



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL
CARRERA DE TELECOMUNICACIONES

Tema:

SISTEMA DE TELEDETECCIÓN AÉREA PARA CONTROL DE
AGENTES PATÓGENOS Y ENFERMEDADES EN CULTIVOS DE
BRÓCOLI CON EL USO DE VISIÓN ARTIFICIAL

Trabajo de titulación modalidad: Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero en Telecomunicaciones

ÁREA: Física y Electrónica

LÍNEA DE INVESTIGACIÓN: Tecnologías de la Información y Sistemas de
Control

AUTOR: Darwin Ronaldo Laura Telenchana

TUTOR: Ing. Elsa Pilar Urrutia Urrutia, Mg.

Ambato – Ecuador

Agosto - 2023

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: SISTEMA DE TELEDETECCIÓN AÉREA PARA CONTROL DE AGENTES PATÓGENOS Y ENFERMEDADES EN CULTIVOS DE BRÓCOLI CON EL USO DE VISIÓN ARTIFICIAL, desarrollado bajo la modalidad Proyecto de Investigación por el señor Darwin Ronaldo Laura Telenchana, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el número 6.3 del respectivo instructivo del reglamento referido.

Ambato, agosto 2023.

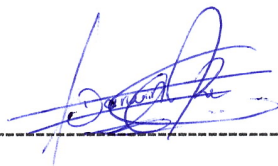
Ing. Elsa Pilar Urrutia, Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación titulado: SISTEMA DE TELEDETECCIÓN AÉREA PARA CONTROL DE AGENTES PATÓGENOS Y ENFERMEDADES EN CULTIVOS DE BRÓCOLI CON EL USO DE VISIÓN ARTIFICIAL, es absolutamente original, autentico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efecto legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto 2023



Darwin Ronaldo Laura Telenchana

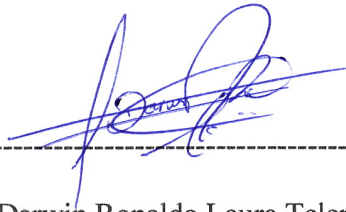
CC. 180529640-5

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, agosto 2023



Darwin Ronaldo Laura Telenchana

CC. 180529640-5

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Darwin Ronaldo Laura Telenchana, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación titulado SISTEMA DE TELEDETECCIÓN AÉREA PARA CONTROL DE AGENTES PATÓGENOS Y ENFERMEDADES EN CULTIVOS DE BRÓCOLI CON EL USO DE VISIÓN ARTIFICIAL, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, agosto 2023.

Ing. Pilar Urrutia Urrutia, Mg.

PRESIDENTE DEL TRIBUNAL

Ing. Franklin Salazar Logroño, Mg.

PROFESOR CALIFICADOR

Ing. Clara Sánchez, Mg.

PROFESOR CALIFICADOR

DEDICATORIA

El presente proyecto de titulación se lo dedico a Dios y a la virgencita del Cisne, por guiarme y darme las fuerzas necesarias para seguir adelante con mis estudios. A mis padres, mi abuelita y mi hermano por ser mi pilar fundamental en toda esta carrera universitaria, soportando cada una de las nostalgias y alegrías que les supe propinar.

Darwin Laura

AGRADECIMIENTO

"Agradezco principalmente a mis padres Alfredo y Elvia quienes supieron ser mi pilar fundamental con sus consejos y apoyo económico, para que mi sueño de terminar mi carrera de ingeniería se pueda cumplir. Un dios les pague a ellos y en conjunto a mi hermano Sebastián quien de igual manera alegraba mis días para no decaer y seguir adelante.

A cada uno de los amigos que he ido formando en el transcurso de la carrera, quienes me han ayudado con cada uno de sus conocimientos para de esta manera lograr mi meta.

A mis docentes, quienes a más de ser mis catedráticos se convirtieron en mis amigos y mi ejemplo a seguir, gracias a los consejos y conocimientos impartidos he podido seguir adelante sin desmayar.

A mi abuelita Transito, que en paz descansa quiero darle mis más grandes agradecimientos por todo lo que ella hizo por mí, desde muy niño con cada uno de sus cuidados y consejos para convertirme en un joven de bien y de muy buenos valores. De igual manera agradecerle de manera muy especial a la persona que me ha brindado un cariño muy sincero y además de su apoyo incondicional en estas últimas etapas de mi carrera universitaria, a mi compañera de vida Alexandra muchas gracias por todo lo vivido hasta el día de hoy. "

Darwin Laura

ÍNDICE GENERAL DE CONTENIDO

APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
Dedicatoria.....	vi
Agradecimiento.....	vii
RESUMEN EJECUTIVO.....	xix
ABSTRACT.....	xx
CAPÍTULO I	1
MARCO TEÓRICO	1
1.1. Tema de investigación.....	1
1.1.1 Planteamiento del problema.....	1
1.2. Antecedentes investigativos	3
1.3. Fundamentación teórica	6
1.3.1 Teledetección	6
1.3.1.1 Plataformas y sensores de detección.....	7
1.3.1.2 Sistema de teledetección	8
1.3.1.3 El Espectro Electromagnético.....	12
1.3.1.4 Sistemas Multiespectrales	13
1.3.1.5 Tipo de imágenes de teledetección	14
1.3.2 Índice de Vegetación	16
1.3.2.1 Fundamentos de la vegetación.....	16
1.3.2.2 Tipos de índices	18
1.3.2.3 Índice de Vegetación de Diferencia Normalizada (NDVI).....	19
1.3.3 Vehículos Aéreos no Tripulados (UAV)	21
1.3.3.1 Drones	22
1.3.3.2 Tipos de drones	23
1.3.3.3 Características de los drones.....	31
1.3.3.4 Componentes de los drones.....	32
1.3.3.5 Vehículos aéreos no tripulados en el campo de la agricultura	34
1.3.3.6 Agricultura de precisión con drones	35

1.3.4	Visión Artificial	36
1.3.4.1	Componentes de un Sistema de Visión Artificial	37
1.3.4.2	Etapas de un sistema de visión artificial	38
1.3.4.3	Tipos de Aprendizaje	39
1.3.4.4	Red Convolutiva YOLO.....	40
1.3.4.5	Makesense.ai	41
1.3.4.6	Teachable Machine	41
1.3.4.7	Herramientas para el procesamiento de imágenes	41
1.3.4.8	Bases de Datos	42
1.3.5	Brócoli	42
1.3.5.1	Características para el cultivo	43
1.3.5.2	Fases de Cultivo	44
1.3.5.3	Agentes Patógenos y enfermedades en los cultivos de brócoli.....	44
1.4.	Objetivos	46
1.4.1.	Objetivo General	46
1.4.2.	Objetivos Específicos	46
CAPÍTULO II.....		47
METODOLOGÍA.....		47
2.1.	Materiales.....	47
2.2.	Métodos.....	48
2.2.1.	Modalidad de investigación.....	48
2.2.2.	Recolección de información	48
2.2.3.	Procesamiento y análisis de datos	48
CAPÍTULO III.....		50
RESULTADOS Y DISCUSIÓN		50
3.1.	Análisis y discusión de los resultados.....	50
3.2	Desarrollo de la propuesta	50
3.2.1	Requerimiento del sistema	50
3.2.2	Esquema General del sistema	51
3.2.3	Análisis de dispositivos.....	54

3.2.4 Selección de los elementos hardware.....	56
3.2.4.1 Elementos hardware Etapa 1.....	56
3.2.5 Selección de elementos software	64
3.2.5.1 Elementos software Etapa 2.....	65
3.2.5.2 Elementos software Etapa 3.....	65
3.2.6 Diseño del sistema	66
3.2.7 Diseño e implementación del sistema general	68
3.2.8 Pruebas de funcionamiento y resultados	152
3.2.9 Confiabilidad del prototipo	172
3.2.10 Presupuesto	174
CAPÍTULO IV	177
CONCLUSIONES Y RECOMENDACIONES	177
4.1. Conclusiones	177
4.2. Recomendaciones.....	179
ANEXOS	188

ÍNDICE DE FIGURAS

Figura 1. Teledetección desde un avión (INTA).....	6
Figura 2. Globos aéreos como plataformas de detección	7
Figura 3. Aeronaves no tripuladas como plataforma de detección	8
Figura 4. Teledetección espacial	8
Figura 5. Formas de detección en un sistema de teledetección: (i) Reflexión; (ii) Emisión; (iii) Reflexión - Emisión	9
Figura 6. Componentes de un sistema de teledetección.....	10
Figura 7. Sensores Pasivos	10
Figura 8. Sensores Activos	12
Figura 9. Espectro Electromagnético	12
Figura 10. Sistemas Multiespectrales	14
Figura 11. Estructura de imágenes multiespectrales	14
Figura 12. Imagen pancromática B/N	15
Figura 13. Imagen fusionada (PS)	15
Figura 14. Imagen Estéreo	16
Figura 15. Espectro de absorción para la clorofila.....	17
Figura 16. Reflectancia espectral de una hoja viva.....	17
Figura 17. Contraste espectral de vegetación sana contra vegetación seca	18
Figura 18. Cálculo de NDVI	20
Figura 19. Rangos de vegetación en NDVI	20
Figura 20. Ejemplo de NDVI	21
Figura 21. NDVI a causa de la reflectancia	21
Figura 22. Dron (UAV)	22
Figura 23. Dron profesional de ala fija	24
Figura 24. Dron multirotor.....	24
Figura 25. Dron Tricóptero	25
Figura 26. Configuración de motores en tricóptero	26
Figura 27. Dron cuadricóptero	27
Figura 28. Configuración de motores en cuadricóptero.....	27
Figura 29. Configuración de motores en cuadricóptero, forma X	28
Figura 30. Dron Hexacóptero	29
Figura 31. Configuración de motores en cuadricóptero	29

Figura 32. Dron octocóptero	30
Figura 33.. Configuración de motores en cuadricóptero.....	31
Figura 34. Sistema de propulsión	32
Figura 35. Motor	32
Figura 36. Variador	33
Figura 37 Hélices	33
Figura 38. Estructura de LIPO	34
Figura 39. Controlador de vuelo	34
Figura 40. Formas de extracción de información, plataformas y sensores útiles en los UAVs	35
Figura 41. Agricultura de precisión mediante el uso de teledetección	36
Figura 42. Estructura de un sistema de visión artificial	37
Figura 43. Componentes del Sistema de Visión Artificial	37
Figura 44. Etapas de un Sistema de Visión Artificial	39
Figura 45. Estructura de una red convolucional YOLO	40
Figura 46. Estructura YOLOv5	41
Figura 47. Brócoli Legacy	43
Figura 48. Ciclo de desarrollo del brócoli	44
Figura 49.. Agentes Patógenos en las plantas de brócoli	45
Figura 50. Enfermedades presentes en los cultivos de brócoli	46
Figura 51. Esquema General del Sistema con cada una de sus etapas.....	52
Figura 52. Etapa 1 – Sistema de Adquisición de datos	53
Figura 53. Etapa 2 – Sistema de Control y Procesamiento de datos.....	53
Figura 54. Etapa 3 – Sistema de visualización y Almacenamiento de datos	54
Figura 55. Dron Realacc x210Pro FPV	58
Figura 56. Microordenador Raspberry Pi 4	60
Figura 57. Módulo SIM 808 GPRS/GSM	61
Figura 58. HUAWEI E586.....	62
Figura 59. AP.25E GPS/GALILEO	64
Figura 60. Diseño del sistema de teledetección	67
Figura 61. Diseño Base superior del dron.....	69
Figura 62. Diseño base inferior del dron.....	70
Figura 63. Diseño base de la batería	71

Figura 64. Diseño de los brazos exteriores	71
Figura 65. Diseño parte central del brazo exterior	72
Figura 66. Diseño de la parte de enganche del brazo exterior	72
Figura 67. Diseño de motores	73
Figura 68. Diseño de hélices	73
Figura 69. Diseño de cámara.....	74
Figura 70. Diseño de Antena receptora de señal.....	74
Figura 71. Diseño 3D renderizado del dron	75
Figura 72. Frame y componentes de ensamblaje	76
Figura 73. Montaje de Frame	77
Figura 74. Ubicación piloto de los componentes	77
Figura 75. Diagrama de conexión de la placa de control.....	78
Figura 76. Conexión de la placa controladora en el frame	79
Figura 77. Diagrama de conexión ESC.....	79
Figura 78. Conexión de la ESC en el frame del dron.....	80
Figura 79. Conexión de motores en cada extremo de los brazos del dron.....	80
Figura 80. Conexión y montaje de cámara FPV	81
Figura 81. Diagrama de conexión de VTX 801	81
Figura 82. Conexión del módulo VTX	82
Figura 83. Ubicación de la batería en su base	83
Figura 84. Emisora - Receptora y Gafas FPV.....	83
Figura 85. Dron Realaccx210Pro completo	84
Figura 86. Puerto de conexión de DRON COM4	85
Figura 87. Reconocimiento de posición del dron en el software Betaflight	85
Figura 88. Activación de puertos	86
Figura 89. Alineación de placas y sensores	87
Figura 90. Configuración de Baliza Dshot.....	87
Figura 91. Ajuste de PID Betaflight.....	88
Figura 92. Receptor Betaflight.....	89
Figura 93. Configuración de Modos de vuelo Betaflight.....	90
Figura 94. Motores Betaflight	90
Figura 95. OSD Betaflight	91
Figura 96. Marco Inercial de UAV	92

Figura 97. Diseño 3D de un dron cuadricóptero en AUTOCAD.....	93
Figura 98. Diseño posicionado en software FUSION 360.....	94
Figura 99. Declaración de variables.....	95
Figura 100. Declaración de funciones para los 3 ejes de coordenadas.....	95
Figura 101. Posicionamiento del UAV dentro de un plano 3D en Matlab.....	96
Figura 102. Declaración de condiciones iniciales y posición final.....	97
Figura 103. Configuración de frame grafico.....	97
Figura 104. Trayectoria del UAV con un ángulo de 45°.....	98
Figura 105. Trayectoria del UAV con un ángulo de 90°.....	98
Figura 106. Matriz jacobiana para control de trayectoria del UAV.....	99
Figura 107. Trayectoria Libre.....	99
Figura 108. Trayectoria en forma octogonal.....	100
Figura 109. Trayectoria en forma circunferencial.....	100
Figura 110. Trayectoria en forma helicoidal.....	101
Figura 111. Trayectoria de referencia.....	102
Figura 112. Modelación del subsistema.....	103
Figura 113. Control PID de la trayectoria del dron.....	104
Figura 114. Estabilización.....	104
Figura 115. Parámetros de entrada para crear controlador PID en Matlab.....	105
Figura 116. Inicialización de variables.....	106
Figura 117. Definición de errores.....	106
Figura 118. Proceso para simular el PID.....	107
Figura 119. Actualización de estados en Matlab.....	107
Figura 120. Visualización de resultados.....	108
Figura 121. Resultado de la simulación de Controlador PID para trayectoria del dron	108
Figura 122. Comportamiento PID antes de despegue.....	109
Figura 123. Comportamiento PID en ascenso de dron.....	110
Figura 124. Comportamiento PID del dron bajo una perturbación.....	110
Figura 125 Implementación de modulo AP.25E GPS/GALILEO en el dron.....	111
Figura 126. Inicio del software Mission Planer de ArduPilot.....	112
Figura 127. Ultima posición geográfica del dron.....	112
Figura 128. Modo de vuelo auto establecido.....	113

Figura 129. Parámetro de velocidad establecido en 200cm/s	114
Figura 130. Punto de partida HOME	114
Figura 131. Puntos estratégicos establecidos a lo largo de la superficie	115
Figura 132 Plan de vuelo grabado en la placa de control	115
Figura 133. Diseño 3D de soporte de cámara	116
Figura 134. Implementación del case protector en el dron	117
Figura 135. Diagrama de flujo del proceso de captura de imágenes	118
Figura 136. Librerías utilizadas.....	119
Figura 137. Variables globales.....	120
Figura 138. Uso de archivo entrenado en yolov	121
Figura 139. Creación de ventanas principales del sistema.....	121
Figura 140. Funciones de conexión con la base de datos	122
Figura 141. Abrir página principal y obtener registros de la bdd	123
Figura 142. Verificación de existencia de alguna petición en el servidor	124
Figura 143. Actualización de la tabla de datos	126
Figura 144. Abrir la ventana de datos y cargar la imagen	127
Figura 145. Datos mostrados en el software ArduPilot	129
Figura 146. Código para ubicar en la interfaz la velocidad del dron	129
Figura 147. Obtención de datos de altura del dron	130
Figura 148. Obtención del dato de la velocidad del viento de la página de Microsoft	130
Figura 149. Obtención del código html de una página determinada.....	131
Figura 150. Código para ubicar en la interfaz la velocidad del viento	131
Figura 151. Función para cambiar el valor de la velocidad el viento	132
Figura 152. Código para ejecutar el proceso de mostrar la velocidad del viento cada minuto	132
Figura 153. Configuración de un widget de ttk.Treeview	133
Figura 154. Interfaz del sistema de teledetección aérea.....	134
Figura 155. Pantalla principal del sistema de teledetección aérea	135
Figura 156. Tabla con lista de capturas.....	135
Figura 157. Botón visualizar	136
Figura 158. Imágenes recolectadas para el entrenamiento del algoritmo YOLO v5x	137

Figura 159. Etiquetado de imágenes con Make Sense	137
Figura 160. Establecimiento de etiquetas.....	138
Figura 161. Etiqueta del brócoli saludable.....	138
Figura 162. Etiqueta 2-Estrés Hídrico.....	139
Figura 163. Etiqueta 3- Mala hierba	139
Figura 164. Cargar etiquetas desde Google drive	140
Figura 165. Descomprimir carpeta que contiene las etiquetas.....	140
Figura 166. Clonar e instalar requerimientos yolov	141
Figura 167. Ejecución del entrenamiento.....	141
Figura 168. Implementación del sistema de geolocalización.....	142
Figura 169. Diseño 3D de frame para sistema de geolocalización	142
Figura 170. Implementación de frame sobre el dron	143
Figura 171. Capturar imagen del dron	144
Figura 172. Petición para capturar imagen.....	144
Figura 173. Conexión a la cámara y toma la captura de la foto.....	145
Figura 174. URL para subir los datos de imagen con la localización.....	145
Figura 175. Dron implementado con el sistema.....	147
Figura 176. Instalación de Firmware	148
Figura 177. Selección de tipo de vehículo, en este caso tipo COPTER.....	148
Figura 178. Calibración de ejes x,y,z.....	149
Figura 179. Calibración del dron mediante posiciones mostradas en el programa.	149
Figura 180. Calibración de nivel	150
Figura 181. Planeación de vuelo tipo cuadrícula	150
Figura 182. Distancias recorridas en un plan de vuelo	151
Figura 183. Lote de terreno seleccionado para la prueba de vuelo	153
Figura 184. Plan de vuelo totalmente estructurado	154
Figura 185. Elevación del dron y ejecución de la misión de vuelo.....	155
Figura 186. Variación de altura de 10 metros en tramos 4 al 5	155
Figura 187. Arribo del dron al punto HOME y tiempo empleado	156
Figura 188. Planes de vuelo programados: A) Etapa 1, B) Etapa 2, C)Etapa 3 y D) Etapa 4.....	164
Figura 189. Matriz de confusión: Background horizontal – Falsos positivos y Background vertical – Falsos negativos.....	170

Figura 190. Muestras positivas en el sistema.....	171
Figura 191. Muestras negativas en el sistema.....	172
Figura 192. Comparación de coordenadas de Posicionamiento.....	174

ÍNDICE DE TABLAS

Tabla 1 Características de los principales sensores utilizados en la teledetección pasiva.....	11
Tabla 2. Historia de los drones (UAV).....	22
Tabla 3. Comparativa entre drones de ala fija y multirotor.....	24
Tabla 4. Ventajas y desventajas de los tricóptero	26
Tabla 5. Ventajas y desventajas de los cuadricópteros	28
Tabla 6. Ventajas y desventajas de los hexacópteros	29
Tabla 7. Agentes Patógenos en cultivos de brócoli.....	44
Tabla 8.. Enfermedades presentes en los cultivos de brócoli.....	45
Tabla 9. Drones con cámaras multiespectrales dedicados a la agricultura tecnificada en Ecuador.....	55
Tabla 10. Características de drones de fácil ensamblaje	56
Tabla 11. Características de microordenadores comerciales.....	59
Tabla 12. Características técnicas de cámaras de alta definición.....	62
Tabla 13. Especificaciones del Frame Martial II	75
Tabla 14. Peso de los componentes externos acoplados al dron.....	146
Tabla 15. Resultados de distancias obtenidas	151
Tabla 16. Relación entre terreno inspeccionado vs tiempo de vuelo	156
Tabla 17. Etapa 1 de cultivo de brócoli - Trasplante.....	157
Tabla 18. Valoración de los resultados obtenidos en la etapa 1	159
Tabla 19. Etapa 2 de cultivo de brócoli – Inducción floral	159
Tabla 20. Valoración de los resultados obtenidos en la etapa 2.....	160
Tabla 21. Etapa 3 de cultivo de brócoli – Fructificación	161
Tabla 22. Valoración de los resultados obtenidos en la etapa 3.....	162
Tabla 23. Etapa 4 de cultivo de brócoli - Cosecha.....	162
Tabla 24. Valoración de los resultados obtenidos en la etapa 4.....	164

Tabla 25. Entrenamiento del algoritmo con 100 épocas	165
Tabla 26. Entrenamiento del algoritmo con 200 épocas	165
Tabla 27. Datos de geolocalización recolectados al momento de cada captura de imagen.....	166
Tabla 28. Resultados de detección obtenidos en cada etapa del cultivo	167
Tabla 29. Comparación de datos de posicionamiento obtenidos por el prototipo y el GPS de cámara GoPro.....	173
Tabla 30. Presupuesto material	174
Tabla 31 Presupuesto total	176

RESUMEN EJECUTIVO

El brócoli es uno de los productos agrícolas en Ecuador con mayor índice de importaciones a nivel mundial, debido a esto el desarrollo del cultivo debe ser inspeccionado de manera rutinaria para contrarrestar los maleficios que perjudiquen su producción como agentes patógenos o enfermedades y conservar su alto índice de calidad productiva. Por esta razón el presente proyecto investigativo se orienta a realizar inspecciones de cultivos de brócoli por vía aérea con planificaciones de vuelo preprogramadas que ayuden al análisis del estado de salud del cultivo para ser tratado a tiempo. El diseño e implementación del sistema de teledetección aérea para control de agentes patógenos y enfermedades en cultivos de brócoli es desarrollado mediante el uso del algoritmo YOLO v5x de visión artificial para el aprendizaje profundo del sistema en diversas circunstancias de producción.

El sistema de teledetección aérea utiliza un dron que ayuda al desplazamiento de los sensores activos y pasivos que interactúan en el sistema. Este dron es de tipo autónomo gracias a la implementación de un módulo GPS en su placa de control de vuelo, lo cual ayuda a la planificación de vuelos preprogramados de tipo cuadrícula. Este sistema recibe el análisis del estado de salud del cultivo de brócoli mediante la captura de imágenes a través de una cámara de alta definición incorporada al dron. Cada imagen es capturada en un lapso de tiempo de 2 segundos para realizar su procesamiento respectivo a través de una interfaz gráfica programada en Python.

La imagen consta de datos de posicionamiento como latitud y longitud obtenidas mediante un sistema de geolocalización que fue implementado en el dron. Estos datos ayudan a localizar el punto específico donde fueron capturadas las imágenes e interpretar los resultados obtenidos en el análisis del cultivo para cada una de ellas. Los resultados del procesamiento de las imágenes son guardados en una base de datos, para el entrenamiento autónomo del algoritmo de visión artificial y mejora del prototipo en detección de falsos positivos y falsos negativos.

Palabras clave: Teledetección, autónomo, dron, YOLO, Python, geolocalización

ABSTRACT

Broccoli is one of the agricultural products in Ecuador with the highest rate of imports worldwide, due to this the development of the crop must be routinely inspected to counteract the curses that harm its production such as pests or diseases and preserve its high rate of productive quality. For this reason, the present research project is oriented to carry out inspection of broccoli crops by air with pre-programmed flight plans that help to analyze the health status of the crop to be treated on time. The design and implementation of the aerial remote sensing system for the control of pathogens and diseases in broccoli crops is developed through the use of the YOLO v5x artificial vision algorithm for deep learning of the system in various production circumstances.

The aerial remote sensing system uses a drone that helps to move the active and passive sensors that interact in the system. This drone is autonomous thanks to the implementation of a GPS module on its flight control board, which helps planning grid-type pre-programmed flights. This system receives the analysis of the health status of the broccoli crop by capturing images through a high-definition camera incorporated into the drone. Each image is captured in a time frame of 10 seconds to carry out its respective processing in real time through a graphical interface programmed in Python.

