del arte: Navegación en robots agrícolas. 1-55.
- Ribo, M., Pinz, A., & Fuhrmann, L. (2001). A new optical tracking system for virtual and augmented reality applications. *Proceedings of the IEEE Instrumentation and Measurement Technical Conference*, 1932-1936.
- ROS. (2019). *ROS:Kinetic*. Obtenido de ROS:Kinetic: <http://wiki.ros.org/es>
- Ruipérez, M., & Conejero, A. (2016). *Diseño y Fabricación de un drone*. Obtenido de Universidad Politécnica de Valencia. Escuela Superior de Ingeniería del Diseño: <http://alejandria.poligran.edu.co/bitstream/handle/10823/1032/EL%20DRONE%20OK.PDF?sequence=1&isAllowed=y>
- Salcedo, V., Aguilar, W. G., Cobeña, B. P., & Proaño, Z. (2018). On-Board Target Virtualization Using Image Features for UAV Autonomous Tracking. En *Lecture Notes in Computer Science* (págs. 384-391). Springer.
- Salgado, M. F., Tierra, A., & Aguilar, W. G. (2017). Travel Planning in Public Transport Networks Applying the Algorithm A\* for Metropolitan District of Quito. *Analysis*, 4, 11.

- Salgado, M. F., Tierra, A., Sandoval, D. S., & Aguilar, W. G. (2017). Travel Time Estimation of Public Transport Networks Based on Commercial Incidence Areas in Quito Historic Center. *Analysis*, 294, 78448.
- Sánchez, C. (2017). Vehículos aéreos no tripulados: Descripciones generales y aplicaciones. *Desarrollo Tecnológico e innovación Empresarial*, 22-25.
- Scaramuzza, D., & Fraundorfer, F. (2011). Visual Odometry. *IEEE Robotics & Automation Magazine*, 80-92. Obtenido de IEEE Robotics & Automation Magazine.
- Šeda, M. (2007). Roadmap Methods vs. Cell Decomposition in Robot Motion Planning. *Proceedings of the 6th WSEAS International Conference on Signal Processing, Robotics and Automation*, 127-132.
- Segarra, D., Caballeros, J., & Aguilar, W. G. (2018). Visual Based Autonomous Navigation for Legged Robots. En *Lecture Notes in Computer Science* (págs. 22-34). Springer.
- ServoCity. (2019). Obtenido de <https://www.servocity.com/10-rpm-gear-motor>
- Shan, G., Li, X., Zhang, Y., Wang, T., & Fang, Y. (2018). A Deep Learning-based Visual Perception Approach for Mobile Robots. *Proceedings 2018 Chinese Automation Congress*.
- Sobrado Malpartida, E., & Tafur Sotelo, J. (2006). Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot. 1-125.
- Solanque, L., Molina, M., & Vásquez, E. (2014). Seguimiento de trayectorias con un robot móvil de configuración diferencial. *Ing. USBMed*, 26-34.
- Solaque, L. (2015). Sistema de transporte un grupo de robots móviles Desarrollado Para un Grupo de Robots. 264-272.
- Suzuki, & Abe. (1985). Topological Structural Analysis on Digitized Binary Imaged by Border Following.

- Tenezaca, D., Canchignia, C., Aguilar, W. G., & Mendoza, D. (2019). Implementation of Dubin Curves-Based RRT\* Using an Aerial Image for the Determination of Obstacles and Path Planning to Avoid Them During Displacement of the Mobile Robot. En *Smart Innovation, Systems and Technologies* (págs. 205-215). Springer.
- Toibero, J., Soria, C., Roberti, F., & Canali, L. (2008). Control Servo Visual basado en Imagen para navegación estable en pasillos. *Congreso Argentino de Control Automático*, 1-6.
- Traslosheros, M., & Zuñiga, J. M. (2009). Servo control visual para tareas de seguimiento dinámico tridimensional, mediante la utilización de una cámara en un robot Delta. *XXIX Jornadas de Automática*, 69-89.
- Trevai, C., Fukazawa, Y., Ota, J., & Yuasa, H. (2013). Cooperative Exploration of Mobile Robots Using Reaction-diffusion equation on a Graph. *International Conference on Robotics and Automation*, 14-19.
- Tumbaco, D., & Quimbita, W. (2014). Diseño y construcción de un prototipo de robot Delta con implementación de un cortador láser CNC utilizando la Plataforma Robotic Operating System (ROS) para la elaboración de artículos publicitarios. 3-9.
- Unipython*. (2019). Obtenido de <https://unipython.com/transformaciones-geometricas-de-imagenes-con-opencv/>
- Uny Cao, Y., Fukunaga, A., & Kahng, A. (1999). *Cooperative Mobile Robotics: Antecedents and Directions*. Obtenido de University of California, Los Angeles: <https://link.springer.com/article/10.1023/A:1008855018923>
- Valavanis, K., & Vachtsevanos, G. (2015). Handbook of Unmanned Aerial Vehicles. *Springer*, 43-55.
- Valencia, J., Montoya, A., & Ríos, L. (2009). Cinematic model of mobile robot differential type and Navigation from the odometric estimation.

- Vallés, M. (2009). Discrete-Time Modeling and Path-Tracking for a Wheeled Mobile Robot. *Computación y Sistemas 13.2*, 142-160.
- Vilardell, F., Artigues, G., Sanz, J., García-Piquer, Á., & Colomé, J. (2016). Using Robotic Operating System (ROS) to control autonomous observatories. 99-132.
- Villalobos, A., & Gallardo, J. (2018). Diseño e implementación de un observatorio robótico teleoperado basado en ROS. 23-36.
- Zurita, M., Aguilar, W. G., & Victor, E. (2019). Toward the Development of Surveillance and Reconnaissance Capacity in Ecuador: Geolocation System for Ground Targets Based on an Electro-Optical Sensor. En *Smart Innovation, Systems and Technologies* (págs. 123-133). Springer.