The image consists of positioning data such as altitude and latitude obtained through a geolocation system that was implemented in the drone. These data help to locate the specific point where the images were captured and interpret the results obtained in the analysis of the culture for each one of them. The results of the image processing are stored in a database, for the autonomous training of the artificial vision algorithm and improvement of the prototype in the detection of false positives and false negatives.

Keywords: Remote sensing, autonomous, drone, YOLO, Python, geolocation

CAPÍTULO I

MARCO TEÓRICO

1.1. Tema de investigación

Sistema de teledetección aérea para control de agentes patógenos y enfermedades en cultivos de brócoli con el uso de visión artificial

1.1.1 Planteamiento del problema

En los últimos años, la agricultura ha desarrollado un rol esencial en el progreso económico de los países a nivel mundial, sobre todo en los países en desarrollo, porque la mayoría de su población depende de ella para su subsistencia [1]. Según un estudio realizado por el Global Hunger Index “GHI” menciona que, la productividad del sector agrícola a nivel mundial debe incrementarse en un 1.75% por año, para de esta manera cubrir las necesidades de cada uno de los habitantes, los cuales se calcula serán de 9700 millones en el año 2050 [2]. Debido a la pandemia global, los agricultores llegaron a experimentar pérdidas de un 12.5% en el cultivo de todos sus productos, mientras que en el ámbito de exportaciones se experimentó un margen de pérdida de 25.6% del total de productos. Las causas principales que llevaron a tener niveles de pérdidas tan grandes se mencionan en la encuesta de Retos para la agricultura familiar en el contexto del COVID-19, entre estos podemos encontrar la falta de control de agentes patógenos debido al difícil acceso de insumos agrícolas y por ende la baja calidad del producto [3]. Uno de los productos que cayó en deterioro de producción durante los años de pandemia es el brócoli. Según la información presentada en la base de datos estadísticos corporativos de la Organización para la Agricultura y la Alimentación “FAOSTAT”, la producción a nivel mundial de brócoli antes de pandemia fue de 25,531,274 toneladas, cosechadas en lotizaciones de 1,357,186 hectáreas, por lo que el rendimiento promedio quedó en 18.8 toneladas por hectárea [4]. Pero actualmente estos datos han decaído en un 2.1% en el mercado agrícola internacional, esto debido al bajo control de agentes patógenos e insectos en estos años de pandemia.

En Ecuador, existe una gran cantidad de cultivos de brócoli, siendo las más representativas las provincias de Cotopaxi, Chimborazo, Tungurahua, Imbabura y Pichincha [5]. Entre enero y junio de 2021, este alimento se ubicó entre los 10 principales productos de exportación del país, ubicándose en octavo lugar, por encima de los aceites vegetales, según la Federación Ecuatoriana de Exportadores (Fedexpor) [6]. En Ecuador, apenas 2000 toneladas son para consumo local; estas se obtienen de la cosecha de unas 2000 hectáreas, mientras que en la venta de brócoli al mundo se obtiene de aproximadamente 9900 toneladas en 9000 hectáreas, siendo los principales destinos de exportación la UE con el 34%, EE.UU. y Canadá con el 36% y el 30% a Japón [7].

Actualmente, la producción en toneladas de brócoli está en auge, luego de la pandemia se ha buscado mejorar las técnicas de cultivo, cuidando el producto mediante la intervención de tecnología aplicada al campo, debido a que la calidad del producto ecuatoriano es muy buena. En Tungurahua, la producción de brócoli se mantiene constante con un 16% de la zona productiva, sin embargo, las fluctuaciones productivas cada año son inusuales. Entre 2016 y 2017, por ejemplo, pasa de producir 392 toneladas a producir 1700 toneladas, crecimiento de más del mil por ciento, mientras que entre 2018 y 2019 la producción disminuye 95% debido a los inicios de la pandemia global [8]. Según el Ministerio de Agricultura y Ganadería, actualmente se produce alrededor de 129764 toneladas en una superficie de 7063 hectáreas. Sin embargo, la producción disminuye paulatinamente debido a factores internos y externos como: agentes patógenos y enfermedades en un 45%, heladas en un 15% y otras causas en un 40%, obteniendo una pérdida de superficie de 112 hectáreas aproximadamente [9]. Por ello, el objetivo de este proyecto es cuidar el producto agrícola de agentes patógenos y enfermedades dañinas evitando el bajo rendimiento por la escasa utilización de nuevas tecnologías en los diversos cultivos, puesto que la mayor parte de agricultores basan su trabajo en un modelo de agricultura tradicional y no en una agricultura tecnificada que permita aumentar los rendimientos y bajar los costos de producción [10].

1.2. Antecedentes investigativos

Después de examinar varios repositorios y fuentes bibliográficas a nivel nacional e internacional, se identificaron investigaciones relacionadas con el tema de investigación, las cuales fueron de gran utilidad como referencia para el desarrollo de este proyecto. Entre estos trabajos se incluyen:

En la investigación desarrollada por Fouquet [11] en la Universidad de Málaga, en el año 2021 con el tema “Inteligencia Artificial aplicada a la agricultura de precisión. Control de hongos en la planta de tomate”, tuvo como objetivo principal desarrollar una herramienta basada en Machine Learning para predecir el grado de severidad de un hongo en los cultivos de tomate. Para el desarrollo de esta investigación, se efectuó un preprocesamiento de datos que incluía índices de vegetación específicos de las plantas. La implementación se llevó a cabo utilizando una red neuronal de tipo Feed-Forward, programada en Python con la ayuda de interfaces como Anaconda, NVIDIA y GeForce Experience. Estas herramientas facilitaron la programación al ser legibles y tener códigos limpios, lo que permitió una programación versátil, multiplataforma y multiparadigma. Como resultado, se logró procesar los datos y generar gráficas para estimar el grado de severidad del hongo *Phytophthora infestans* en los cultivos de tomate. Se utilizaron índices de vegetación específicos después de la implantación del patógeno mediante los cuales las predicciones realizadas por el modelo mostraron un alto grado de fiabilidad, con un porcentaje de precisión del 90.67% en cada una de las predicciones realizadas. [11].

En la investigación desarrollada por Román y Ruiz [12] en la Universidad de Piura, en el año 2021, con el tema “Detección de macronutrientes y enfermedades en campos de cultivo de banano orgánico con Machine Learning”, donde se expone la importancia de Machine Learning para la agricultura de precisión. El propósito de esta investigación es reducir las pérdidas en las plantaciones de banano orgánico debido a la deficiencia de macronutrientes y la aparición de enfermedades. Para lograrlo, se utilizó una combinación de tres algoritmos de clasificación híbridos (Support Vector Classifier, K-Nearest Neighbors y Random Forest) y redes neuronales convolucionales que incorporan filtros convolucionales con una arquitectura predominante de 11 capas y VGG16. Se evaluaron estos algoritmos y

redes neuronales convolucionales utilizando métricas como la precisión (accuracy), el puntaje F1 (f1 score) y la matriz de confusión. Se utilizaron dos conjuntos de datos de imágenes de hojas de banano orgánico, uno para la detección de deficiencia de macronutrientes y otro para enfermedades. Los resultados muestran que el algoritmo que mejor se desempeña en ambos conjuntos de datos, obteniendo los mejores resultados en precisión, puntaje F1 y matriz de confusión, es Random Forest. Le sigue el Support Vector Classifier (SVC), luego la red neuronal convolucional (CNN) y finalmente K-Nearest Neighbors (KNN). La diagonal de la matriz de confusión para el algoritmo Random Forest muestra los valores más altos, lo que demuestra una clasificación correcta tanto en el conjunto de datos de enfermedades como en el de deficiencia de macronutrientes [12].

En la investigación desarrollada por Roldan, Roshan y Sánchez [13], publicada en la revista *Research in Computing Science* en el año 2019, con el tema “Detección de enfermedades en el sector agrícola utilizando Inteligencia Artificial” se realizó una investigación con el objetivo de detectar enfermedades en los cultivos agrícolas, utilizando técnicas de inteligencia artificial. Se destacan el uso de tecnologías emergentes como el aprendizaje automático y el reconocimiento de patrones. Para lograr esto, se adquirieron imágenes utilizando cámaras digitales, lo que permitió entrenar la inteligencia artificial en el reconocimiento de enfermedades en los cultivos agrícolas. Se utilizaron diversos algoritmos de aprendizaje automático, como Fuzzy logic, KNN (K-Nearest Neighbors) y CNN (Convolutional Neural Network). Entre estos algoritmos, se observó que la CNN tuvo un rendimiento destacado, aunque con un alto costo computacional. Este enfoque se destaca por su mayor precisión, pero también puede requerir una cantidad considerable de imágenes para obtener resultados confiables. Los resultados obtenidos indican que el uso de redes neuronales convolucionales es la mejor opción para detectar enfermedades en los cultivos agrícolas. Sin embargo, la principal limitación actual radica en la disponibilidad de grandes cantidades de datos para entrenar de manera efectiva la red neuronal [13].

En la investigación desarrollada por Calderón y Hurtado [14], en el año 2019, en la Universidad Distrital Francisco José de Caldas, se llevó a cabo la investigación

titulada, “Machine learning en la detección de enfermedades en plantas”, en donde se expone la relevancia a nivel tecnológico del uso del machine learning en la identificación de enfermedades en plantas. Este estudio aborda diversas técnicas de reconocimiento que permiten identificar y diagnosticar agentes patógenos en plantas y cultivos. Estas técnicas son fundamentales para tomar decisiones y acciones que eviten la propagación de enfermedades. Se discuten dos enfoques para la detección de enfermedades en plantas: los métodos manuales y la aplicación de Machine Learning. El uso de Machine Learning resulta más factible debido a su capacidad de procesamiento de imágenes, lo cual permite detectar características específicas de enfermedades en tallos, ramas, hojas y flores de las plantas. Esto a su vez facilita la creación de una red de entrenamiento con una amplia variedad de muestras, lo que mejora la precisión en la detección de enfermedades. Los algoritmos de Machine Learning como de Deep Learning, como el bosque aleatorio (Random Forest), redes neuronales artificiales, máquinas de vectores de soporte (SVM), lógica difusa y redes neuronales convolucionales, han sido utilizados para aumentar la tasa de reconocimiento, fiabilidad y precisión en el proceso de detección de enfermedades en plantas. El resultado obtenido depende del enfoque de estudio y la aplicación específica, lo que determinará qué algoritmo de Machine Learning se adapte mejor a las condiciones del estudio, logrando así un rendimiento y precisión superiores en la identificación de enfermedades. Es importante tener en cuenta que cada planta tiene sus propias enfermedades, las cuales pueden variar según la ubicación geográfica y las condiciones específicas de cada planta. Por lo tanto, es necesario considerar estas características individuales al seleccionar el enfoque y algoritmo de detección de enfermedades más adecuados. [14].

De igual manera, en la investigación desarrollada por Bautista, Constante, Gordon y Mendoza [10], publicada en la revista INFOCIENCIA, en el año 2018 se presentó el tema “Diseño e implementación de un sistema de visión artificial para análisis de datos NDVI en imágenes espectrales de cultivos agrícolas obtenidos mediante una aeronave pilotada remotamente”, en el país Ecuador. En este artículo se hizo referencia al Índice de Vegetación de Diferencia Normalizada (NDVI), el cual se utiliza ampliamente en el campo de la agricultura como un indicador de la salud de las plantas. Este índice permite diferenciar las zonas vegetadas de otras coberturas y

distinguir la vegetación sana de la senescente o bajo estrés. Se presentó una técnica de visión artificial aplicada a la detección de características en el desarrollo de cultivos. Esta técnica se basó en un sistema de visión por computadora que utilizó imágenes espectrales adquiridas mediante una cámara Survey 2 NDVI Red + NIR, la cual fue adaptada a una plataforma aérea (drone). Las imágenes fueron procesadas utilizando el lenguaje de programación Matlab. Los resultados obtenidos incluyeron la selección de aspectos cualitativos y cuantitativos que permitieron ampliar el área de interés y mostrar los resultados del NDVI en una escala de colores, así como los valores obtenidos de la selección de la Región de Interés (ROI). [10].

Las investigaciones bibliográficas encontradas contribuirán al progreso del proyecto, dado que se ha establecido a través de la revisión de investigaciones anteriores que las redes neuronales y la inteligencia artificial son fundamentales para agilizar la detección de enfermedades y agentes patógenos en los cultivos agrícolas.

1.3. Fundamentación teórica

1.3.1 Teledetección

La teledetección es la ciencia o el arte mediante la cual se obtiene teorías o conceptos de la detección de objetos mediante sus características relevantes como forma y tamaño, estos datos se procesan mediante un sistema de sensores que almacenan la información y la registran luego de detectarlos [15]. Un sensor es un dispositivo tecnológico cuya función es captar imágenes a diferentes distancias de la superficie terrestre y que se moviliza en una plataforma aérea. Su método de obtener información parte la captura de imágenes en distintas partes del espectro consideradas como canal o banda [16].

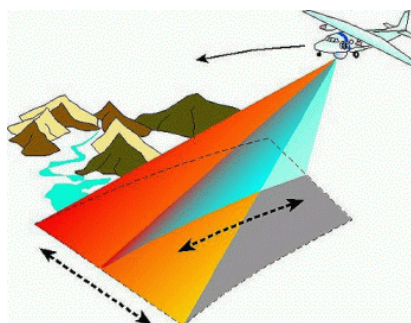


Figura 1. Teledetección desde un avión (INTA) [16]

1.3.1.1 Plataformas y sensores de detección

Plataformas de detección

Los satélites son considerados como las principales plataformas de detección quienes son los encargados de transportar los dispositivos (sensores) que se necesitan para la adquisición de información mediante las capturas de imágenes aéreas [17].

Los tipos de plataformas existentes hasta la actualidad son las siguientes [15]:

GLOBOS: Son utilizados para la fotografía aérea, aunque su uso es limitado debido a que no puede elevarse a muy grandes alturas para capturar superficies terrestres extensas como se puede apreciar en la Figura 2.



Figura 2. Globos aéreos como plataformas de detección [15]

AVIONES: Los aviones son la plataforma de detección usados en su mayoría actualmente, debido a facilitar la adquisición de datos y a la interacción usuario-máquina que es de forma directa o indirecta a través de radio controles. Permite la captura de imágenes a diferentes alturas, ayudando al análisis de grandes superficies terrestres. La elección de una altura propicia ayuda en el tratamiento de la imagen gracias a su buen enfoque y resolución, Figura 3.

Estas aeronaves tienen la ventaja de equiparse con diferentes accesorios como cámaras FPV, cámaras infrarrojas, cámaras multispectrales entre otros sensores adicionales para la obtención de información. Las desventajas en estas plataformas de detección son su alto costo y la dependencia de los factores climáticos dado que de esto depende su estabilidad en el aire.

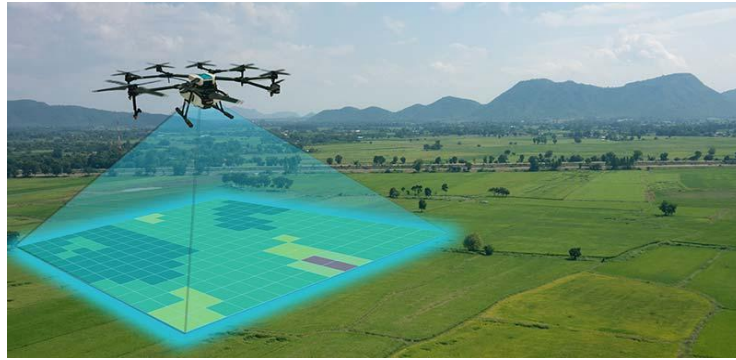


Figura 3. Aeronaves no tripuladas como plataforma de detección [18]

SATÉLITES: Los satélites rodean a la Tierra para una cobertura total en la obtención de información, Figura 4. No poseen tripulación a bordo debido a su percepción remota controlada desde una base en la superficie terrestre.

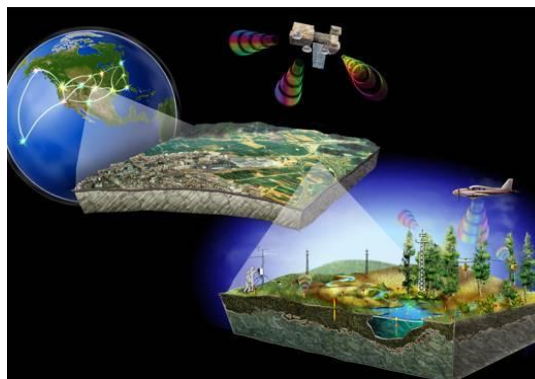


Figura 4. Teledetección espacial [19]

1.3.1.2 Sistema de teledetección

El mercado de la teledetección ha crecido en los últimos tiempos, debido a la fácil obtención de información por el análisis de las imágenes capturadas mediante los sensores transportados a través de las diferentes plataformas. Estos sensores van mejorando la resolución de imágenes a distintas bandas o frecuencias [20].

Los sistemas de teledetección están conformados por tres elementos necesarios: el sensor, el objeto a observar y un flujo de energía que correlaciona a ambos. El flujo de energía que es emitido por el objeto, puede ser por motivo de la reflexión de la luz solar, algún tipo de energía emitida por el objeto o también por el sensor. Dependiendo del flujo de luz podemos obtener información de las siguientes maneras: emisión, por reflexión y por emisión – reflexión [21], figura 5.

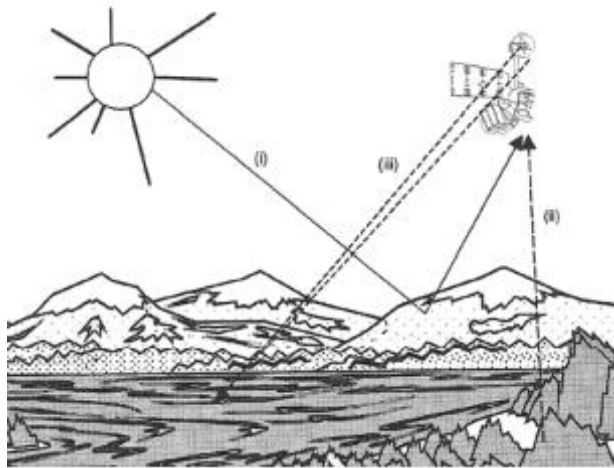


Figura 5. Formas de detección en un sistema de teledetección: (i) Reflexión; (ii) Emisión; (iii) Reflexión - Emisión [21]

Elementos de un sistema de teledetección

Un sistema de teledetección se estructura de los siguientes elementos [22], figura 6:

- **Fuente de energía**

Es causada por la radiación electromagnética que va a captar el sensor. Esta fuente de energía puede ser pasiva (luz solar) o activa (emitido por el sensor).

- **Cubierta terrestre**

Son los rasgos naturales como masas de vegetación, suelos, agua o los causados por el hombre como construcciones; las cuales reflejan o emiten radiación energética acorde a sus propiedades.

- **Sistema sensor**

Esta parte del sistema está compuesta por el sensor y la plataforma que lo transporte. El sensor este encargado de captar la información para codificar, grabar y enviar a un sistema de recepción.

- **Sistema de recepción-comercialización**

Capta la información que recibe el sistema sensor, guarda y procesa estos datos para distribuirlos a un usuario.

- **Intérprete**

Es la persona experta encargada de incluir un valor productivo a la información recolectada como es la generación de mapas temáticos.

- **Usuario final**

Es la persona que se beneficia del documento final y hacer uso de este para su estudio sobre sus consecuencias y ventajas.

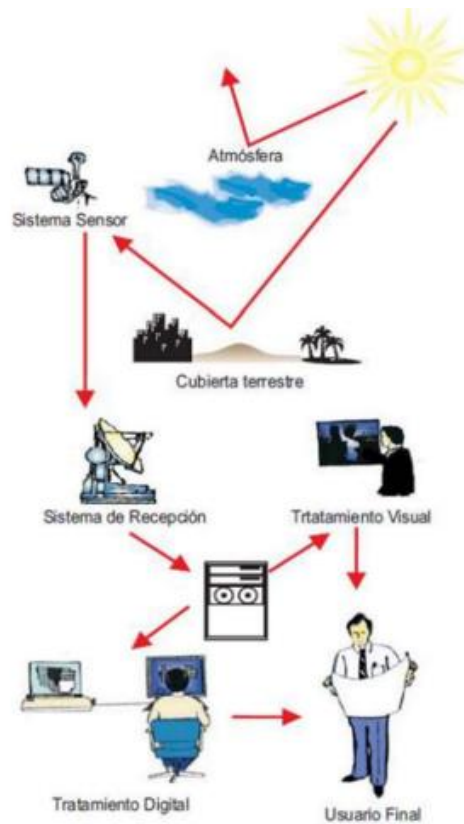


Figura 6. Componentes de un sistema de teledetección [22]

Tipos de sensores de detección

Existen dos tipos de detección: detección pasiva y detección activa [15] [23]:

Los sensores de detección pasivos. - Estos sensores detectan la radiación de luz natural que refleja o emite un objeto o área definida, figura 7. Algunos ejemplos pueden ser la fotografía y los infrarrojos.

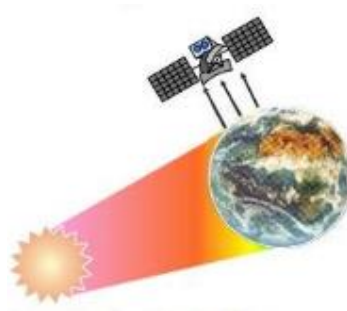


Figura 7. Sensores Pasivos [23]

La manera en la que se realiza la teledetección por medio de estos sensores es mediante cuatro tipos de resolución, como se muestra en la Tabla 1:

- Resolución espacial (tamaño de pixel)
- Resolución temporal (tiempo en captar dos imágenes en el mismo lugar)
- Resolución espectral (reconociendo del número de bandas disponibles)
- Resolución radiométrica (capta el número de intervalos de intensidad)

Tabla 1 Características de los principales sensores utilizados en la teledetección pasiva.

Sensor	Resolución Espacial	Tamaño de imagen	Resolución Temporal	Resolución Radiométrica	Resolución Espectral
Meteosat	2500 m	Toda la esfera	0.5 horas	256 ND	1Vis 1Ir 1It
NOAA AVHRR	1100 m	2700x2700 Km	12 horas	1024 ND	2Vis 1Ir 1It
Landsat TM	30 m	185x185 Km	16 días	256 ND	3Vis 3Ir 1It
SPOT HRV	20 m	60x60 Km	20 días	256 ND	2Vis 1Ir
SPOT Vegetation	1150 m	2200x200 Km	1 día	1024 ND	2Vis 2Ir
MODIS	250–100 m	2330x2330 Km	1	1024 ND	36 bandas
IKONOS	4 m	100x100 Km	A petición	2048 ND	3Vis 1Ir

Fuente: [24]

Los sensores de detección activos. - Estos sensores tienen la capacidad de iluminar algún objeto con una fuente de radiación propia, en donde la iluminación procedente de este provocará que se emita radiación (fluorescencia) en un objeto o a su vez provocar que se refleje la radiación emitida por el sensor, como se observa en la figura 8. Los sensores de detección activos son útiles con frecuencia cuando la radiación natural en una banda particular del espectro no llega a ser suficiente para iluminar completamente a un objetivo, esto se entiende mejor como la radiación natural por un nivel inferior al umbral de señal para

ruido. Este tipo de detección se diferencia por lo que transmite y por lo que llega a determinar.



Figura 8. Sensores Activos [23]

1.3.1.3 El Espectro Electromagnético

El espectro electromagnético se define como cualquier energía radiante que esté en función de su longitud de onda y de frecuencia. A pesar de que los valores de longitudes de onda son continuos, se establece una serie de bandas en donde la radiación electromagnética denomina un comportamiento aparente [23], figura 9.

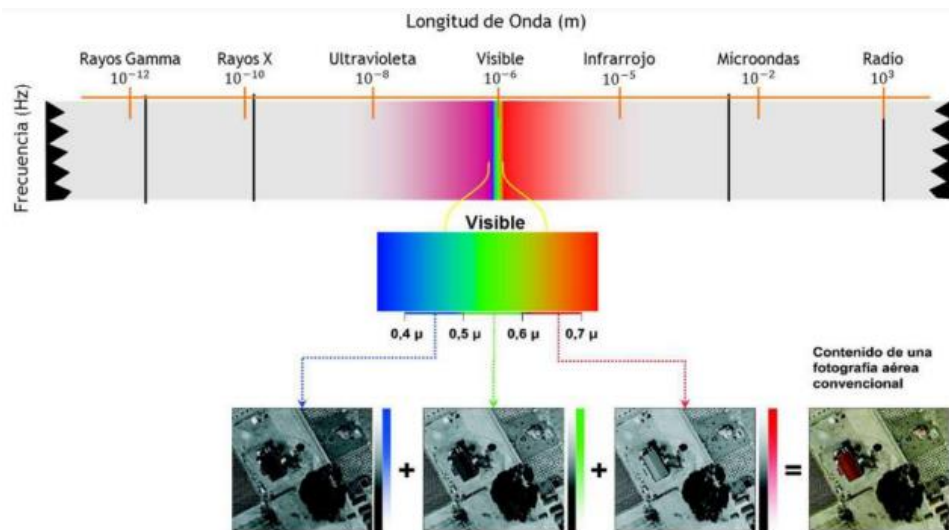


Figura 9. Espectro Electromagnético [22]

Las bandas más empleadas dentro del espectro electromagnético para teledetección son las siguientes [23]:

- **Espectro visible:** (400 nm a 700 nm) Es la única radiación electromagnética que es capaz de ser percibida por nuestros ojos, como la radiación solar. Aquí se logran distinguir 3 bandas primordiales: Azul (400 nm a 500nm), verde (500nm a 600nm) y rojo (600nm a 700nm).

- **Infrarrojo próximo:** (700 nm a 1300 nm) en esta franja se procede a separar las superficies vegetales de la humedad o sequía mediante espectros NDVI.
- **Infrarrojo medio:** (1,3 um a 8 um) en esta franja se mezclan los procesos de reflexión y emisión de luz en la superficie.
- **Infrarrojo lejano o térmico:** (8 um a 14 um) Detección de las emisiones de radiación corporal de un cuerpo debido a su temperatura.
- **Microondas:** (a partir de 1 um) Esta franja es utilizada para medir la emisión terrestre. Existen pequeñas perturbaciones atmosféricas y refleja transparencia con las nubes. Los sensores activos son mayormente utilizados en esta franja.

1.3.1.4 Sistemas Multiespectrales

Para la teledetección es de mucha importancia tener imágenes multiespectrales, debido a que facilita la toma de datos en varias imágenes sobre una misma banda espectral. Las imágenes multiespectrales se las obtiene de diferentes maneras como: el uso de cámaras fotográficas de tipo MK4 o a su vez con un sistema de vidicon presentes en múltiples cámaras de video. Para la captura de una zona específica estas cámaras pueden ser alineadas o colocadas en forma estratégica, evidentemente con los filtros correspondientes que nos proporcione la radiación electromagnética deseada [15], figura 10.

Los registros en los sistemas de imágenes multiespectrales tienen dos principales desventajas [15]

- Cada una de las imágenes se obtienen con un sistema de lentes distintos y esto puede hacer muy difícil el registro de imágenes entre ellas.
- Se encuentra restringido el rango espectral de los sistemas de cámara para la región visible y la región infrarrojo próximo.

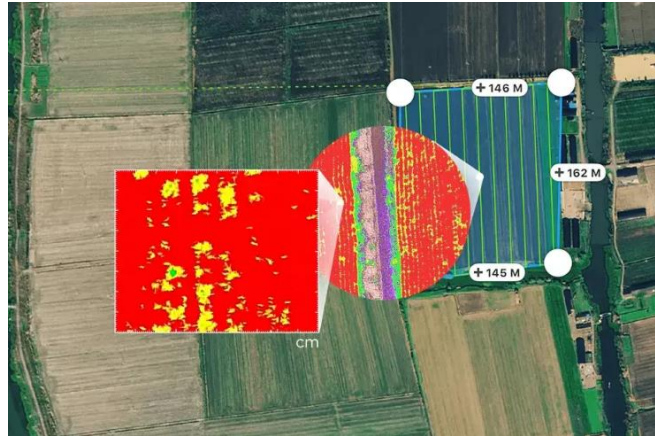


Figura 10. Sistemas Multiespectrales [15]

1.3.1.5 Tipo de imágenes de teledetección

Imagen multiespectral (MS)

Este tipo de imagen lleva consolidado varios valores numéricos en cada pixel, dependiendo de las bandas espectrales que sea capaz de detectar el sensor, figura 11. En un principio, estas imágenes nos proporcionan un tipo de producto más útil debido a la firma espectral de cada uno de los elementos que se encuentran en la imagen. Estas imágenes se captan por medio de un sensor digital, cuya reflectancia es medida en varias bandas [22].

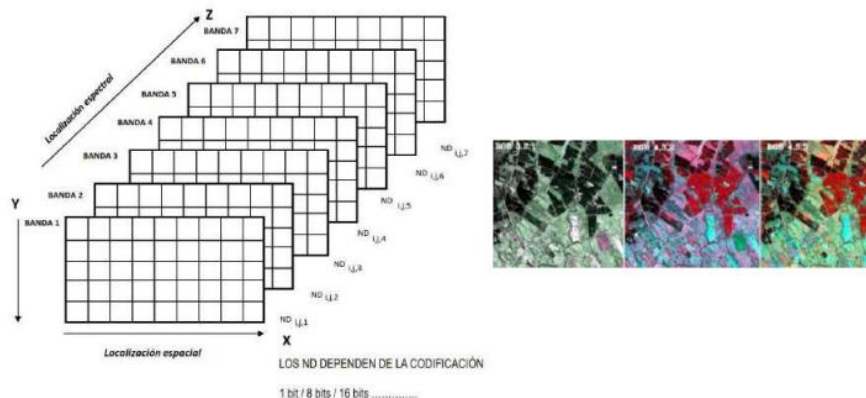


Figura 11. Estructura de imágenes multiespectrales [22]

Imagen pancromática (PAN)

Este tipo de imagen tiene una sola banda espectral, mediante la cual se observa gran parte del visible y principios del infrarrojo, obteniendo como resultado una imagen en escala de grises (imagen blanco y negro), figura 12. Como ventaja de estas imágenes, poseen una mayor resolución espacial a comparación que las imágenes

multiespectrales. Es por esta razón que son demasiado importantes dentro de la detección de elementos pequeños de la superficie terrestre, las cuales no son distinguibles en las imágenes multiespectrales [22].

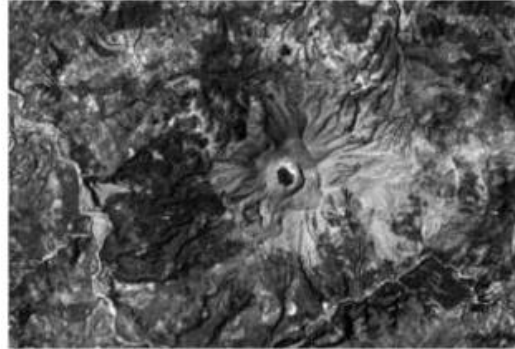


Figura 12. Imagen pancromática B/N [22]

Imagen fusionada (PS)

Esta imagen es la combinación de una imagen multiespectral y una imagen pancromática, figura 13. Consiste en añadir a cada pixel de una imagen pancromática los valores obtenidos de un algoritmo que fusione la imagen pancromática con la multiespectral [22].



Figura 13. Imagen fusionada (PS) [22]

Imagen Estéreo

Este tipo de imágenes hace referencia a dos imágenes tomadas de una misma zona con distintos ángulos de visión, figura 14. Algunos satélites tienen la capacidad redireccionar a los sensores y esto permite la toma en una o simultáneas pasadas [22].

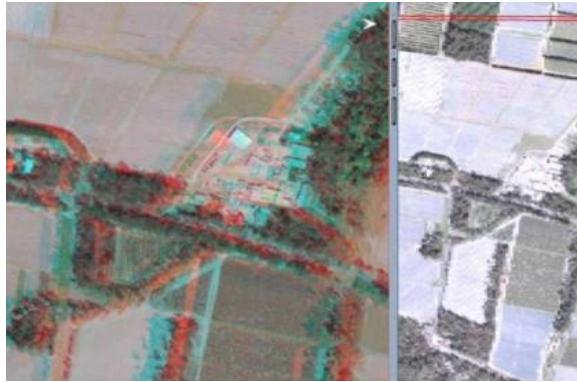


Figura 14. Imagen Estéreo [22]

1.3.2 Índice de Vegetación

La evaluación, control y monitoreo de los recursos naturales son la clave para el análisis de la vegetación y detección de trastornos y deficiencias en los cambios de los patrones de vegetación. Esta es una de las aplicaciones más importantes dentro del ámbito de la teledetección debido a que conlleva a un manejo propicio de los recursos naturales y a su vez la toma de decisiones. Los índices de vegetación reflejan una transformación que involucra la combinación matemática entre el almacenamiento de niveles digitales los cuales están almacenados dentro de la misma imagen en dos o más bandas espectrales [22][25].

Las bandas espectrales que se encuentran entre 0.45 y 0.67 μm son utilizadas para que la clorofila absorba fuertemente energía. Es por esta razón que a los ojos humanos la percepción de vegetación saludable se la caracteriza de color verde, esto debido a la gran impregnación en azul y rojo por las hojas y el verde mediante la reflexión. La clorofila disminuye mientras menos saludable es la vegetación, dándonos como resultado la reflectancia espectral mayormente en el color rojo y como consecuencia de esto las hojas se tornan de un color amarillento (mezcla verde – rojo) [22].

1.3.2.1 Fundamentos de la vegetación

Comportamiento espectral de la hoja

La clorofila de las hojas no absorbe uniformemente toda la luz que proporciona el sol. Las moléculas de la clorofila absorben con mayor preferencia la luz roja y azul, figura 15, puesto que esta es usada en su mayoría por la fotosíntesis. Estas moléculas deben absorber luz incidente en un rango de 70% a 90%. Con respecto a la luz verde,

esta es menos absorbida y mucho más reflejada (figura 16), por ende, el ser humano observa solo el espectro visible (luz verde), como la vegetación sana [22].

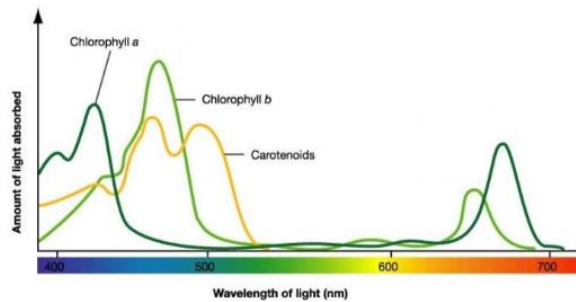


Figura 15. Espectro de absorción para la clorofila [22]

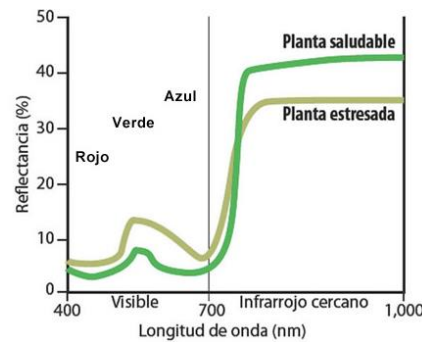


Figura 16. Reflectancia espectral de una hoja viva [22]

El espectro ultrarrojo cercano es de gran utilidad para el estudio de la vegetación, gracias a que ayuda a la distinción de las superficies terrenas con vegetación de las sin vegetación, dado que en el ultrarrojo cercano se las visualiza muy oscuras. Los cambios de vigor de vegetación y las imágenes captadas en el espectro infrarrojo son muy valiosa para el campo de la teledetección, presencia y distinción de enfermedades en los cultivos. Además, los cambios de madurez que se presentan en las hojas de los cultivos vienen sujetas a este tipo de espectro infrarrojo. Las bandas espectrales rojas que se encuentran entre 0.6 a 0.7 μm y el infrarrojo cercano de 0.7 a 1.1 μm , muestra un comportamiento característico de vegetación sana. De esta manera logrando separar la vegetación saludable de otras cubiertas por dicho contraste espectral [22], figura 17.

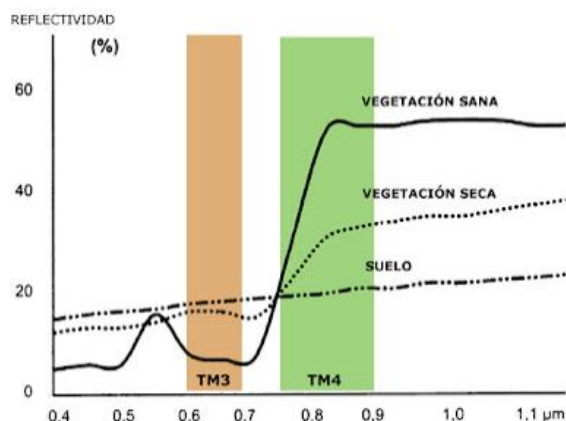


Figura 17. Contraste espectral de vegetación sana contra vegetación seca [22]

1.3.2.2 Tipos de índices

Existen algunos tipos de índices de vegetación, entre los cuales tenemos [22]:

- **Índice de vegetación proporcionada – RATIO:** Se representa la relación que tiene las reflectividades del infrarrojo cercano con el rojo, obteniendo un cociente simple (Ci).
- **Índice Vegetación Diferencial – DVI:** Se obtiene la diferencia que existe entre las bandas infrarrojo cercana y rojo de un sensor.
- **Índice de Vegetación de Diferencia Normalizada – NDVI:** Se lo utiliza para obtener la diferencia normalizada que existe entre las reflectancias del ojo y del infrarrojo cercano, brindando información sobre el estado de la cantidad, calidad y desarrollo de la vegetación en superficies terrestres extensas.
- **Índice de vegetación mejorada – EVI:** Surgió a partir del NDVI, dando un aumento a la sensibilidad de alta biomasa y una mejora considerable en el monitoreo de la vegetación.
- **Índice de vegetación Transformado – TVI:** Comparte los valores de NDVI, solo que se agrega una constante de 0.5 para evitar valores negativos.
- **Índice de Vegetación Atmosféricamente Resistente – ARVI:** Es un acondicionamiento del NDVI con respecto a las condiciones atmosféricas sobresaltando la dispersión de los canales rojo y azul del espectro.

- **Índice de Vegetación Avanzada – AVI:** Tiene cierta similitud al NDVI, proporcionándonos un indicador numérico y utiliza las bandas espectrales IRC y rojo.
- **Índice de vegetación Ajustado al Suelo – SAVI:** Se utiliza este índice en áreas semiáridas, en donde el suelo cumple un papel muy importante con su contribución.
- **Índice de Área foliar – LAI:** Este índice consiste en un indicador de biomasa con respecto a cada pixel de una imagen.
- **Índice de Suelo desnudo – BSI:** Consiste en un indicador numérico que requiere de las bandas espectrales rojo, azul e IRC.

1.3.2.3 Índice de Vegetación de Diferencia Normalizada (NDVI)

El Índice de Vegetación de Diferencia Normalizada se utiliza en su mayoría para el tratamiento de imágenes de la biomasa vegetal, mediante lo cual se logra la distinción de la vegetación sana de otras superficies o coberturas. Las bandas espectrales que se utilizan para este tipo de índice son del rojo (0.6-0.7 um) e IRC (0.7-1.3 um). Cada uno de los valores nos representa el estado de la vegetación, mientras más alto el valor del índice la vegetación es vigorosa, esto debido a la mayor reflectividad emitida por el IRC y la baja reflectividad por el rojo visible. Las nubes y el agua arrojan valores negativos en el índice debido a una mayor reflectancia en el infrarrojo visible. Por otro lado, se tiene los valores que son cercanos a cero, estos representan un suelo vacío sin vegetación o a su vez zonas rocosas, puesto que las reflectancias entre el IRC y el rojo visible son iguales [21][25].

Fórmula para el cálculo de NDVI

La fórmula diferencial para calcular el índice de vegetación normalizada sirve para identificar la cantidad de crecimiento de cada planta sobre la tierra, esta consiste en el IRC menos el rojo o espectro visible dividido entre el IRC más el rojo o espectro visible.

Matemáticamente, se representa como la siguiente formula:

$$NDVI = \frac{NIR-VIS}{NIR+VIS}$$

Para los cálculos de NDVI los resultados de cada uno de los pixeles varían en el rango de -1 a 1, como se presenta en la figura 18. No obstante, llegar a obtener un valor cercano a 0 para vegetación verde es imposible debido a que este valor identifica un suelo desnudo y por otro lado obtener valores cercanos a uno (0.8-0.9) nos indica la presencia de hojas verdes en la más alta densidad [25], como se indica en la figura 19.

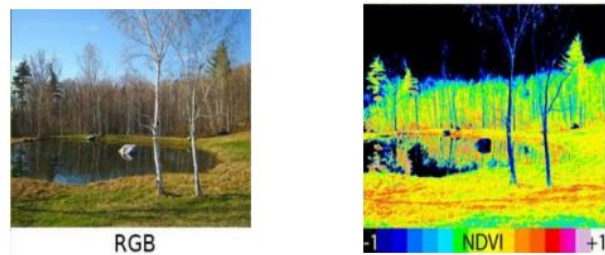


Figura 18. Cálculo de NDVI [25]

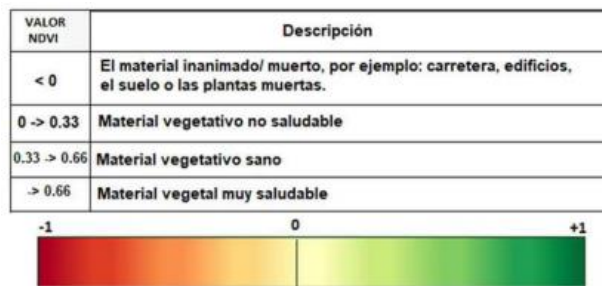


Figura 19. Rangos de vegetación en NDVI [22]

El índice de vegetación normalizado no emplea longitudes de ondas propias, debido a que este sistema es implementado según los requisitos requeridos [25], se puede observar un ejemplo en la figura 20.

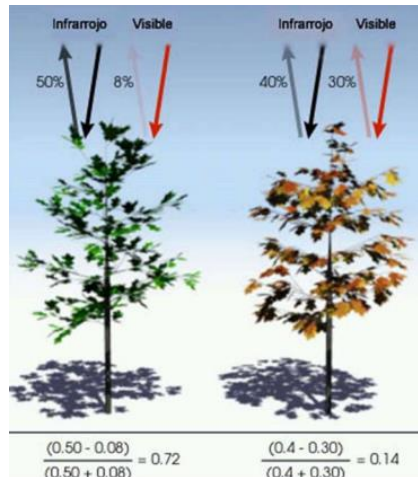


Figura 20. Ejemplo de NDVI [22]

Aplicación del Índice de Vegetación Normalizado

El índice NDVI nos ofrecen información cuantitativa y cualitativa para su análisis. Para un análisis rápido se emplea el tratamiento de las imágenes de manera cualitativa como se observa en la figura 21 el estado respectivo de una superficie vegetal. Mediante este análisis se puede determinar de manera directa los factores que afectan a la producción de la biomasa presentes en las zonas vegetales. De esta manera evaluando los efectos sobre la vegetación por esquías, incendios, agentes patógenos, enfermedades, granizo, heladas o carencias nutricionales [25].

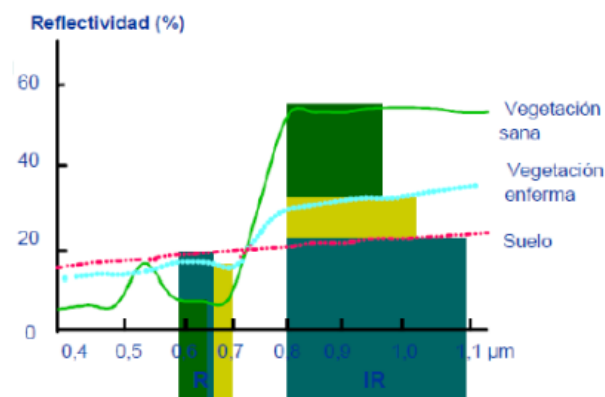


Figura 21. NDVI a causa de la reflectancia [25]

1.3.3 Vehículos Aéreos no Tripulados (UAV)

En la actualidad existen un sin número de sistemas de teledetección aéreo que en su mayor parte lo representan los satélites, cuyo uso es primordial para el mapeo de las

características de la biomasa sobre la superficie de la Tierra. A pesar de todos los usos que se le puede dar a esta plataforma posee desventajas tales como un alto costo de inversión tanto en su infraestructura, como en equipamiento, permisos de navegación, entre otros.

Los sistemas aéreos no tripulados (Unmanned Aerial Vehicle, UAV) han tenido un gran auge hasta el momento, tiene grandes ventajas como el transporte de sensores para la teledetección de la superficie terrestre. Al igual que los vehículos aéreos tripulados, los UAV nos ofrecen grandes ventajas y características importantes depende su lugar de uso.

Los drones son el grupo mayormente representativo dentro de los sistemas aéreos no tripulados, los cuales poseen un gran número de aplicaciones dentro de diferentes campos como, por ejemplo: agricultura, vigilancia, mapeo entre otros [26].

1.3.3.1 Drones

Los drones están definidos por la Real Academia Española como “Aeronave no tripulada “por ende, todo vehículo aéreo que pueda navegar y correr en el aire sin la necesidad de una tripulación será considerado como un dron. Por otro lado, los drones pueden albergar un sistema de autopiloto, que los convierte en autónomos. Son conocidos por sus siglas UAV (figura 22), aunque también se los conoce por sus siglasUCAV (Unmanned Combat Aerial System), los cuales hacen referencia estrictamente a no ser tripulados [27].



Figura 22. Dron (UAV) [27]

Historia de los Vehículos Aéreos no Tripulados

Tabla 2. Historia de los drones (UAV)

Historia de los Vehículos Aéreos no Tripulados

Años	Conceptualización
1918 – Charles Kettering	En sus inicios se impulsó al desarrollo de una máquina de vuelo, la cual era controlada mediante un giroscopio y tenía un ciclo de uso muy corto debido a que los números de giro en la hélice eran limitados.
1940 – Auge de la Segunda Guerra mundial	Apareció el uso de misiles balísticos, por ende, los drones fueron utilizados como armas de tiro por la Fuerza Aérea y la Marina.
1950	Apareció el control aéreo de drones pilotados a distancia.
1960	Se realizó una recategorización de los drones, separando lo nuevo de lo ambiguo y obsoleto.
Ultima década	En los últimos años los vehículos aéreos no tripulados (UAV) han tenido un gran auge comercial, debido a sus beneficios en el momento de su uso, principalmente en el ámbito de la fotografía. Teniendo como desventaja las restricciones de las leyes de aviación civil.

Fuente: [28]

1.3.3.2 Tipos de drones

Los drones se clasifican en dos tipos según su estructura, los cuales son:

Drones de ala fija

Estas aeronaves están compuestas por un par de alas fijas y un fuselaje, como se muestra en la figura 23. Muy parecidos con el diseño de los aviones convencionales y su uso es predominante en las fuerzas militares. Poseen una gran capacidad para volar, recorren largas distancias debido a que son capaces de llegar alcanzar velocidades muy grandes. Este tipo de UAVs tiene un sistema de propulsión compuesto por un motor de turbinas o combustión [29].



Figura 23. Dron profesional de ala fija [29]

Drones Multirotor o Multicóptero

Este tipo de drones son considerados como un helicóptero el cual está formado por dos o más motores o rotores. Estos motores están sujetos a una serie de brazos, donde además se alojan todos los componentes electrónicos y la circuitería necesaria para su funcionamiento. La forma de vuelo que adopta este tipo de drones es con giros traspuestos y sincrónicos de las hélices del UAV. Dos de los motores hacen girar las hélices en sentido horario mientras que los dos restantes en sentido contrario, de esta manera ejerciendo la fuerza necesaria para el despegue o alza de la aeronave. Estos drones logran mantenerse estables y estáticos gracias a los estabilizadores o giroscopios que forman parte de sus componentes electrónicos [29].



Figura 24. Dron multirotor [29]

Tabla 3. Comparativa entre drones de ala fija y multirotor

	Drones de ala fija	Drones multirotor
Energía	Explosión/Eléctrica	Eléctrica
Carga Útil	Menor	Mayor
Vuelo	Velocidad y altitud	Estable y estacionario

Autonomía	Mayor	Menor
Maniobrabilidad	Menor	Mayor

Fuente: [29]

Entre los drones multirrotor se tiene su clasificación por el número de motores presentes en su estructura, así como:

- **Tricóptero**

Estos drones están estructurados por tres motores que le permiten su vuelo en el aire, figura 25. Es la manera más simplificada de las estructuras de los drones multirrotor. La maniobrabilidad de estos UAV en el aire depende de las revoluciones de los motores ya sea que estos aumenten o disminuyan. Poseen un cuarto punto de giro que se encuentra dentro del fuselaje, guiando al servomotor en su control del módulo de giro, haciendo posible su equilibrio de suspensión en el aire [30].



Figura 25. Dron Tricóptero [30]

Metodología de Vuelo

Para el manejo de un Tricóptero es necesario controlar el número de revoluciones. Según las normas de vuelo, este multirrotor se logra al girar dos de estos motores en sentido contrario al tercer motor, figura 26. En los laterales se ubican dos motores que ayudan al tricóptero a oscilar en forma horizontal. Mientras que el tercer motor se ubica en la cola, el cual es el encargado de dar potencia de empuje para avanzar o retroceder, además con este servomotor se lograr la rotación del dron en su propio eje [27].

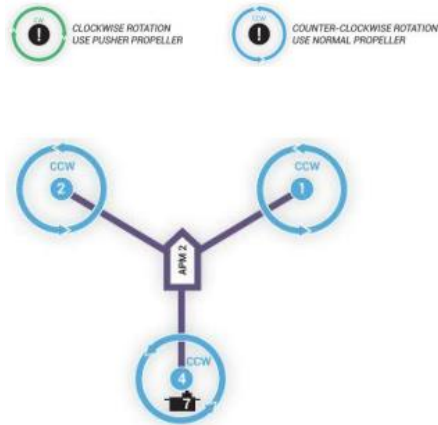


Figura 26. Configuración de motores en tricóptero [27]

Tabla 4. Ventajas y desventajas de los tricóptero

Ventajas	Desventajas
Gran maniobrabilidad por su tercer motor.	En caso de avería, descenderá fuera de control.
Transporte y almacenaje cómodo.	Baja estabilidad y potencia.
Peso ligero	Soporte de carga limita.
Económico	Sin carcasa de protección para el fuselaje.
Gran número de movimientos.	Los motores adquieren grandes revoluciones para ejecutar su vuelo.

Fuente: [27]

- **Cuadricóptero**

Estos drones están formados por cuatro motores en conjunto con sus hélices, los cuales proporcionan la fuerza de vuelo a la aeronave (figura 27). Tienen una configuración común para sus multirotores [30].



Figura 27. Dron cuadricóptero [30]

Metodología de Vuelo

Para generar el vuelo de estos drones, dos de sus motores giran en sentido del reloj mientras que los restantes en sentido antihorario, de esta manera neutralizando la fuerza que se genera y llegar a un punto de equilibrio. Es obligatorio que la suma de las fuerzas sea igual a las que se generan cuando el dron está en movimiento, si esto no se cumple el dron perderá el equilibrio y caerá fuera de control [27].

La configuración en forma de cruz (+), figura 28, logra un fácil manejo de forma vertical y horizontal [27].

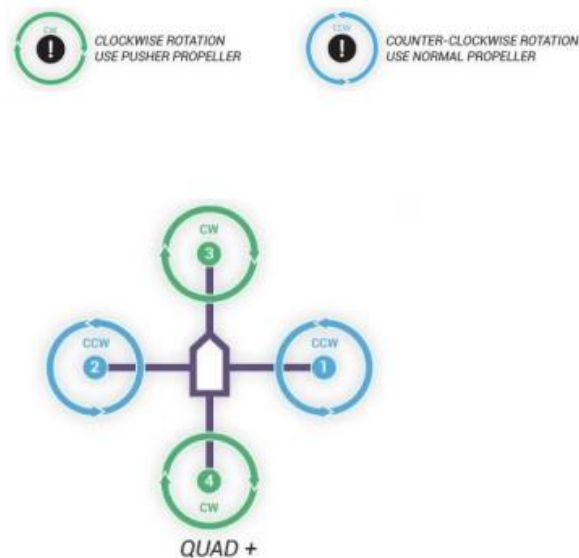


Figura 28. Configuración de motores en cuadricóptero [27]

La configuración en forma de X o H tienen una separación de 45° entre todos sus brazos (figura 29), esta estructura es la más

recomendable si en el dron estará fijada una cámara, debido a que se tiene un punto de visión amplio sin interrupciones [27].



Figura 29. Configuración de motores en cuadricóptero, forma X [27]

Tabla 5. Ventajas y desventajas de los cuadricópteros

Ventajas	Desventajas
Aceleración rápida	Limite en su estabilidad y velocidad
Estructura simple y robusta	Transporte y almacenaje complicado
Capacidad de carga mayor	Rígido
Muy ágil	En caso de daño, no tiene redundancia
Amplia variedad en el mercado	Carga limitada

Fuente: [27]

- **Hexacóptero**

Este dron consta de 6 motores en conjunto con sus hélices, es mucho más funcional y trabaja de mejor manera debido a su alta potencia y estabilidad, figura 30. La falla de uno de sus motores no perjudica el vuelo y se puede ejecutar un aterrizaje apropiado [30].



Figura 30. Dron Hexacóptero [30]

Metodología de Vuelo

Su manera de vuelo se basa los drones cuadricópteros. Todos los motores se encuentran posicionados al mismo nivel, tres motores giran con sentido horario y otros tres con sentido antihorario. Se presentan configuraciones de vuelo en forma de + y x [27], figura 31.

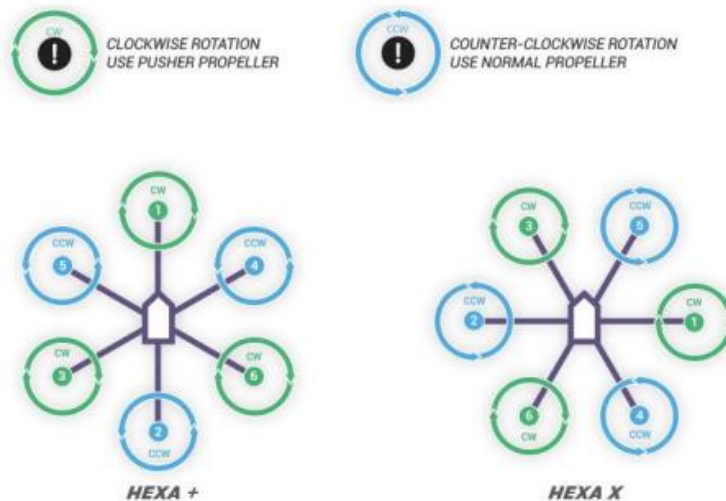


Figura 31. Configuración de motores en cuadricóptero [27]

Tabla 6. Ventajas y desventajas de los hexacópteros

Ventajas	Desventajas
Mayor estabilidad en el vuelo.	Costos muy altos.
Gran nivel de carga.	Difícil transporte del equipo.
Instalación de accesorios con gran facilidad.	Mayor consumo de batería.
Redundancia al existir una avería	Conocimientos de nivel avanzado

en uno de los motores.	para su manipulación.
------------------------	-----------------------

Fuente: [27]

- **Octocóptero**

Estos drones son de un nivel muy superior, debido a su potencia y a su costo. Están formados por 8 motores en conjunto de sus hélices, figura 32. Tras la falencia de uno de los motores no se pierde el control y se efectúa un vuelo estable. Su uso es profesional y posee la capacidad de llevar carga muy pesada [30].



Figura 32. Dron octocóptero [30]

Metodología de Vuelo

De igual manera que los hexacópteros, este tipo de drones basa su configuración en los cuadricópteros. Cuatro de sus motores giran en sentido de las manecillas del reloj mientras que los cuatro restantes en sentido opuesto, de esta manera estabilizando e igualando la fuerza de vuelo. Se tiene configuraciones en + y x [27], figura 33.

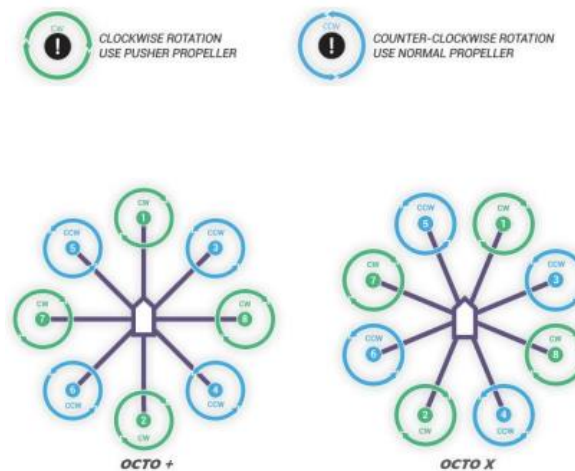


Figura 33.. Configuración de motores en cuadricóptero [27]

1.3.3.3 Características de los drones

Los drones actualmente poseen un gran número de ventajas y características, entre las cuales están:

Velocidad

La velocidad promedio de los drones es de 20m/s, esto puede variar debido a su estructura, componentes y demás accesorios que se ubique al equipo [30].

Altitud

La altitud que alcanzan los drones varía según su estructura, tamaño y peso. Es necesario que los drones sean ensamblados con materiales resistentes, fuselajes, motores y hélices [30].

Nivel de autonomía

El nivel de autonomía depende directamente de la batería que se utilice y para que fines, debido a que el tiempo promedio de vuelo para un dron es de 25 minutos dependiendo su calidad. Para obtener una excelente autonomía es necesario que la batería sea pesada, así se tiene un voltaje apto para una buena potencia en los motores brindando mayor aceleración y rapidez [30].

Dimensiones y peso

Esta característica varía dependiendo de cada tipo de dron, ya sea por su estructura, elementos y accesorios. El peso promedio de un dron cuadricóptero es de 320 a 450 gramos [30].

1.3.3.4 Componentes de los drones

Sistema de propulsión

Este sistema es el encargado del movimiento de la aeronave (figura 34), el cual está integrado por: el motor, las hélices, el variador, la batería. Estos elementos son muy importantes para suspender al dron en el aire, dependiendo del tamaño y la potencia del UAV [30].



Figura 34. Sistema de propulsión [30]

Motores

Los brushless (sin escobillas) son quienes impulsan al motor, están compuestos por dos partes una móvil conocido como rotor y otra fija llamada estator, figura 35. El factor importante a considerar dentro de este componente es el “kV”, que nos indica las revoluciones por minuto que se efectúan por cada voltio de electricidad [27].



Figura 35. Motor [27]

Variadores

Conocido como ESC (Electronic Speed Controller), este componente es quien controla la velocidad de los giros del motor, además nos ayuda a controlar la velocidad y se usa como freno dinámico, figura 36. Se conecta en conjunto con el controlador de vuelo y los motores, aunque la sincronización entre ambos debe ser precisa para que los motores den un giro correcto [27].



Figura 36. Variador [27]

Hélices

Son elementos muy importantes para el vuelo de un dron, debido que de esto depende la fuerza con la que el UAV transite. Las hélices con mayor grado de inclinación y tamaño provocan un empuje mayor brindando la capacidad de soportar carga pesada [27], figura 37.

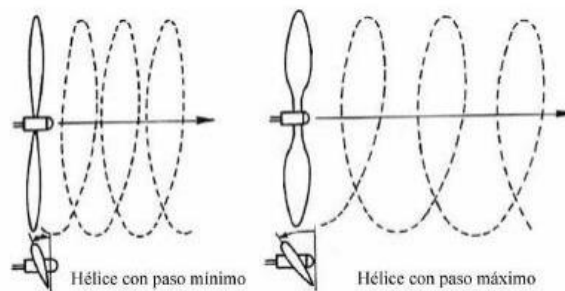


Figura 37 Hélices [27]

Batería

Las baterías que están formadas por iones de litio son conocidas como LIPO o Li-Ion, estas se utilizan para la alimentación de los drones debido a que poseen un alto nivel de densidad energética. Estas baterías se componen por un número estimado de celdas, las cuales poseen una carga proporcional a la duración de la misma. Recalcar que a mayor peso de un dron mayor es el consumo de la batería [27].

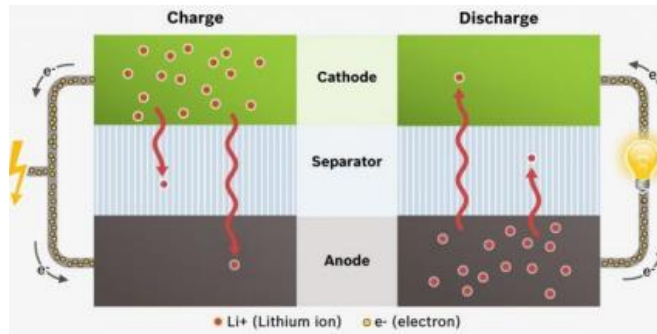


Figura 38. Estructura de LIPO [27]

Controlador de vuelo

Es considerado como el cerebro del UAV, su función principal es accionar el giro de los motores de forma correcta para el manejo del piloto o su vez seguir una ruta en vuelo automático. El controlador es además el encargado de procesar información que reciben los sensores, figura 39. En la actualidad los controladores vienen adjuntos a una Unidad de Medición Inercial conocida como IMU, convirtiendo al UAV en una nave inteligente [27].

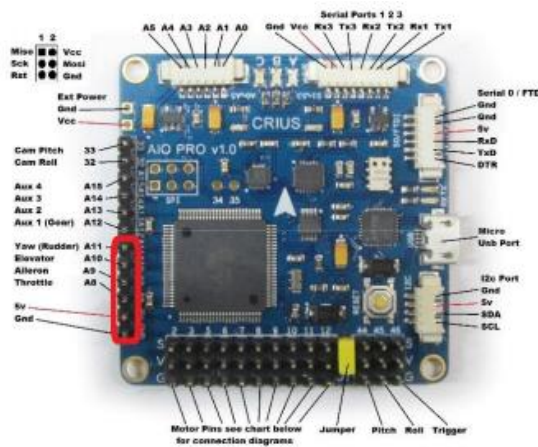


Figura 39. Controlador de vuelo [27]

1.3.3.5 Vehículos aéreos no tripulados en el campo de la agricultura

Los vehículos aéreos no tripulados tienen muchas aplicaciones dentro de cualquier campo, debido a que su uso se ha vuelto muy versátil gracias a la facilidad de adquisición de imágenes remotas, sobre todo en un campo muy extenso como la agricultura [25].

Los UAV son utilizados en el agro para la evolución de la producción, tomando en cuenta tres escenarios distintos que pueden afectar gravemente a los cultivos:

- Detección de superficies infestadas por malas hierbas en cultivos agrícolas.
- Detección de áreas en donde es necesario un control de riego, ya sea menos o mayor para los cultivos.
- Detección de cultivos infectados por agentes patógenos, hongos o enfermedades que disminuyan la producción.

El uso de drones para la agricultura se vuelve más común con el pasar del tiempo, gracias a los buenos resultados y eficiencia que estos han proporcionado a los agricultores y a los investigadores. Estos sistemas de detección con el uso de drones ayudan a los agricultores a desarrollar una agricultura tecnificada [25].

1.3.3.6 Agricultura de precisión con drones

El tema de agricultura de precisión o tecnificada con el uso de drones es de mucho interés en la actualidad, debido al monitoreo de amplias superficies del campo. De esta manera los drones al tener sensores y accesorios como cámaras térmicas brindan un gran desarrollo en tecnología y servicio al campo agrícola, figura 40. El uso de UAVs dentro de la agricultura permite capturas imágenes térmicas, mapeo de volumen, fotogramétricos, mediante los cuales se logra la recolección de grandes sumas de información, necesaria para conocer la condición de cada uno de los cultivos, por el tratamiento de imágenes [31].



Figura 40. Formas de extracción de información, plataformas y sensores útiles en los UAVs [31]

La observación de las capturas de imágenes aéreas fuera del rango del espectro visible es decir lo que llega a percibir el ojo humano, se lo puede realizar mediante el uso de las regiones del infrarrojo térmico y reflejado, figura 41, con el uso de cámaras predestinadas a esta función como lo son las multispectrales, mediante lo cual los datos obtenidos generan más confianza y brinda una caracterización mejorada a lo sucedido en el campo de cultivo mediante los índices de vegetación [25].

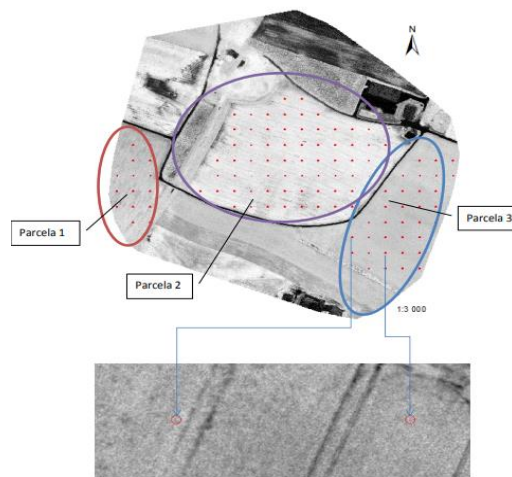


Figura 41. Agricultura de precisión mediante el uso de teledetección [25]

1.3.4 Visión Artificial

La inteligencia artificial consta de muchas ramas dentro de su campo. Una de ellas es la de visión artificial, que ejecuta tras un código de programación en un interfaz de computadora, sobresaltando los rasgos más relevantes dentro del objeto posicionado en la imagen. Tiene un sin número de aplicaciones en cada campo, como en la agricultura para el control y verificación de los estados de salud de la vegetación. Para obtener resultados satisfactorios los sistemas de visión artificial constan de algunos factores [25], observado en la figura 42.

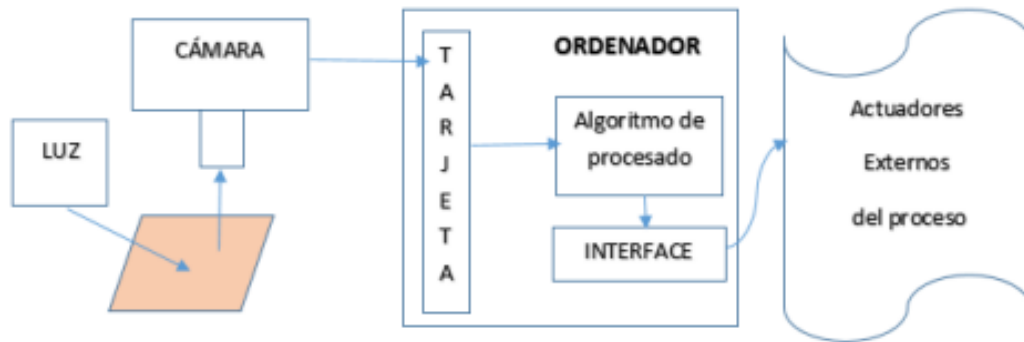


Figura 42. Estructura de un sistema de visión artificial [25]

Los procesos que constan de un sistema de visión artificial son [32]:

- Captar una imagen
- Conversión en formato digital
- Procesado en un monitor u ordenador
- Obtención de resultados

1.3.4.1 Componentes de un Sistema de Visión Artificial

Los sistemas de visión artificial constan de elementos principales tales como, la iluminación, sensor de imagen, procesamiento de visión y comunicación [32], figura 43.

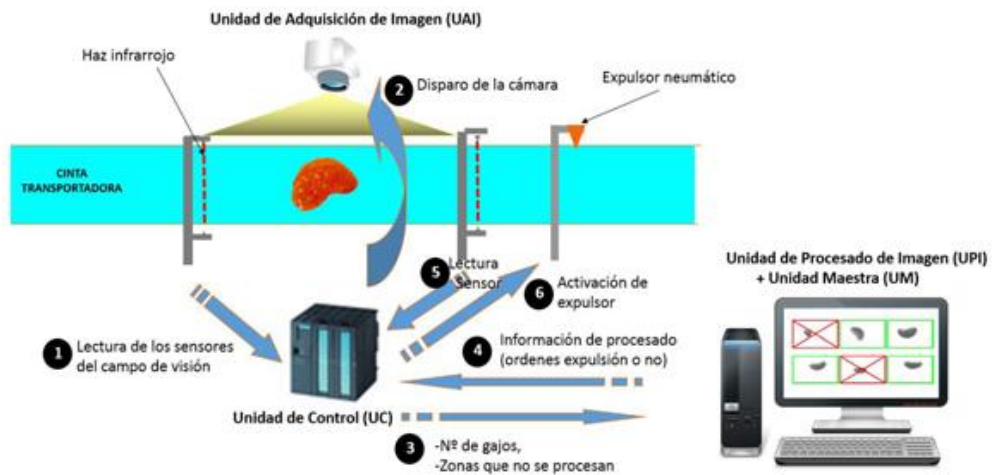


Figura 43. Componentes del Sistema de Visión Artificial [32]

Iluminación

Es uno de los parámetros más importantes dentro de la visión artificial, dado que, mediante un aceptable nivel de luz visible, las capturas de las imágenes pueden ser claras y nítidas para un tratamiento propicio de las mismas [32].

Lentes

Los lentes son quienes definen la calidad de la imagen y una resolución nítida. Las cámaras están compuestas por dos tipos de lentes, uno en montaje C que pueden ser intercambiables y los fijos para un enfoque automático [32].

Sensor de la imagen

Este componente es muy importante dentro de la captura de imágenes, debido a que es quien permite obtener una fotografía con una iluminación correcta para la inspección del objeto. Mientras la resolución de la cámara sea mayor, los detalles a considerar dentro del tratamiento de la imagen son mayores [32].

Procesamiento de Visión

Aquí se extrae la información captada de una imagen digital, mediante el uso de softwares. Obteniendo como resultado las características propicias, mediciones y comparación para la toma de decisiones [32].

Actuadores

Los sistemas de visión artificial son grandes procesos, brindándonos una amplia cantidad de resultados mediante robots, imágenes espectrales y cualquier tipo de actuador [25].

1.3.4.2 Etapas de un sistema de visión artificial

Un sistema de visión artificial consta de las siguientes etapas, que están relacionadas en hardware y software [32], como se muestra en la figura 44:

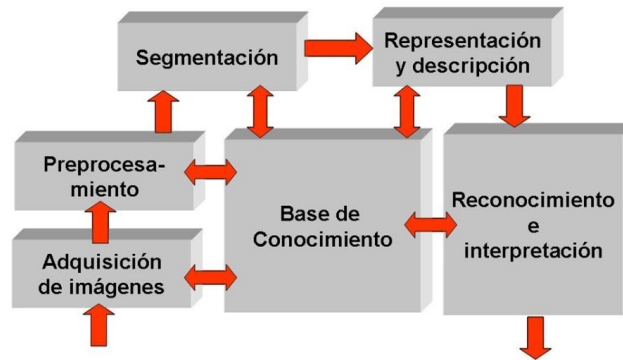


Figura 44. Etapas de un Sistema de Visión Artificial [32]

Las etapas de un sistema de visión artificial son [32]:

1. **Adquisición de imagen digital.** - La imagen capturada debe ser nítida y con una excelente claridad, para que el tratamiento de la imagen sea satisfactorio.
2. **Imagen digitalizada.** - Una vez obtenida la imagen, esta se dirige a una etapa conocida como preprocesamiento en donde su objetivo es el de mejorar la imagen.
3. **Segmentación.** - Es la parte más importante del proceso, debido a la segmentación de la imagen por fragmentos o partes para el reconocimiento de objetos.
4. **Salida del proceso de segmentación.** – En esta parte del proceso se contienen los puntos (propiedades internas de la imagen) y las fronteras (propiedades externas de la imagen) de la región detectada.
5. **Parametrización.** – Se obtiene la información cuantitativa y los rasgos básicos para diferenciar los objetos de una imagen.
6. **Reconocimiento e interpretación.** – Se realiza el etiquetado de los objetos reconocidos en una imagen para clasificarlos.

1.3.4.3 Tipos de Aprendizaje

Los tipos de aprendizaje en un sistema de visión artificial son los siguientes [33]:

- **Redes híbridas.** – Mejora la convergencia brindando un enfoque mixto, ejemplo de esto son las redes de base radial.
- **Aprendizaje reforzado.** – Esta entre el aprendizaje autoorganizado y el aprendizaje supervisado.
- **Aprendizaje supervisado.** – Este aprendizaje necesita de la entrada de datos, los cuales se encuentren clasificados para el reconocimiento de su objetivo.

- **Aprendizaje no supervisado.** – No es necesario ningún conjunto de datos de entrada, dado que su aprendizaje es automático.

1.3.4.4 Red Convolutiva YOLO

YOLO se define como una red de análisis convolutiva la cual predice de manera sincronizada un sin número de delimitación por cuadro. La función de YOLO es dividir la imagen procesada en una matriz SxS. Si el objeto se encuentra en el centro de un cuadro de la matriz, este es el encargado de detectar el objeto. Esta red convolutiva se entrena mediante el uso de imágenes para brindar un óptimo rendimiento en el reconocimiento de objeto. Adicionalmente YOLO puede efectuar procesamiento de video con una latencia de 25 milisegundos en tiempo real [34].

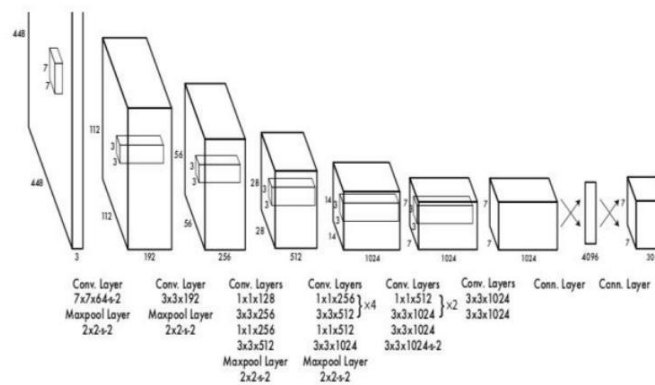


Figura 45. Estructura de una red convolutiva YOLO [34]

Yolov5

Este algoritmo consiste en una red neuronal end to end, debido a que ayuda a clasificarla las imágenes a la clase que pertenece y además envuelve cada objeto detectado en cajas o frames. Esta herramienta se utiliza en las librerías de código de programación Python debido a su adaptación con Pytorch, mediante esta librería es posible instalar dispositivos IoT, mediante un modelo de programación más estructurado [35].

YOLOv5 está formado por una capa de entrada de imágenes, Backbone (extender las características de la imagen), Neck (Encuentro adecuado de píxeles), Head (resultados de la imagen) [32], como se muestra en la figura 46.

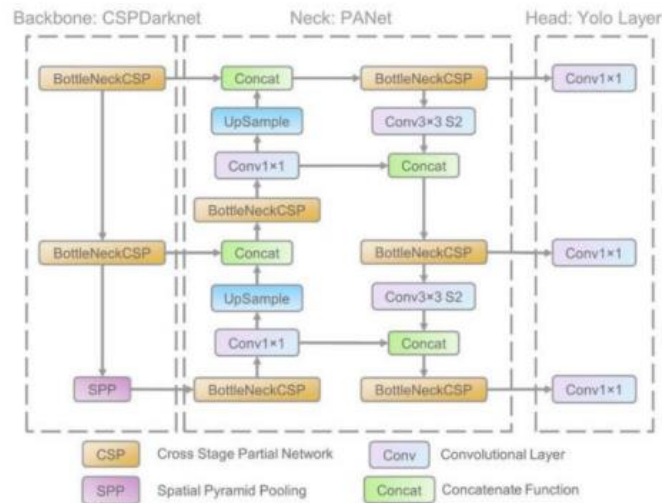


Figura 46. Estructura YOLOv5 [32]

El entrenamiento de esta herramienta se realiza con el uso de redes neuronales, esto quiere decir que se necesita varios ejemplos o capturas de imágenes para que YOLOv5 vaya alimentando sus neuronas mediante un proceso iterativo, es necesario configurar parámetros tales como batch, épocas y peso inicial, dado que dependerá mucho de los resultados y precisión del modelo obtenido al final [32].

1.3.4.5 Makesense.ai

Esta herramienta online se utiliza para etiquetar imágenes, no requiere de instalación en el ordenador debido a que se utiliza directamente en el navegador por esta razón su uso es gratuito. Es muy recomendado para trabajos de visión artificial, gracias a su amigable plataforma y fácil preparación del conjunto de datos. Trabaja con formatos como VOC, JSON, YOLO, XML, CSV, VGG [32].

1.3.4.6 Teachable Machine

Es una herramienta de uso online que facilita la programación de modelos de aprendizaje automático. Teachable Machine entrena a un computador u ordenador para reconocer posturas, sonidos e imágenes. Es netamente relacionado con el interfaz de programación Python dado que esta herramienta utiliza la extensión .js en sus archivos.

1.3.4.7 Herramientas para el procesamiento de imágenes

Existen muchas herramientas para el tratamiento de imágenes digitales, las cuales se pueden utilizar mediante programas destinados a la programación de interfaces, de

acuerdo con las necesidades del creador mediante el uso de librerías, lenguajes y módulos.

Python. – Este es un lenguaje de programación abierto, que tiene la facilidad de trabajar con gran eficiencia en varios sistemas y con semántica dinámica de un nivel muy alto ayudando al óptimo desarrollo de aplicaciones. Es compatible para Windows y Linux, aunque es limitado por sus versiones anteriores y funcionalidad. Este interfaz de programación posee un lenguaje dinámico debido a que no existe la necesidad de identificar la iniciación de variables en los tipos de datos. Es un interfaz muy amigable con el desarrollador, permite optimizar el tiempo al programar de forma correcta [32].

Pythorch. – Esta librería brinda sustento al aprendizaje profundo de las redes neuronales Python, realizando entrenamientos de la inteligencia de forma práctica y rápida. Se diseño para ejecutar algoritmos de redes neuronales en un corto tiempo y tiene la facilidad practica en la depuración de imágenes para Python [32].

1.3.4.8 Bases de Datos

Alberga grandes cantidades de información dirigidos a mismo contexto, de manera estructurada y organizada, facilitando así el uso de estos datos. Existen programas para la creación de bases de datos, tales como:

MYSQL. – Es un gestor de base de datos, muy conocido por su uso gratuito y debido a que sus características y seguridad van mejorando. El manejo de datos es optimizado gracias a que no permite el desperdicio de recursos en el servidor. MYSQL esta desarrollado para quienes necesitan un gran sistema de almacenamiento y a su vez limitado [32].

PHPMYADMIN. – Esta herramienta funciona conjuntamente con las bases de datos de MYSQL, ofreciéndonos el interfaz grafico de la misma para editar y manipular la información en todos sus campos: insertar y eliminar datos, agregar usuarios, conceder los permisos, copias de seguridad, modificar tablas, etc [32].

1.3.5 Brócoli

El brócoli es considerado una hortaliza de climas fríos y templados, su parte comestible es la cabeza o pella que se encuentra en la parte superior de la planta. En el Ecuador la especie de brócoli predominante en los campos y en las zonas de

exportación es de la marca Legacy como se observa en la figura 47, debido a su amplia productividad durante todo el año y su alto nivel de calidad en cada una de sus etapas de crecimiento [36].



Figura 47. Brócoli Legacy [36]

1.3.5.1 Características para el cultivo

Entre las características necesarias para este cultivo se tiene:

- a) **Requerimiento climático:** La temperatura promedio para el cultivo de esta hortaliza es de 15 a 25°C [36].
- b) **Suelo y PH:** Esta hortaliza se cultiva de mejor manera en suelos arenosos u arcillosos con un nivel pH de 6 a 6.8, debido a que el brócoli es muy delicado al contacto con suelos ácidos [36].
- c) **Establecimiento del cultivo:** El cultivo de esta hortaliza se lo realiza mediante un proceso que se lo conoce como trasplante. Las semillas germinan luego de 30 días y las plántulas que tienen de cuatro o cinco hojas están listas para este proceso de trasplante. [36]
- d) **Marco de Plantación:** Para la plantación de esta hortaliza se considera una distancia promedio de 30 a 35 centímetros, realizando la siembra en forma de zigzag para una hilera. En una hectárea es posible que se lleguen a plantar alrededor de 55 mil plántulas [36].
- e) **Cosecha:** La cosecha de este cultivo se la realiza una vez cumplidas 14 semanas o 16 semanas dependiendo del clima o el suelo. La cabeza principal conocida como pella llega a un diámetro de 15 a 18 centímetros y un peso de 450 gramos al momento de su cosecha, estos valores pueden variar por los factores externos a los que está expuesto el producto [36].

1.3.5.2 Fases de Cultivo

Las fases de este producto agrícola con respecto a su desarrollo (figura 48), son las siguientes [37]:

- **Crecimiento:** la plántula está conformada solo por hojas a los alrededores de su tallo.
- **Floración inicial:** En la planta siguen apareciendo nuevas hojas y además la formación de una pequeña flor en el centro de la misma.
- **Formación de pellas:** La pequeña flor pasa a formar una pella en la yema terminal del tallo, y además la fase de floración inicial pasara a un segundo plano con los nuevos rebrotes a los lados del tallo.
- **Fructificación:** La pella o cabeza del brócoli llega a completar su desarrollo, ofreciendo un fruto y semillas.

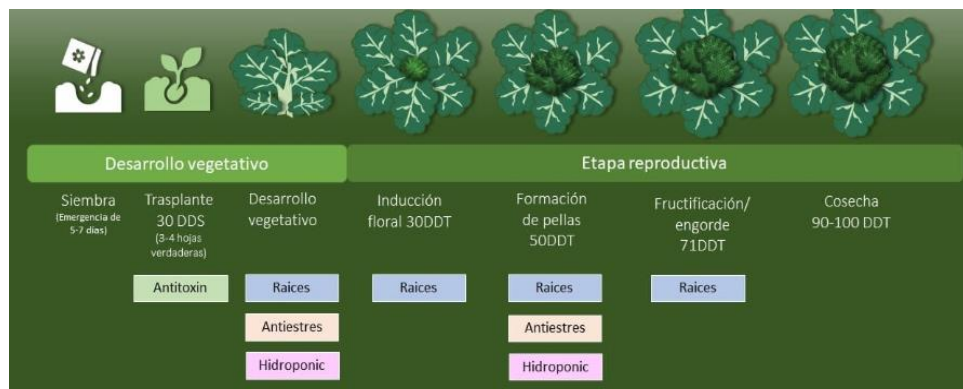


Figura 48. Ciclo de desarrollo del brócoli [37]

1.3.5.3 Agentes Patógenos y enfermedades en los cultivos de brócoli

Agentes Patógenos

Existe una gran variedad de agentes patógenos que afectan con gravedad al cultivo de brócoli, causando lesiones que dañen su producción, figura 49. En la tabla 7 se enlistan los más comunes.

Tabla 7. Agentes Patógenos en cultivos de brócoli

Agentes Patógenos en cultivos de brócoli	
Minador de hojas	Plaga extensa y dañina que puede llegar a destruir todo el cultivo.
Mosca de la col	Mosca con forma de pupa que afecta especialmente a las

	hojas.
Oruga de la col	Grandes daños para las hojas del cultivo, exceptuando los nervios.
Gorgojo de las coles o falsa potra	Abultamientos en las hojas producido por larvas y que dan paso a la formación de llagas.
Polilla de las crucíferas	Larvas causantes de dañar las hojas al dejarla totalmente tamizadas.
Pulguilla de la col	Aparecen en las plantas que se acaban de trasplantar y causan gran daño a las hojas y raíz.

Fuente: [38]



Figura 49.. Agentes Patógenos en las plantas de brócoli [38]

Enfermedades

Las enfermedades que se presentan en los cultivos de brócoli son muy perjudiciales para el desarrollo de la planta, llegando a bajar la producción en grandes cantidades, figura 50. En la tabla 8 se enlistan las enfermedades más comunes de este cultivo.

Tabla 8.. Enfermedades presentes en los cultivos de brócoli.

Enfermedades presentes en los cultivos de brócoli	
Alternaria	Formación de manchas negras con un color muy fuerte presente en la pella.
Potra de la col	Grandes protuberancias que afectan a las raíces.
Mildiu	Pequeñas manchas de color amarillo presentes en el fruto.
Rizoctonia	Enfermedad que afecta al tallo y raíz, adquieren un color negro.
Roya	Afecta a toda la planta, produciendo deformaciones, más que todo en sus hojas.

Fuente: [38]

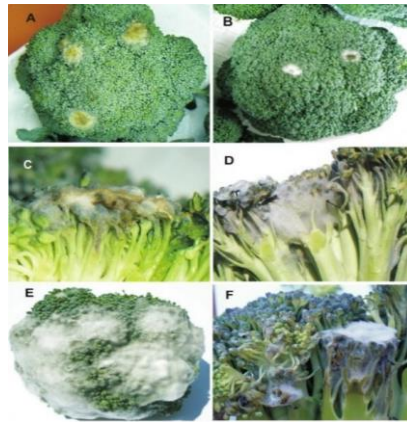


Figura 50. Enfermedades presentes en los cultivos de brócoli [38]

1.4. Objetivos

1.4.1. Objetivo General

Implementar un prototipo de sistema de teledetección aérea para control de agentes patógenos y enfermedades en cultivos de brócoli con el uso de visión artificial.

1.4.2. Objetivos Específicos

- Analizar los diferentes componentes y variables a considerar para el desarrollo de un dron de monitoreo de cultivos.
- Diseñar y desarrollar un dron para el monitoreo de cultivos de brócoli.
- Implementar un algoritmo de entrenamiento de visión artificial en base a imágenes espectrales para la detección de enfermedades en cultivos de brócoli.
- Desarrollar un sistema de teledetección para el control de enfermedades y agentes patógenos en cultivos de brócoli integrado en el dron.

CAPÍTULO II

METODOLOGÍA

2.1. Materiales

En el presente proyecto de investigación se utilizaron diferentes dispositivos para la implementación de un sistema de teledetección aérea para el control de agentes patógenos y enfermedades en cultivos de brócoli mediante el uso de visión artificial para mejorar la producción. La implementación del proyecto fue dividida en tres secciones: el armado del dron, el sistema de teledetección y el sistema de geolocalización, estos dos sistemas recogen datos y los muestran en una interfaz programada en código abierto por lenguaje Python.

Para el armado del dron se utilizaron distintos materiales como: frame o cuerpo de un dron compuesto por fibra de carbono, una placa de control u ordenador que es programada en el software Betaflight para recibir las órdenes del usuario, variadores ESC los cuales distribuyen y regulan la potencia que se entrega a los motores de 5 a 12 voltios, PDB es la placa que distribuye la energía tomando el voltaje de la batería LIPO (4 celdas), los motores que conjuntamente con las hélices ayudan al empuje del dron, modulo GPS implementado para el posicionamiento de la aeronave y ejecución de los planes de vuelo programados, módulo de VTX usado para la transmisión de video mediante la cámara y las gafas FPV, el buzzer que realiza la comunicación del dron mediante los pitidos que emite y el dispositivo de emisora – receptor que controla las funciones del dron en el aire. Luego de la construcción del dron, se adaptaron los sistemas de teledetección y geolocalización mediante el uso de soportes 3D. Estos sistemas usaron dispositivos como: Cámara GoPro Hero 5 mediante la cual se realiza la captura de imágenes para su procesamiento mediante la interfaz programada en el computador y además se almacena la información en un SGBD para el entrenamiento de la visión artificial, una Raspberry Pi 3 que en conjunto con la tarjeta Shield GPRS/GSM SIM800C proporciona el geoposicionamiento requerido para la localización del dron, la alimentación de estas tarjetas se las realiza mediante la adaptación a una batería de 3.7v con un módulo de

carga y descarga. Los datos proporcionados por el sistema de geolocalización se almacenan de igual manera en la base de datos de phpMyAdmin.

2.2. Métodos

2.2.1. Modalidad de investigación

El presente proyecto de investigación se desarrolló a partir de la sustentación bajo un estudio aplicado, dado que se utilizó los conocimientos adquiridos en el tiempo de formación académica y permitió la determinación de la problemática que se desarrolla por la presencia de agentes patógenos y enfermedades en los cultivos de brócoli proponiendo una solución con la implementación de un sistema mediante el uso de algoritmos de visión artificial.

Investigación bibliográfica, debido a que se recopiló información mediante una profunda investigación referente a sistemas de teledetección con drones y control de agentes patógenos para cultivos, en artículos, tesis, libros, revistas, informes, fichas bibliográficas, trabajos de investigación y bases de datos confiables que ayudaron al desarrollo satisfactorio del proyecto.

Se realizó investigación de campo, para adquirir información relacionada a los métodos y técnicas empíricas que se utilizan por el agricultor para el control de enfermedades en los cultivos de brócoli que son propicios para un óptimo desarrollo del producto.

Por último, se efectuó investigación experimental para determinar la detección oportuna de enfermedades y agentes patógenos en los cultivos de brócoli, mediante una serie de pruebas pertinentes en el monitoreo y control de los sistemas de teledetección y geolocalización, evaluando de esta manera la eficiencia del proyecto y la confiabilidad de los resultados obtenidos.

2.2.2. Recolección de información

Para la recolección de información, al ser una investigación de campo se emplearon varias guías de observación para valorar el método más óptimo en la prevención de agentes patógenos en los cultivos y con ello, analizarlas. Y mediante la investigación bibliográfica se realizó la determinación de las mejores técnicas para la prevención de enfermedades en los cultivos de brócoli.

2.2.3. Procesamiento y análisis de datos

Tras la recolección de información necesaria para el desarrollo del proyecto, se procede a seguir los siguientes pasos:

- Estudio de toda la información recolectada para su análisis.
- Estudio de las propuestas de solución planteadas para detección de enfermedades y agentes patógenos en los cultivos de brócoli con visión artificial.
- Planteamiento de la propuesta de solución, mediante información relevante que contribuya al desarrollo del proyecto.
- Propuesta de diseño e implementación del prototipo, para su control verificación de funcionalidad.

CAPÍTULO III

RESULTADOS Y DISCUSIÓN

3.1. Análisis y discusión de los resultados

El inicio operacional del sistema de teledetección aérea para el control de agentes patógenos y enfermedades en cultivos de brócoli con el uso de visión artificial consiste en el control de agentes patógenos existentes dentro de una superficie de terreno cultivado, mediante la captura de imágenes aéreas que son procesadas por un algoritmo de visión artificial para reconocer el estado de salud de cada parte del cultivo, además permite la obtención de los datos de geoposicionamiento como altitud y latitud de cada imagen capturada, mismos datos que serán almacenados en un base para el entrenamiento prolongado del algoritmo de visión artificial tras cada inspección llevada a cabo. El desarrollo de este proyecto ayuda a disminuir el déficit de pérdidas en cultivos de brócoli con la optimización de recursos, asimismo se considera como una herramienta versátil, debido a que puede ser aplicada o replicada al análisis de otros cultivos.

3.2 Desarrollo de la propuesta

El presente tema de investigación se desarrolló con el objetivo de dar solución a la problemática reflejada en el sector de la agricultura por la presencia de agentes patógenos y enfermedades en los cultivos de brócoli, dando solución mediante la implementación de un sistema de teledetección aérea para la captura de imágenes y posterior a ello el procesamiento digital mediante un interfaz de programación, con el objetivo de menorar el riesgo de pérdidas en la cosecha de este cultivo.

3.2.1 Requerimiento del sistema

El desarrollo del sistema de teledetección aérea destinado al control de agentes patógenos y enfermedades en cultivos de brócoli, permitirá conocer el estado de salud del cultivo mediante el procesamiento digital de las imágenes adquiridas, además la ubicación geográfica del dron en los diferentes sectores sobre los cuales se ha desarrollado la planeación aérea y de esta manera organizar el cuidado del cultivo dependiendo de la zona evaluada, para lo cual se detallan los siguientes requerimientos:

- Adquirir imágenes de la superficie del cultivo en intervalos de 2 segundos.
- Adquirir datos de geoposicionamiento de la nave que transporta el sistema, para generar un reporte de los sectores del cultivo ya inspeccionados.
- Establecer un módulo de carga para batería que alimente el sistema de geoposicionamiento durante la inspección del terreno.
- Procesar digitalmente las imágenes capturadas mediante un interfaz de programación en computador.
- Almacenar y organizar la información obtenida luego del proceso digital de las imágenes así como también los datos de geoposicionamiento en una base de datos local para el entrenamiento de la visión artificial.
- Visualizar e interpretar los datos obtenidos de inspección y geoposicionamiento de la plataforma sobre el cultivo a través de una interfaz didáctica.

3.2.2 Esquema General del sistema

El esquema general del sistema de teledetección de la figura 51 se compone de 3 etapas fundamentales, mediante las cuales se logra almacenar, recopilar y procesar la información obtenida al momento de la inspección de la superficie del cultivo. Cada una de estas etapas van a ser descritas a continuación para comprender de mejor manera su papel fundamental dentro del sistema:

La primera etapa, figura 52, corresponde al sistema de adquisición de datos, en donde se ejecuta la parte primordial del sistema como la captura de fotografías en intervalos de 2 segundos, teniendo en cuenta las características relevantes de la misma como nitidez, estabilidad, nivel de luz y resolución, apoyadas de la cámara la cual brinda especificaciones técnicas propias para el cumplimiento de estos parámetros. Por otro lado, en esta etapa se desarrolla un sistema de geolocalización mediante el cual se pretende obtener datos de posicionamiento a nivel de la superficie inspeccionada para el reconocimiento de las partes del terreno ya evaluadas.

La segunda etapa del proyecto, figura 53, corresponde al sistema de control y procesamiento de datos, en esta parte cada uno de los datos obtenidos mediante los diferentes sensores a bordo son procesados para la obtención de los resultados dentro de la superficie de cultivos. Además, en esta etapa mediante el sistema de control

que es exclusivamente para la manipulación del dron, se realizaran planes de vuelo convirtiendo al dron en uno autónomo y sin la necesidad de ser controlado por un mando. Estos planes de vuelo serán estructurados dependiendo de las dimensiones y la forma del terreno cultivado.

La tercera y última etapa, figura 54, corresponde a un sistema de visualización y almacenamiento de datos, en el cual podremos interpretar los análisis de la superficie evaluada de forma gráfica y a su vez almacenando cada uno de estos resultados dentro de una base de datos, la cual sirve para el entrenamiento prolongado de los algoritmos de visión artificial.

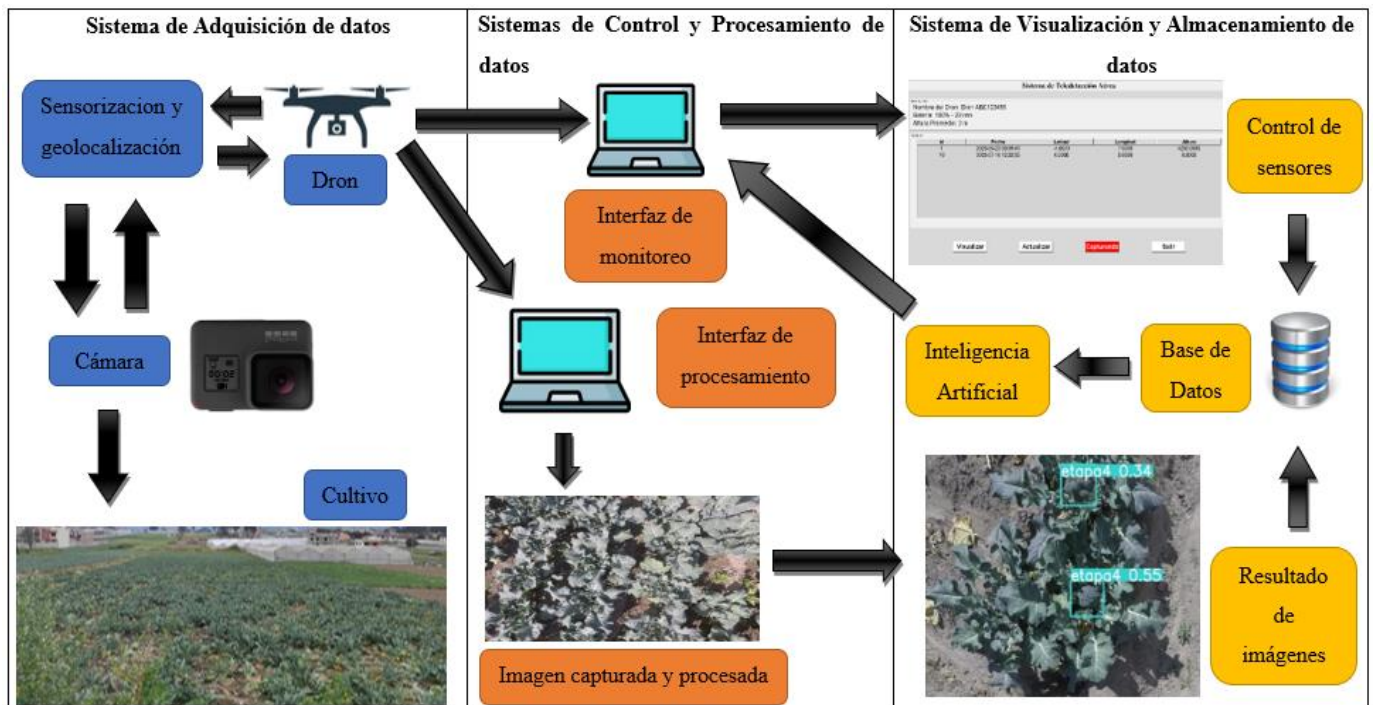


Figura 51. Esquema General del Sistema con cada una de sus etapas.

Elaborado por el investigador

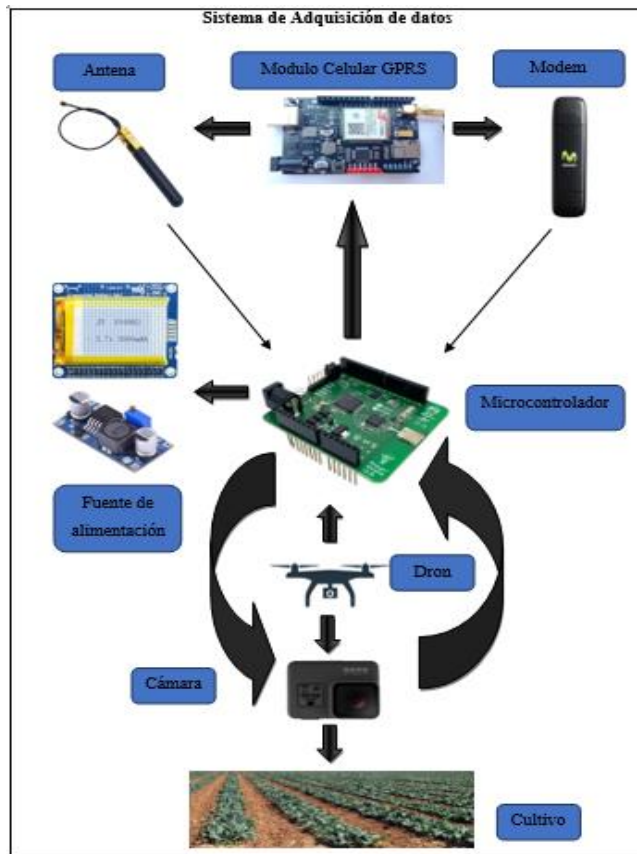


Figura 52. Etapa 1 – Sistema de Adquisición de datos

Elaborado por el investigado

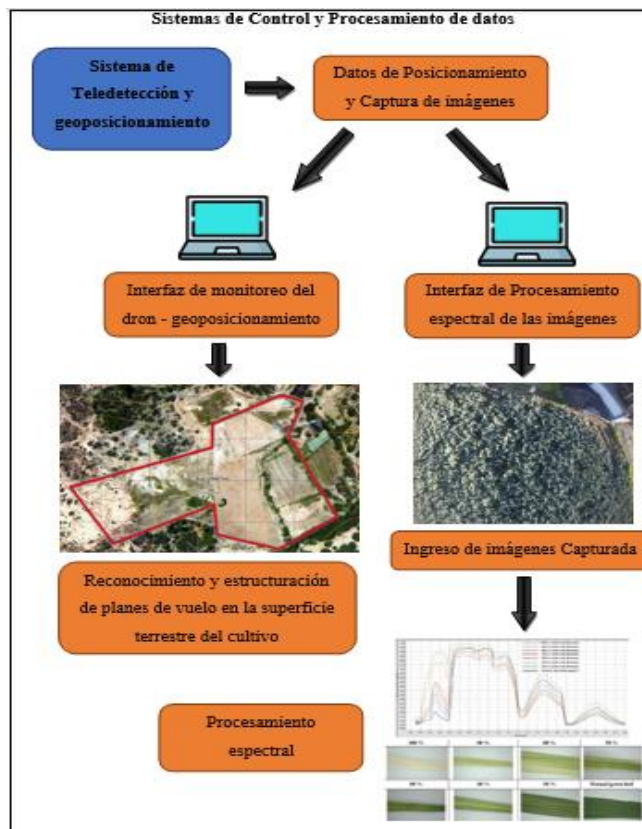


Figura 53. Etapa 2 – Sistema de Control y Procesamiento de datos

Elaborado por el investigador

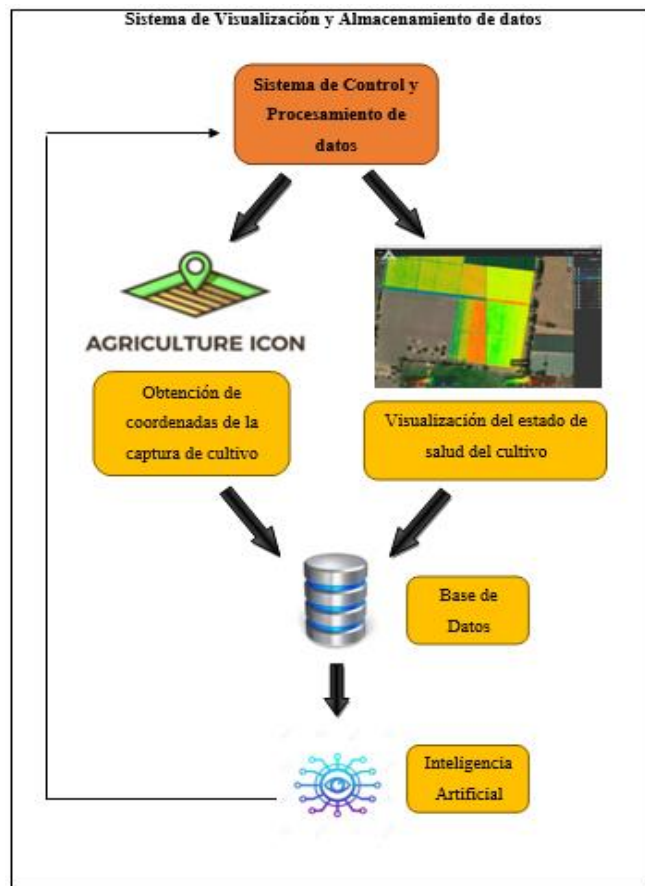


Figura 54. Etapa 3 – Sistema de visualización y Almacenamiento de datos



Elaborado por el investigador

3.2.3 Análisis de dispositivos

En Ecuador existen un número limitado de drones importados con el uso de cámaras multiespectrales para la aplicación en la agricultura tecnificada. Esta tecnología llegó al país en el año del 2021, con el objetivo de contrarrestar la deficiencia y pérdida de producción en los cultivos, ayudando así al ahorro de tiempo y de recursos económicos por parte del agricultor. Estos drones están compuestos por un GPS de precisión para conocer la superficie del terreno inspeccionado y una cámara espectral mediante la cual se logra localizar falencias como agentes patógenos o sequías dentro del cultivo. Esta tecnología fue aceptada con gran acogimiento dentro del campo agrícola, dado que ha presentado muy buenos resultados [39].

La tabla 9 nos muestra los drones multiespectrales dirigidos a la agricultura que existen dentro del mercado ecuatoriano:

Tabla 9. Drones con cámaras multispectrales dedicados a la agricultura tecnificada en Ecuador

Características Técnicas	DJI PHANTOM 4 Multispectral	DJI MAVIC 3 Multispectral	DJI MATRICE 300 RTK
			
Peso neto	1487 gramos	951 gramos	6.3 kilogramos
Dimensiones	350x290x141 mm	347.5x283x139.6 mm	810x670x430 mm
Tiempo de vuelo	27 minutos	43 minutos	55 minutos
Cobertura máxima	140 hectáreas por vuelo	200 hectáreas por vuelo	200 hectáreas por vuelo
Alcance de transmisión	7 kilómetros	15 kilómetros	15 kilómetros
Cámaras espectrales	5	4	1
Cámaras RGB	1	1	2
Intervalo de fotografías	0.5 segundos	0.7 segundos	0.5 segundos
Tecnología	RTK	RTK	RTK
Planeación de viajes	Si	Si	Si
Sistema	OcuSync	OcuSync	OcuSync
Numero de ejes	3	4	4
Almacenamiento interno	No	No	Si
Batería	5870 mAh	5000 mAh	5935 mAh
Carga	Rápida	Rápida	Rápida
Precio	\$ 5000	\$ 4190.21	\$ 9990

Fuente: Elaborado por el Investigador, en base a [40]–[44]

Cada uno de estos drones llevan incorporados un sistema de teledetección mediante la presencia de cámaras RGB y cámaras multispectrales; sin embargo, el dispositivo

más completo y compacto para presentar buenos resultados en la inspección de cultivos es el DJI MATRICE 300 RTK. Este dron posee características importantes a considerar para el desarrollo del proyecto, debido a que permite conocer los parámetros necesarios en los cuales se puede hacer énfasis al momento de seleccionar el dron para la implementación del sistema.

3.2.4 Selección de los elementos hardware




Los elementos que compondrán a cada una de las etapas del sistema de teledetección, han sido expuestos mediante los diagramas de las figuras 51,52,53,54. A continuación se realiza una comparativa de los parámetros técnicos entre cada elemento necesario para la implementación del sistema y de esta manera seleccionar los dispositivos o elementos que brinden mayor portabilidad y resultados de operación satisfactorios.

3.2.4.1 Elementos hardware Etapa 1

Dron (UAV)

Es el elemento principal dentro del proyecto de investigación, dado que cumple la función de plataforma de soporte para los sistemas a implementarse. Como se observa en la tabla 9, existen hasta el momento 3 drones dedicados plenamente a la inspección de cultivos dentro de la agricultura tecnifica, debido a esta minoría sus precios son elevados. Por lo cual se decide realizar la construcción de un dron tomando en cuenta las características que más se asemejen a los drones profesionales tales como peso, potencia, dimensiones, tiempo de vuelo, tipo de batería, alcance de transmisión, entre otros; en la tabla 10 se presentan una variedad de drones de alto vuelo que podrían ser útiles en la implementación del proyecto.

Tabla 10. Características de drones de fácil ensamblaje

Características	Realacc x210Pro FPV	BETA FPV Beta85X	KINGKONG 260	WALKERA FURIOUS 320
				
Material	Fibra de carbono	Fibra de carbono	Fibra de carbono	Fibra de carbono
Tipo	COPTER	COPTER	COPTER	COPTER

Peso	98 gramos	90 gramos	90 gramos	810 gramos
Dimensiones	300x260x110 mm	85x85x60m m	248x240x50m m	328x254x118m m
Tiempo de vuelo	20 minutos	15 minutos	15 minutos	10 minutos
Peso de soporte	500gramos	100 gramos	300 gramos	420 gramos
Tiempo de vuelo con soporte máximo	15 minutos	8 minutos	10 minutos	5 minutos
Alcance de transmisión	800 metros	620 metros	560 metros	750 metros
Capacidad de batería	4S de 1300mAh	2S de 450mAh	4S de 1300mAh	5S de 2600mAh
Peso de batería	302 gramos	155 gramos	302 gramos	360 gramos
Cámara FPV	Si	Si	Si	No
Motores	BR 2300 Kv	ECO1106 6000kv	BR 2300 Kv	WK-WS1700KV
Número de motores	4	4	4	4
Potencia de motores	408 watts	146 watts	408 watts	350 watts
Tipo de vuelo	Configurable	Free Style	Configurable	Configurable
Velocidad Máxima	150 km/h	210 km/h	150 km/h	220 km/h
Autonomía	Configurable	No	Configurable	Configurable
Máxima velocidad de viento operacional	20 km/h	10 km/h	20 km/h	30 km/h
Frecuencia de Recepción	2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz

Fuente: Elaborado por el investigador, en base a [45]–[47]

Al identificar las características técnicas de cada uno de los drones comerciales que se encuentran disponibles en el mercado, se procede a elegir el que más se acerque a nuestras necesidades y exigencias en el entorno de implementación.

La selección se dio por el dron Realacc x210Pro FPV, figura 55, el cual posee características importantes que se dirigen plenamente al desarrollo del proyecto. El ensamblado de este dron es de un nivel medio, dado que no requiere muchos conocimientos en el armado de la parte física, pero si al momento de programar la placa controladora. Posee características muy relevantes como su peso de 98 gramos gracias a que el material del cual este fabricado en su mayoría es fibra de carbono, permitiendo de esta manera que el dron sea compacto y fácil de movilizar. Sus dimensiones son de 300x260x110mm brindando de esta manera un espacio amplio en su parte inferior y superior para colocar los accesorios necesarios soportando un peso adicional de hasta 500 gramos. Posee una batería tipo LiPO de 1300 mAh que consta de 4 celdas con 325mAh cada una, con lo cual el tiempo de vuelo promedio ronda entre los 20 minutos y con el peso adicional entre los 15 minutos. Este dron funciona con 4 motores BR 2300 Kv los mismo que tiene una potencia de 408 watts, estos motores son semiprofesionales por ende ayudan al dron alcanzar grandes velocidades y a mantener su estabilidad en el aire gracias a su gran potencia en las 4 hélices, lo que es muy importante debido a que este dron trabajara como una plataforma de movimiento y su estabilidad es muy importante al momento de la captura de imágenes, además que consta de configuración de planes de vuelo por su autonomía presente al acondicionarle una tarjeta GPS a su placa de control.




Figura 55. Dron Realacc x210Pro FPV [46]

Microordenador

Los microordenadores son utilizados para desarrollar proyectos académicos de gran amplitud. Están conformados por pines predestinados a propósitos generales que varían dependiendo de la marca y el modelo, necesitan una alimentación de 5.3 voltios aproximadamente, constan de un almacenamiento interno para procesar información de archivos o ejecutar aplicaciones y además tienen puertos de comunicación los cuales están estandarizados por sus protocolos.

A continuación, en la tabla 11 se presenta dos de los microordenadores más utilizados para proyectos académicos con sus principales características:

Tabla 11. Características de microordenadores comerciales

Características	Raspberry Pi 4	Banana Pi BPI-M5
		
Dimensiones	85x53 mm	92x60 mm
Peso	45 gramos	48 gramos
Alimentación	15.3 W – 5.1 V – 3Amp	5 V – 2Amp
Procesador	ARM Cortex-A72	A20 ARM Cortex-A7 Dual-Core
Memoria	1 GB / 2 GB / 4 GB LPDDR4 SDRAM	1GB DDR3 DRAM
Sistema Operativo	GNU/Linux ARM	AROS
GPU	Broadcom VideoCore VI	ARM Mali400MP2Complies
Núcleos	4	4
Frecuencia	1.5GHz	800 MHz
Conectividad	Bluetooth 5.0, Wi-Fi 802.11ac, Gigabit Ethernet	Bluetooth 4.2, WiFi 802.11 b/n/g/e/i
Puertos	GPIO, micro HDMI, USB, USB-C, Conector de audio, micro SD.	USB 3.0, USB 2.0, USB-C

Fuente: Elaborado por el investigador, en base a [48]–[50]

Mediante el análisis de cada una de las características técnicas, se escogió la tarjeta Raspberry Pi 4, figura 56, debido a las especificaciones técnicas que posee para facilitar la implementación del proyecto de investigación. Esta tarjeta consta de 40 pines GPIO mediante los cuales se puede integrar accesorios tales como sensores que permitan la recolección de datos. Posee una capacidad de almacenamiento de 4GB lo cual permite renderizar la información recolectada, brindando una confiabilidad mayor al momento de la captura y recolección de imágenes, dado que estas deben ser almacenadas para el posterior análisis del nivel de salud en intervalos de tiempo de 2 segundos, además el programa desarrollado en conjunto con las librerías utilizadas y el código de programación necesitan un espacio óptimo de almacenamiento en un 0.30% para esta tarjeta de desarrollo, siendo una gran ventaja en comparación a la tarjeta Banana Pi. Para la conectividad, esta tarjeta consta de varios protocolos de comunicación como Bluetooth 5.0, WiFi 802.11ac y Giga Ethernet, lo cual es primordial para la comunicación entre ordenador, tarjeta y módulo GPRS.



Figura 56. Microordenador Raspberry Pi 4 [50]

Módulo de transmisión GPRS

El módulo de transmisión GPRS es considerado como una terminal remota para GSM/GPRS. Mediante el cual se puede realizar conexiones inalámbricas para red móvil como texto, datos y voz. Además, permite conocer la localización GPS gracias a que cuenta con un GPRS integrado.

Se decidió trabajar con la tarjeta SIM 808, figura 57, que posee características primordiales para el desarrollo del proyecto, tales como un consumo de energía de 2 Amperios lo cual permite trabajar con la batería portable del microordenador Raspberry Pi en conjunto, dado que esta tarjeta es compatible con la mayoría de módulos y tarjetas programables. Posee un GPRS de clase 12 con un enlace de

comunicación de 85,6 kbps de subida y de bajada que proporciona la conexión a internet obteniendo de esta manera datos de GPS en longitud y altitud con una precisión en posición horizontal mayor a 2,5m de error circular probable (CEP), con lo cual se estima un 95% en los rangos aceptables. Consta de 2 antenas, una antena GPS para obtención de datos de localización que trabaja a una frecuencia de 1578,4MHz y otra antena GSM que permite la conexión para voz, SMS (texto), datos con una frecuencia de 1800MHz [51].



Figura 57. Módulo SIM 808 GPRS/GSM [51]

Mobile Wi-Fi

Este dispositivo es un Wifi portátil o también considerado como un punto de acceso móvil mediante el cual logra la conexión a internet para cualquier dispositivo con tarjeta de red Wireless. Estos dispositivos son capaces de albergar 5 usuarios, dependiendo de la marca y modelo.

Para el desarrollo del proyecto se eligió trabajar con Mobile Wifi Huawei E586, figura 58, perteneciente a la operadora Movistar, gracias a sus características relevantes como pantalla OLED que brinda la información del estado de conexión y demás funciones como nivel de batería, hora, velocidad de carga y descarga de datos. La fuente de alimentación para este dispositivo consta de una batería de 1500mAh. La velocidad de transmisión es de 54 Mbps para WiFi de 2.4 GHz, esta característica es importante dentro del sistema debido a que es necesaria la transmisión de datos para la interacción entre usuario y sistema [52].



Figura 58. HUAWEI E586 [52]

Cámara

La cámara es el dispositivo encargado de capturar las imágenes necesarias bajo ciertos parámetros como nitidez, brillo, iluminación y enfoque. En la tabla 12 se muestran los parámetros técnicos de las cámaras seleccionadas en referencia a una resolución igual o superior a 12 megapíxeles.

Tabla 12. Características técnicas de cámaras de alta definición

	Cámara GoPro Hero 5	DJI Pocket 2	APEMAN A87
Características			
Dimensiones	62×44.6×32.7mm	124.7×38.1×30 mm	61 x 45 x 24 mm
Peso	117 gramos	117 gramos	65 gramos
Memoria	Micro SD hasta 128GB	Micro SD hasta 256GB	Micro SD hasta 128GB
Batería	1220 mAh	875 mAh	1050 mAh
Duración de batería	150 minutos en su máxima calidad	140 minutos en su máxima calidad	80 minutos en su máxima calidad

Resolución imagen	12 megapíxeles	64 megapíxeles	20 megapíxeles
Modos de captura	Bucle, Ráfaga, Time lapse, Baja luz	Normal, Time-lapse, Hiperlapse	Ráfaga, Normal
Conectividad	Wifi, Bluetooth, GPS, HDMI	Wifi	Wifi
Estabilización	4 ejes	3 ejes	3 ejes – 4K
Resistencia al agua	Si, hasta 60 metros	Si, hasta 40 metros	Si, hasta 40 metros

Fuente: Elaborado por el investigador, en base a [53]–[55]

Una vez identificados los parámetros técnicos de cada una de las cámaras, se llega a la decisión de utilizar la cámara GoPro Hero 5 Black gracias a sus características favorables que podrán ayudar en la implementación del proyecto, tales como su tamaño compacto de 62×44.6×32.7mm, llegando de esta manera a tener un peso de 117 gramos y su forma rectangular que ayuda a su estabilización en 4 ejes, características que facilitan la portabilidad de la misma en la plataforma aérea. Además, posee una batería interna de 1220mAh con una duración de 150 minutos en su máxima calidad de imágenes o video, este punto es favorable en el sistema debido a que la batería debe tener un periodo de duración mayor o igual a 25 minutos, que es el tiempo de vuelo del dron. Este dispositivo tiene conectividad inalámbrica mediante Wifi y Bluetooth que es importante para la transferencia de las capturas de imágenes al sistema de control y procesamiento de datos, aunque por otro lado cuenta con una entrada tipo C para la transferencia física de datos, que es una opción más confiable al momento de transferir información sin pérdidas por interferencias. Al ser una cámara deportiva tiene una buena resistencia como por ejemplo al agua, hasta una profundidad de 60 metros con el uso de una carcasa especial. La característica más valorada en esta cámara es la toma de imágenes con una calidad de hasta 12 megapíxeles con tipos de captura como bucle, ráfaga, time lapse y baja luz, con esta calidad de imagen se pueden obtener buenos resultados en el sistema debido a que su alta definición facilitaría la detección de anomalías en el cultivo.

Modulo GPS

El módulo GPS es un sistema de posicionamiento global que ayudara a la referenciación terrestre del dron en su desplazamiento al proporcionar datos como su ubicación, el nivel de velocidad y la latitud a la que se encuentra el dron cuadricóptero. Este módulo en combinación con el controlador de vuelo presente en el sistema electrónico convierte al dron en uno autónomo, permitiendo realizar planes de vuelo o vuelos preprogramados para su ejecución sobre una superficie terrestre.

Para el proyecto de investigación se eligió el módulo AP.25E GPS/GALILEO que se muestra en la figura 59, debido a sus especificaciones técnicas como tamaño y peso propicios que no influyen de manera negativa en el vuelo del dron cuadricóptero.



Figura 59. AP.25E GPS/GALILEO

Elaborado por el investigador

3.2.5 Selección de elementos software

Los elementos de software necesarios para la implementación del sistema de teledetección, se encuentran identificados en los diagramas de bloques correspondientes a la etapa 2 de control y procesamiento de datos y a la etapa 3 de visualización y almacenamiento de datos que se presenta en las figuras 51, 53 y 54. Los elementos de software deben contener parámetros específicos que logren cubrir las necesidades de la implementación del proyecto, por ende se seleccionan cada uno de estos softwares mediante un análisis profundo de características y versiones actuales.

3.2.5.1 Elementos software Etapa 2

Para la etapa 2 referente al control y procesamiento de datos es necesario realizar un interfaz gráfico para la interacción con el usuario. Este interfaz será desarrollado con el uso del software de programación Python versión 3.10.10, esta versión es una de las más recientes puesta a disposición en febrero del 2023 siendo de uso libre y gratuito. Python es considerado uno de los lenguajes de programación más completos y de alto nivel, esto gracias a que conlleva un uso extenso de sus librerías que están orientadas a Big Data e Inteligencia Artificial. Python en relación con la Inteligencia Artificial se basa en un entrenamiento de modelos autónomos, los cuales brindan un aprendizaje para desarrollar una acción de forma independiente, esto partiendo de las instrucciones desarrolladas por el programador para que la tecnología reaccione por cuenta propia con una respuesta adecuada [56].

El software Python posee muchos beneficios en cuanto al desarrollo del proyecto de investigación debido a que su uso es general y puede ser utilizado por cualquier tecnología o dominio. Su estructura de programación gestiona una memoria automática con lo cual se genera una menor cantidad de líneas de programación logrando de esta manera una mejor compresión del lenguaje por parte del desarrollador. El uso de Python es de código abierto y ejecuta librerías tales como Numpy, Pybrain y Scipy para el aprendizaje autónomo, convirtiéndose de esta manera en una característica clave para el desarrollo de la interfaz del proyecto, gracias a sus aplicaciones para Inteligencia Artificial [56].

3.2.5.2 Elementos software Etapa 3

Para el desarrollo de la última etapa referente a la visualización y almacenamiento de datos, es necesario el uso de una base de gestión de datos y plataformas o algoritmos que faciliten el entrenamiento de la visión artificial.

phpMyAdmin

Este sistema de gestión de base de datos ayuda a la visualización gráfica de la información almacenada para de esta manera ser evaluada e interpretada por el usuario. La interfaz presentada dentro de esta base de datos es amigable con el usuario y de fácil acceso mediante servidores remotos. Los datos almacenados dentro de esta base pueden ser modificados a disposición y requerimientos del usuario. En el desarrollo del proyecto de investigación es de suma importancia administrar cada

uno de los datos obtenidos al momento de realizar la inspección de los cultivos, dado que el almacenamiento de esta información puede ser utilizada para agrandar el aprendizaje de los algoritmos de visión artificial.

Makesense.ia

La herramienta online Makesense.ia se utiliza para etiquetar, reconocer y distinguir objetos de una imagen, ofrece ventajas tales como su uso gratuito debido a que se ejecuta mediante un navegador. Esta herramienta es muy recomendada para trabajos de visión artificial, gracias a su amigable plataforma y fácil preparación del conjunto de datos. Trabaja con formatos como VOC, JSON, YOLO, XML, CSV, VGG.

GoogleColab

Esta herramienta online permite ejecutar, escribir o editar un código de programación de Python a través del navegador. Está dirigido netamente para desarrolladores de algoritmos de aprendizaje autónomo, como visión artificial.

YOLO V-5

El algoritmo de entrenamiento de visión artificial YOLO v-5 es uno de los más utilizados en proyectos relacionados a aprendizaje autónomo con respecto al reconocimiento de imágenes. Este algoritmo tiene como base una red convolucional mediante la cual se detectan delimitaciones por cada clase de objeto, este sistema interactúa con la imagen completa para contrarrestar conflictos en la regresión lineal y por ende la ejecución de algoritmos de la red neuronal. Este algoritmo de entrenamiento se considera favorable para la implementación del proyecto debido a que no existe la necesidad de convertir la imagen a diferentes formatos para un análisis individual, sino realizar el aprendizaje con la imagen completa y detectar anomalías que se tenga dentro de cada imagen.

3.2.6 Diseño del sistema

El diseño del sistema utilizado para el proyecto de investigación está basado en una arquitectura de inteligencia artificial destinada a la inspección de cultivos de brócoli para identificar agentes patógenos y enfermedades que puedan deteriorar la calidad y productividad del cultivo mediante el uso de algoritmos de visión artificial y aprendizaje autónomo de la tecnología YOLO v 5x. En la figura 60, se presentan las etapas principales que componen este sistema de teledetección como son: el sistema

de adquisición de datos, el sistema de control y procesamiento, el sistema de visualización y almacenamiento, cada una de estas con sus componentes electrónicos seleccionados.

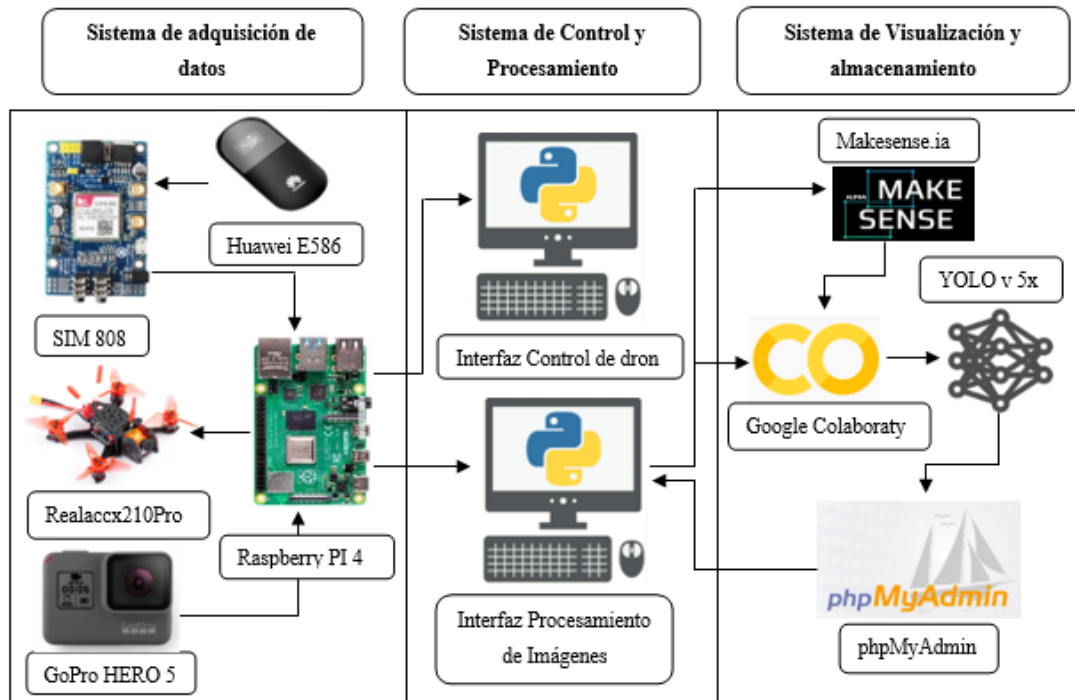


Figura 60. Diseño del sistema de teledetección

Elaborado por el investigador

Con cada uno de los elementos electrónicos y el modelo de entrenamiento de visión artificial seleccionados para cada una de las etapas del sistema, se tiene comienzo con la primera etapa que es el sistema de adquisición de datos e imágenes a través de la tarjeta Raspberry Pi 4. Esta tarjeta funciona como la unidad central del sistema dado que en ella se programan las funciones a ejecutar para cada uno de los sensores dependientes como la captura de imágenes a través de la cámara GoPro Hero 5 en intervalos de tiempo de 2 segundos y la obtención de datos de posicionamiento a través de la tarjeta SIM808, estos datos conllevan a su análisis a través de la conexión IP con el módulo Huawei E586. Esta parte del sistema, lleva consigo la utilidad de un dron autónomo, el cual sirve como plataforma de traslado o movimiento para los elementos electrónicos. Recolectados los datos de inspección pasan a un sistema de control y procesamiento mediante los interfaces de usuarios desarrollados. Para las imágenes capturadas se procede a interactuar con el interfaz de procesamiento de imágenes en donde se obtiene el análisis del estado saludable y

presencia de agentes patógenos o enfermedades presentes en el cultivo, mientras que por otro lado el interfaz de control está dedicado a los datos de geolocalización o posicionamiento del dron al momento de realizar la inspección. Esta información se verá reflejada de forma gráfica y guardada en una base de datos creada mediante phpMyAdmin para ser utilizada en el entrenamiento de los algoritmos de visión artificial del modelo YOLO v 5x. El entrenamiento procede a realizarse a través Google Colab y Makesense.ia con el objetivo de obtener una gran cantidad de datos recopilados y sean menos los riesgos a obtener falsos positivos en la ejecución del sistema.

3.2.7 Diseño e implementación del sistema general

Para iniciar con la implementación del sistema, en primera instancia fue necesario desarrollar la plataforma aérea que dará movilidad a los elementos electrónicos del sistema, en este caso se trata del dron Realaccx210Pro. Anteriormente se conoció cada una de sus características más relevantes para de esta manera comenzar con su diseño y construcción.

3.2.7.1 Diseño de la estructura de dron Realaccx210Pro

El diseño de la estructura del dron se realizó con el fin de reconocer el frame o cuerpo de esta plataforma aérea y consecuente a esto diseñar los soportes extras para la ubicación de los accesorios adicionales como cámara, micrófono y módulo GPS, los mismo que se acoplaran al dron para la captura y envío de datos.

A través del uso del software Autodesk Fusion360 se desarrolló el diseño estructural 3D de cada parte física del dron cuadricóptero Realaccx210Pro (UAV), tales como: Base superior del dron, base inferior, base de batería, brazos exteriores, hélices y los motores, cada una de estas partes se diseñaron de manera individual y al final se ensamblaron para que el dron tome su forma original.

Base Superior del Dron

En la figura 61, se muestra la estructura 3D de la base superior del dron que tiene dimensiones de 220mm de largo, 250mm de ancho y de profundidad de extrusión de 5mm, además se diseñó los espacios de cada componente ubicados sobre esta base superior, que se las representa con pequeñas grietas de diferentes formas para que al

momento de ensamblar cada componente se tenga una mejor visualización del entorno.

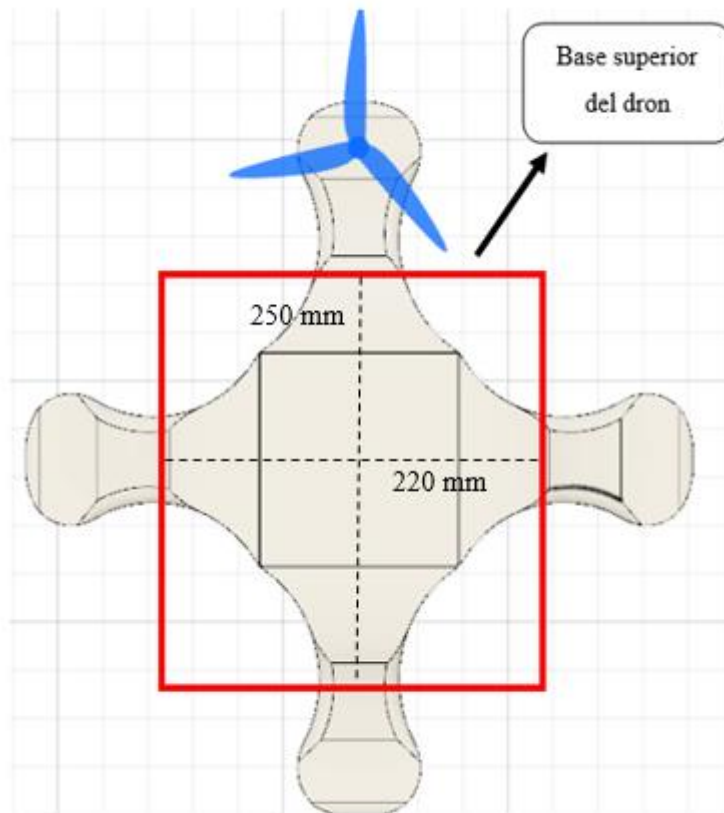


Figura 61. Diseño Base superior del dron

Elaborado por el investigador

Base Inferior del Dron

La base inferior del dron posee dimensiones 220mm de largo, 250 mm de ancho y una profundidad de extrusión de 5mm, obteniendo de esta manera un área de 55000 mm², como se puede observar en la figura 62. En cada extremo de la base se adjuntó 4 extremidades que servirán de soporte para el UAV, con un largo de 49.65 mm y un ancho de 30mm en cada uno.

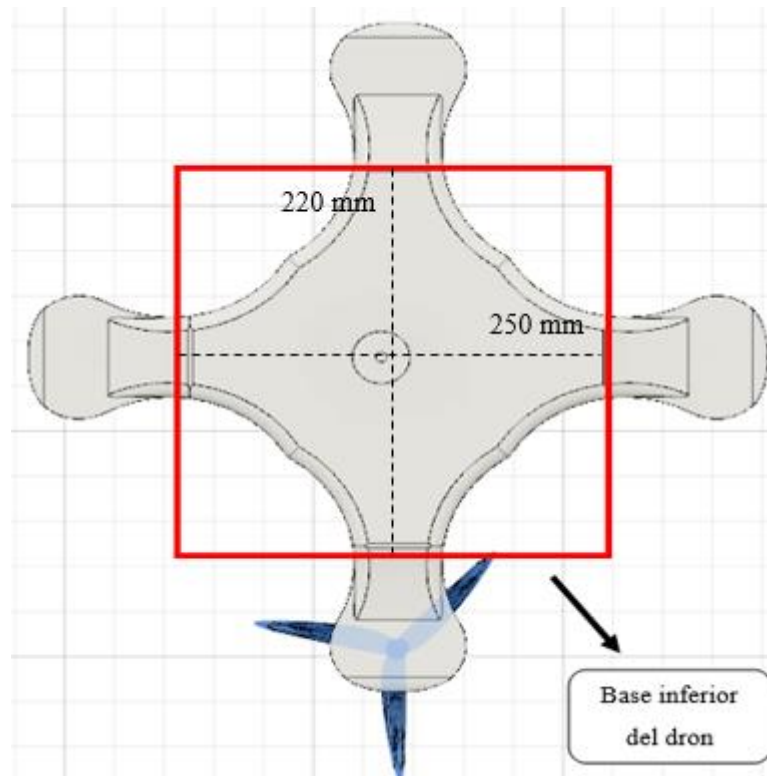


Figura 62. Diseño base inferior del dron

Elaborado por el investigador

La base inferior es la parte fundamental del UAV, debido a que aquí se van a ensamblar la base superior y todos los componentes que conformen la estructura electrónica del dron, además en la base inferior irán integrados cada uno de sus cuatro brazos con sus respectivos motores y sus hélices en cada uno de los extremos correspondientes.

Base de la batería

En la figura 63 se observa el diseño 3D de la base de la batería que tiene la forma de una mesa, donde iría colocada la misma para energizar a todo el dron y puede desplazarse en el aire. Las dimensiones que tiene esta base son 90mm de largo, 140 mm de ancho y una profundidad de 5mm, obteniendo un área de 12600 mm². Los soportes de la base tienen una dimensión de 70mm de largo y el círculo que se forma en la parte inferior del soporte tiene un diámetro de 6mm.

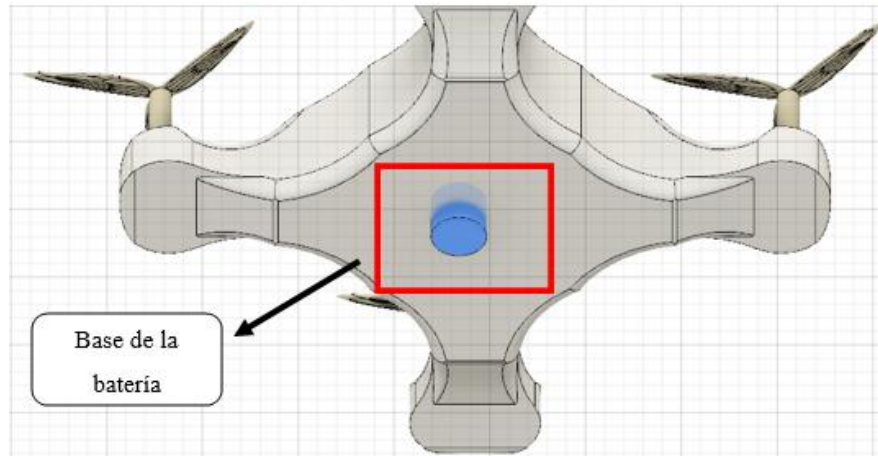


Figura 63. Diseño base de la batería

Elaborado por el investigador

Brazos exteriores

Los brazos exteriores para el dron son fundamentales debido a que en ellos se integran los motores y las hélices para el movimiento del dispositivo, como se observa en la figura 64.

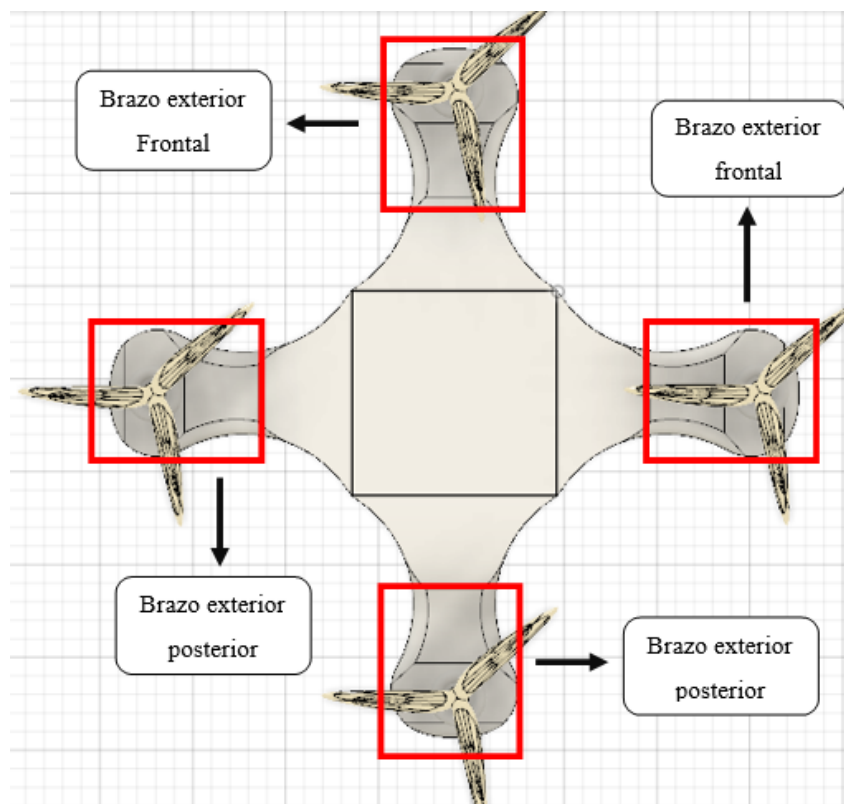


Figura 64. Diseño de los brazos exteriores

Elaborado por el investigador

En la figura 65 se muestra la parte central que compone el brazo con las siguientes dimensiones 100mm de largo, 40 mm de ancho y un área de 4000 mm².

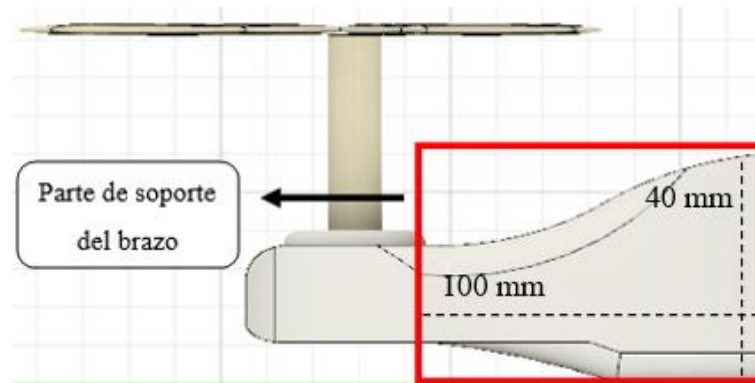


Figura 65. Diseño parte central del brazo exterior

Elaborado por el investigador

En la figura 66 se muestra la parte de enganche que por un extremo se fija al brazo en la base inferior del dron y en su otro extremo se fija a cada uno de los motores. Tiene dimensiones de 112mm de largo, 15 mm de ancho, 5mm de profundidad y un área de 1680mm².

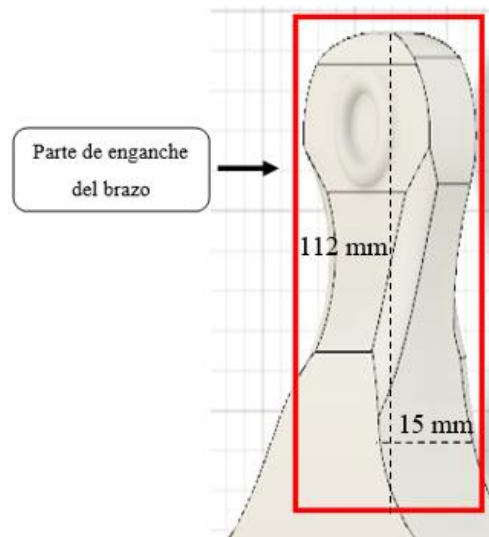


Figura 66. Diseño de la parte de enganche del brazo exterior

Elaborado por el investigador

Motores

Los motores son la parte de arranque y fuerza de suspensión para el dron, figura 67, aquí van incorporadas las hélices, permitiendo que el UAV despegue y realice su

vuelo. Las dimensiones del diseño 3D de este componente son 13.85mm de radio, un diámetro de 26.70 mm y una longitud de 25 mm en el cuerpo.

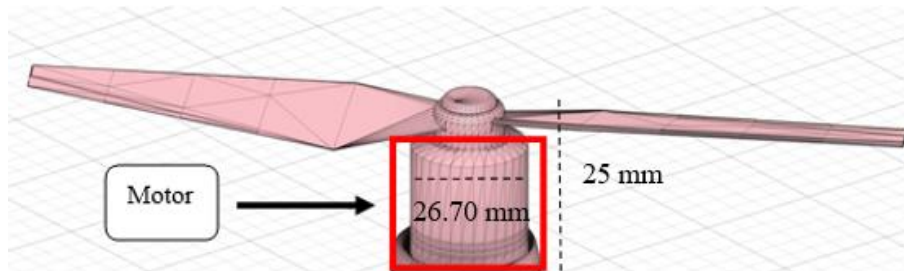


Figura 67. Diseño de motores

Elaborado por el investigador

Hélices

Las hélices conllevan la función de darle velocidad al dron dependiendo del número de hélices que este conformado y su grado de inclinación. En la figura 68, se observan las dimensiones del diseño, las cuales son 45.598 mm de largo para cada lado, 13.19 mm de ancho y con un grado de inclinación 20°. En la parte del medio se encuentra un orificio con un radio de 12.6mm y un diámetro de 6.3mm, el cual fija la hélice al motor.

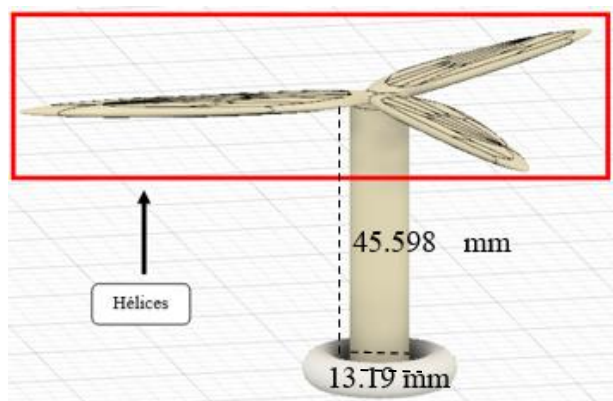


Figura 68. Diseño de hélices

Elaborado por el investigador

Cámara y soporte de cámara

La cámara presente en el dron nos da la función de FPV es decir video en primera persona, tiene la utilidad de identificar la posición que está sobrevolando la aeronave en tiempo real. En la figura 69 se observa el diseño de la cámara que tiene la forma

de un cubo donde se aloja el lente de 2.1mm, el cubo tiene las siguientes dimensiones: 62mm de largo, 44.6 mm de ancho y un área de 2765.2mm². Además, se diseñó un soporte de protección con las siguientes dimensiones: 63.5mm de largo, 46.1 mm de ancho y un área de 2927.35mm².

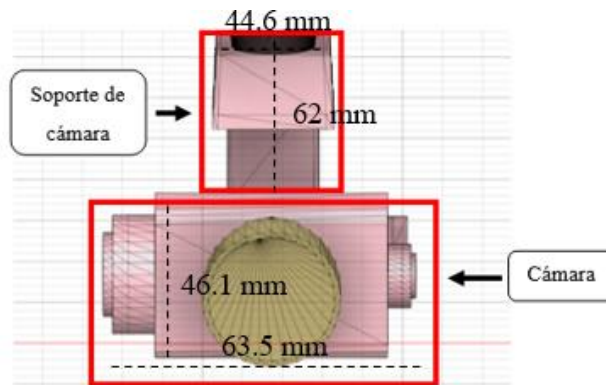


Figura 69. Diseño de cámara

Elaborado por el investigador

Antena Receptora de señal

Esta antena, figura 70, recibe cada una de las señales emitidas por el radiocontrol terrestre. El frame de soporte para la antena está diseñada con las siguientes dimensiones: 31.646mm de longitud de contorno, 25.3mm de ancho y un área de 800.64mm². La antena tiene 95mm de longitud y un diámetro de 8mm.

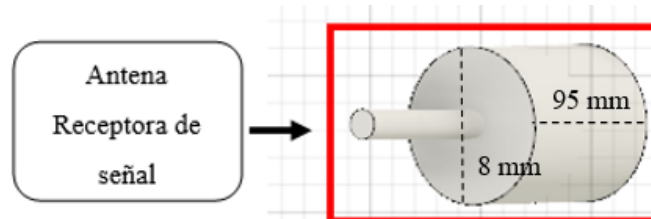


Figura 70. Diseño de Antena receptora de señal

Elaborado por el investigador

Diseño 3D renderizado del dron compuesto por cada una de sus partes

Se presenta en la figura 71, la renderización de la estructura física del dron con cada uno de sus componentes mecánicos y electrónicos. Este diseño fue desarrollado acorde al modelo de dron Realaccx210Pro con el objetivo de reconocer cada uno de sus elementos y hacer más viable su construcción.



Figura 71. Diseño 3D renderizado del dron

Elaborado por el investigador

3.2.7.2 Implementación de dron Realaccx210Pro

El primer paso a realizar es montar el cuerpo del dron, debido a que este es el soporte para los demás componentes que influyen en su funcionamiento.

Se debe tener en cuenta que existen diversos factores que influyen al momento de realizar el montaje de un dron, uno de esos factores es el peso neto que va a poseer la aeronave, para esto se debe tener en cuenta el tipo de frame que se va a utilizar.

Frame

El frame en si trata acerca del cuerpo del dron en el cual van montados los elementos que se irán detallando más adelante. Para este caso al ser un dron de carreras se recomienda la adquisición de un frame de fibra de carbono por su elevada resistencia ante los impactos además de su peso ligero. Para este caso se tiene el FRAME MARTIAN II, con las siguientes especificaciones presentes en la tabla 13:

Tabla 13. Especificaciones del Frame Martial II

FRAME MARTIAN II	
Material	Fibra de carbono
Tamaño (largo x ancho)	220 x 250 mm
Peso	160 gr
Distancia entre ejes	220 mm

Grosor del brazo	4 mm
Grosor de la placa (superior e inferior)	1.5 mm

Fuente: Elaborado por el investigador, en base a [57]

Montaje de Frame

Para el montaje del frame o cuerpo del dron existen piezas minúsculas las cuales pueden ser extraviadas con facilidad como se puede observar en la figura 72, por ende se debe tener precaución al momento de ensamblarlo dado que está compuesto de tornillos que van a ir ubicados específicamente en un lugar de acuerdo al modelo de frame además, estos deben ir bien sujetos debido a que al principio de la práctica en el manejo del dron puede llegar a sufrir algunos choques o movimientos bruscos por esta razón se recomienda ubicar silicona bajo los tornillos para mayor fijación y con ello evitar la pérdida de los mismos.



Figura 72. Frame y componentes de ensamblaje

Elaborado por el investigador

Cabe destacar que se debe contar con destornillador plano y estrella, además de un hexagonal de 2mm y un dado del # 8 para los diferentes tornillos que componen el cuerpo, figura 73.



Figura 73. Montaje de Frame

Elaborado por el investigador

Por consiguiente, es necesario conocer los modos y los diagramas de conexión de cada uno de los elementos presentes en la parte eléctrica y motriz del dron. En la figura 74, se presentan un reconocimiento general de los componentes relacionados entre sí antes de efectuar su conexión respectiva.

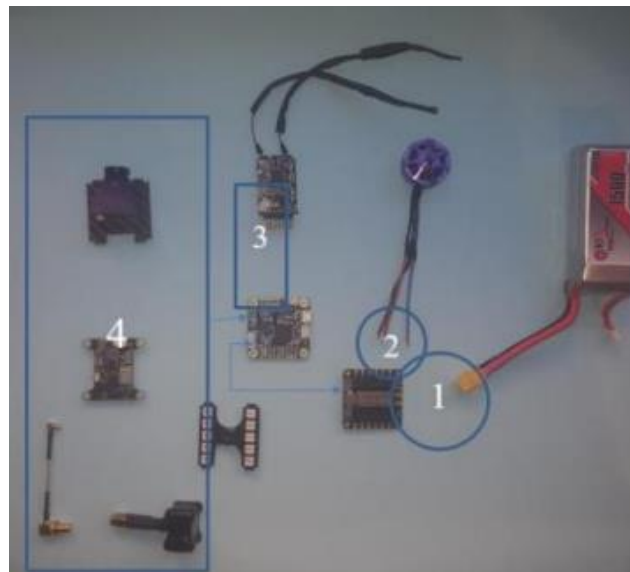


Figura 74. Ubicación piloto de los componentes

Elaborado por el investigador

- El número 1 representa el circuito de alimentación: Se alimenta la placa de los ESC mediante la batería. Esta placa se conectará a la Placa controladora mediante un puerto UART.

- El número 2 es el circuito de potencia: Se compone por los variadores y los motores.
- El número 3 es el circuito de control: Está formado por el receptor y la placa controladora y es el encargado de recibir las órdenes del mando y transmitir las al dron.
- El número 4 es el circuito FPV: se compone de la cámara, el VTX y la antena, también la placa controladora forma parte de él debido a que en ella es donde se introduce el OSD a la imagen final.

Una vez familiarizado con la ubicación y las funciones que realiza cada uno de los componentes se procede al armado eléctrico y motriz en el frame o cuerpo del dron.

Placa de control

La placa de control es uno de los componentes más importantes dentro del armado del dron, dado que trabaja como procesador y mediante él se recibe y envía señales indicadas por el usuario.

Para el armado del dron se utiliza la placa controladora MASTKSYS FLIGHT CONTROLLER F405-STD. Esta placa es de fácil programación mediante el software BETAFLIHT y su conexión con los diferentes elementos que conforman al dron es sencilla.

En la figura 75 se presenta el diagrama de conexión para la placa controladora, con cada uno de sus elementos.

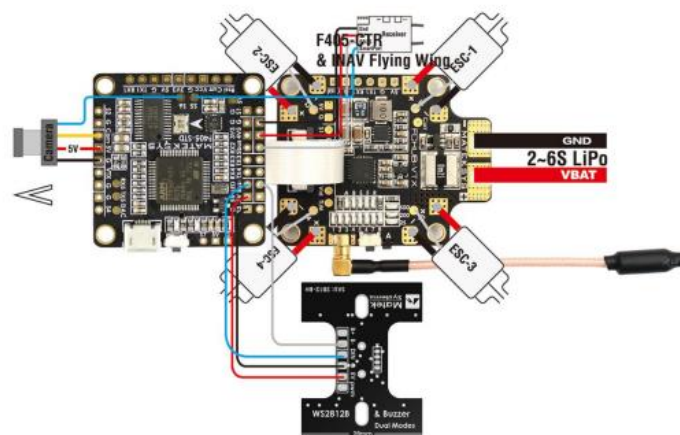


Figura 75. Diagrama de conexión de la placa de control

Elaborado por el investigador

Una vez conocidos cada uno de los pines de conexión en el diagrama, se procede a identificar los componentes y soldar respectivamente en el lugar que corresponden, figura 76.



Figura 76. Conexión de la placa controladora en el frame

Elaborado por el investigador

Variadores (ESC)

Los variadores de velocidad son los circuitos electrónicos que controlan la potencia que se entrega a los motores. Se utiliza la placa PDB-XT60 W/BEC 5V/12V, el cual es un variador 4 en 1 por su facilidad de montaje y bajo costo. En la figura 77, se presenta el diagrama de conexión para la ESC.

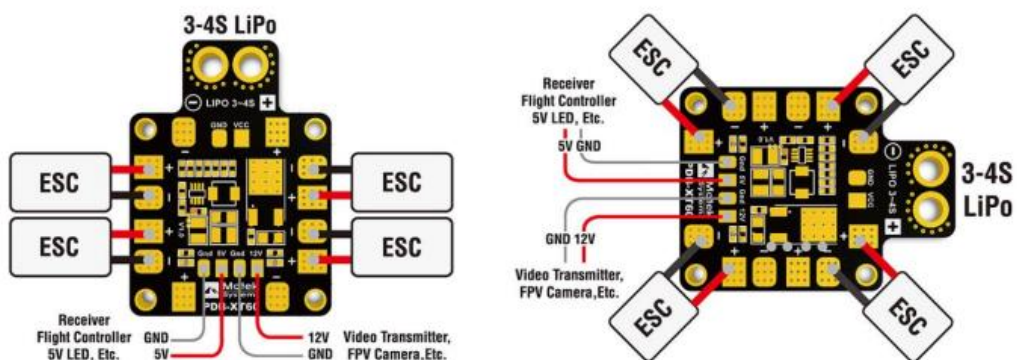


Figura 77. Diagrama de conexión ESC

Elaborado por el investigador

Revisado el diagrama de conexión del variador ESC, se procede a montar las conexiones del circuito en el cuerpo del dron, figura 78.



Figura 78. Conexión de la ESC en el frame del dron

Elaborado por el investigador

Motores

Los motores utilizados para el armado del dron son motores brushless, esto quiere decir que son motores sin escobilla. En este tipo de motores la potencia puede ser evidenciada en la etiqueta del motor (Kv) que significa la revolución por minuto y por voltio recibido, en este dron se implementan motores BR 2205 /2300 Kv.

En la figura 79, se puede observar el ensamblaje y conexión de cada uno de los motores necesarios para el dron cuadricóptero.



Figura 79. Conexión de motores en cada extremo de los brazos del dron

Elaborado por el investigador

Cámara FPV

La cámara FPV o conocidas como cámara de primera persona, es un elemento importante debido a que esta se encargará de transmitir la imagen con la cual se podrá pilotar. La conexión de esta cámara es conjunta a la del módulo VTX, dado que contiene un par de cables de alimentación y uno de datos conectados a dicho módulo, figura 80.



Figura 80. Conexión y montaje de cámara FPV

Elaborado por el investigador

VTX

El VTX es el encargado de enviar la imagen a las gafas, lo cual permite pilotar cuando se desee utilizar. Para este caso se utiliza Eachine TX801 que cuenta con 72 canales con una frecuencia de 5.8Ghz. En la figura 81, se presenta el diagrama de conexión para este módulo de transmisión.

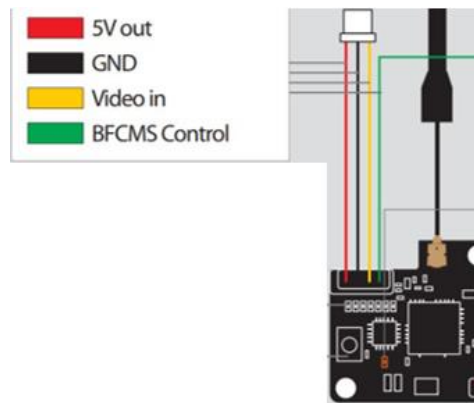


Figura 81. Diagrama de conexión de VTX 801

Elaborado por el investigador

En la figura 82, se puede observar la conexión del módulo VTX y el montaje en el frame o cuerpo del dron.



Figura 82. Conexión del módulo VTX

Elaborado por el investigador

Batería LIPO

Las baterías se clasifican por el número de celdas que tienen, lo más común son 4s (4 celdas) y 5S (5 celdas). Los motores tienen mucho que ver con el número de celdas que tenga la batería, dado que cada motor posee una potencia diferente y si se le inyecta un voltaje mayor podrían quemarse. Para los motores de este dron, lo recomendable es utilizar una LIPO de 4 celdas con 1300mAh que está dentro del rango de los Brushless. En la figura 83, se muestra ubicación de la batería en su base respectiva.



Figura 83. Ubicación de la batería en su base

Elaborado por el investigador

Emisora - Receptora y Gafas FPV

Estos dispositivos son parte externa en la implementación del dron, pero cada uno de ellos conlleva su una función específica muy relevante. La emisora receptora es un radiocontrol de la marca Fly-Sky FS i6 mediante el cual se logra la comunicación entre usuario y aeronave a una frecuencia de 2.4GHz, figura 84. Por otro lado, las gafas FPV EACHINE VR006 nos permiten visualizar la trayectoria del dron en el aire mientras ejecuta su vuelo y además en la pantalla de este se presentan datos de velocidad, altitud y posicionamiento, figura 84.



Figura 84. Emisora - Receptora y Gafas FPV

Elaborado por el investigador

Dron cuadricóptero Realaccx210Pro implementado

Una vez terminado el montaje de la estructura del dron y la conexión de cada uno de los componentes, se presenta el dron armado completamente con sus dispositivos de comunicación respectivos, figura 85.



Figura 85. Dron Realaccx210Pro completo

Elaborado por el investigador

3.2.7.3 Configuración de dron Realaccx210Pro en software Betaflight

Para la programación del dron es necesario el uso del software Betaflight, para la programación de vuelo en la placa controladora e instalación del firmware en variadores ESC.

Este software posee un sin número de configuraciones para los vuelos de un dron, en este caso se va a configurar los parámetros netamente necesarios para ejecutar la elevación de la aeronave, a continuación, se muestran los pasos desarrollados para calibración y configuración de la placa controladora MASTKSYS FLIGHT CONTROLLER F405-STD.

En primera instancia se debe conectar el dron a la laptop para visualizar la entrada con a que está conectado el dron, en este caso COM6 (Puerto del dron), figura 86. En el proceso no es recomendable realizar una actualización del dron, si el programa se lo pide, debido a que la memoria de la placa controladora quedaría bloqueada por uso de la versión reciente.

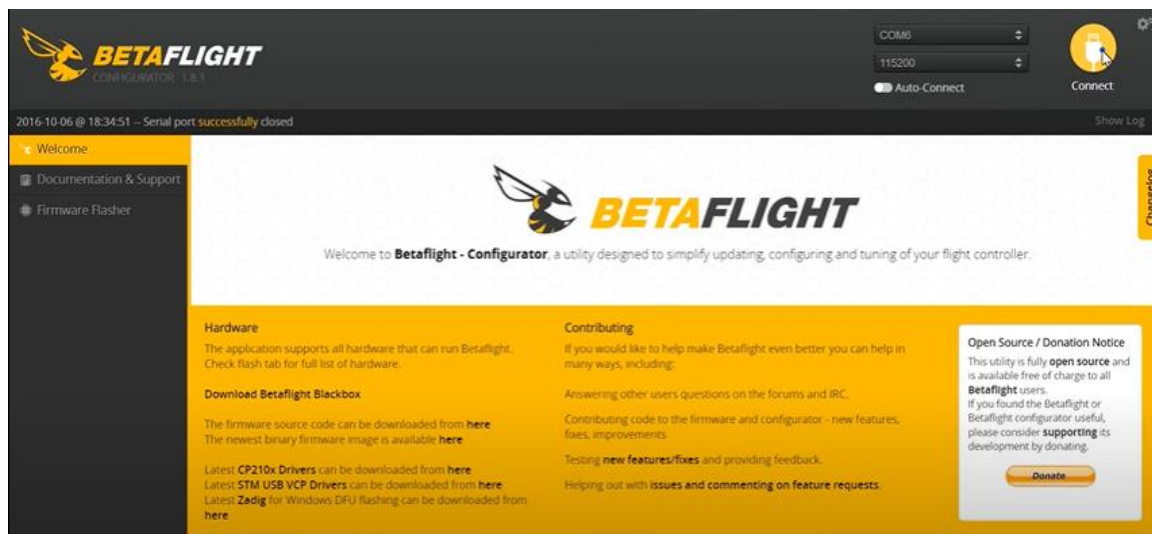


Figura 86. Puerto de conexión de DRON COM6

Elaborado por el investigador

Seguido a esto se procede a conectar al dron mediante los mandos del software, figura 87, para calibrar el dron es necesario que se encuentre sobre una base plana, debido a que el software se maneja por ángulos de nivel y estabilización.

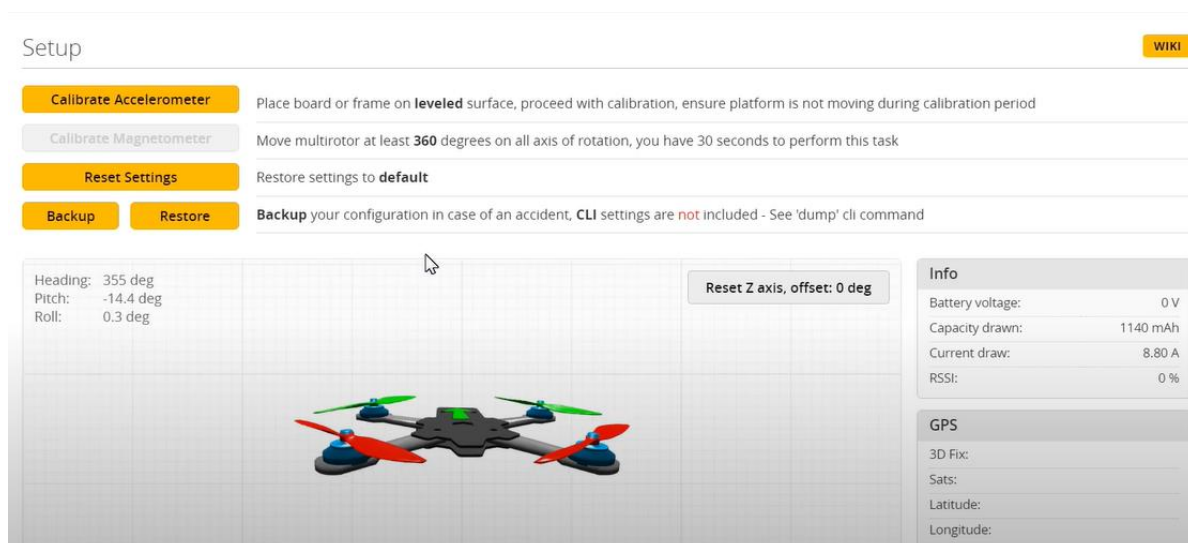


Figura 87. Reconocimiento de posición del dron en el software Betaflight

Elaborado por el investigador

Una vez que el dron haya sido reconocido por el software y se encuentre posicionado a un nivel plano, se empieza con la configuración de cada componente como puertos, modos de vuelo, motores, radiocontrol VTX.

Puertos

Se deben activar los puertos UART2, figura 88, debido a que recibe la señal que por defecto nos da para conectar el receptor (radio), además, en el puerto periférico se activó el VTX porque tiene un puerto IRT el cual hace que desde el mando se pueda cambiar el canal de gafas).

Identificador	Configuración/MSP	Rx Serie	Salida de Telemetría
USB VCP	<input checked="" type="checkbox"/> 115200 ▼	<input type="checkbox"/>	Desactivado ▼ AUTO ▼
UART1	<input type="checkbox"/> 115200 ▼	<input type="checkbox"/>	Desactivado ▼ AUTO ▼
UART2	<input type="checkbox"/> 115200 ▼	<input checked="" type="checkbox"/>	Desactivado ▼ AUTO ▼
UART3	<input type="checkbox"/> 115200 ▼	<input type="checkbox"/>	Desactivado ▼ AUTO ▼
UART4	<input type="checkbox"/> 115200 ▼	<input type="checkbox"/>	Desactivado ▼ AUTO ▼

Figura 88. Activación de puerto UART2

Elaborado por el investigador

A partir de esto se procede a la configuración de la placa definiendo la frecuencia a la que trabaja que según las características es de 4kHz, además se debe activar el acelerómetro y el barómetro.

Alineación de placa y sensores

Giro CW180: Es la posición de la placa (Giro horario).

Se debe realizar el cambio de los 4 movimientos los cuales son: Grados roll (Giro), Grados Pitch (Inclinado, se refiere a que mientras más inclinado esta corre más y mientras menos inclinado este corre menos), Grados Yaw (Giro sobre su propio eje), Giro Trotel (Subir y bajar), como se observa en la figura 89.

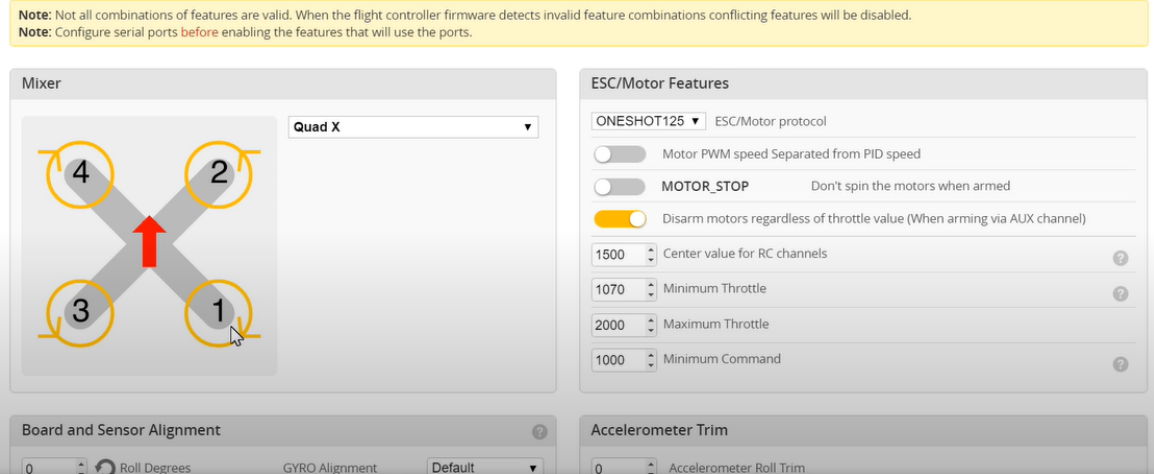


Figura 89. Alineación de placas y sensores

Elaborado por el investigador

Configuración de Baliza Dshot

Mediante esta configuración se activan los pitidos para los variadores ESC, figura 90, los cuales harán sonar a los motores cuando se pierda la señal del receptor, se active la señal y en el momento en que esté conectada la batería.

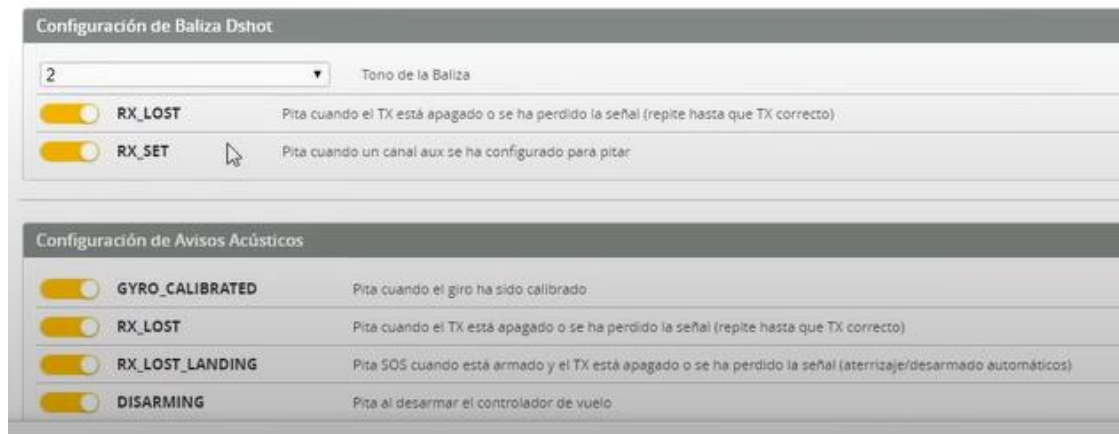


Figura 90. Configuración de Baliza Dshot

Elaborado por el investigador

Ajustar PIDs

Configurar los PIDs en un dron es esencial para lograr un vuelo estable y controlado. En primer lugar, se colocan valores iniciales predeterminados o los valores sugeridos

por el fabricante. Se realiza ajustes graduales y pequeños para no causar inestabilidad. Dentro de los PIDs es necesario configurar lo siguiente:

Proporcional (P): Se ajusta el valor P para controlar la sensibilidad del dron a las entradas del control. Un valor alto hará que el dron responda más rápido, pero demasiado alto puede causar oscilaciones.

Integral (I): El valor I se ajusta para corregir errores acumulativos en la posición del dron. En caso de que el dron empiece a desviarse en un periodo de tiempo es necesario aumentar este valor.

Derivativo (D): Se ajusta el valor D para controlar la respuesta del dron a cambios rápidos en la posición. Si existe la presencia de oscilaciones excesivas durante el vuelo del UAV es recomendable aumentar estos valores.

Estos parámetros deben ser configurados para cada uno de los movimientos del dron (ROLL, PITCH y YAW), en la figura 91 se observan los valores determinados para un vuelo estable y controlado de la aeronave.

	Proportional	Integral	Derivative	D Min	Feedforward
Basic/Acro					
ROLL	46	102	42	24	86
PITCH	49	108	46	26	91
YAW	36	108	0	0	101

Note: Sliders range is restricted because you are not in expert mode. This range should be suitable for most builds and beginners.

	Low	Default	High
Master Multiplier:	1.2		
PD Balance:	0.9		
P and D Gain:	1		
Stick Response Gain:	0.8		

PID Controller Settings

- Feedforward transition: 0
- Acro Trainer Angle Limit: 20
- Throttle Boost: 5
- Absolute Control: 0
- I Term Rotation:
- Vbat PID Compensation:
- Integrated Yaw:
- I Term Relax:
 - RP Axis
 - Setpoint Type
 - 20 Cutoff
- D Min:
 - 30 Gain
 - 0 Advance
- Anti Gravity:
 - Smooth Mode
 - 10 Gain

Figura 91. Ajuste de PID Betaflight

Elaborado por el investigador

Receptor

El receptor que se seleccionara es a través del puerto UART el cual es el RX y el TX que se tiene en las placas y el lenguaje es el Airbus. Por ende, dentro de la configuración del receptor lo que se visualizará es el movimiento que se tiene de los distintos lados del mando de control, figura 92.

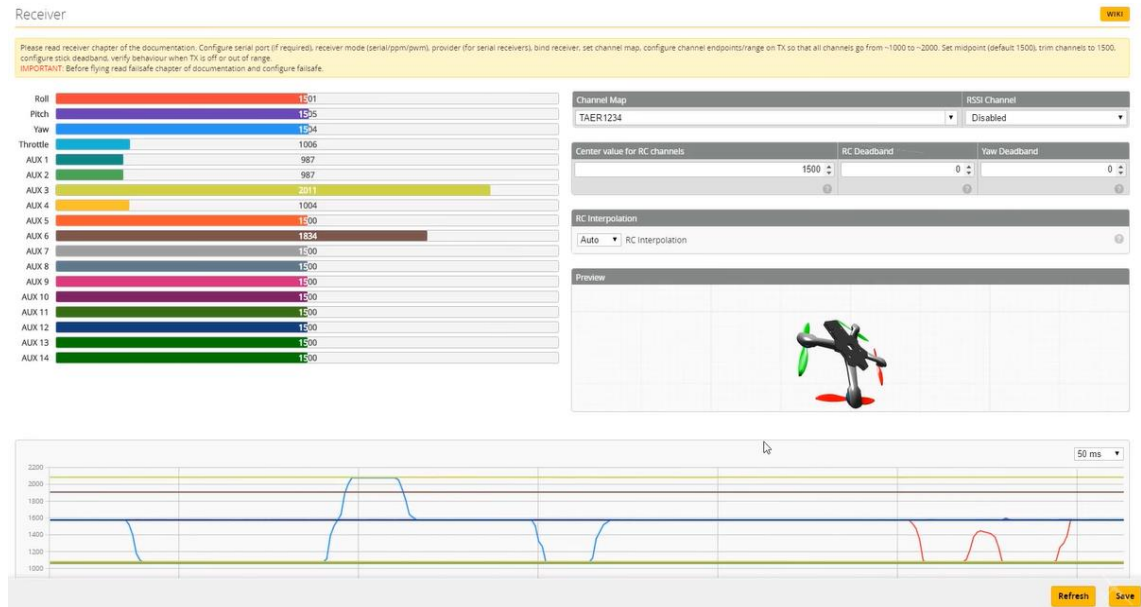


Figura 92. Receptor Betaflight

Elaborado por el investigador

Modos de vuelo

Se configuran los modos de vuelo necesarios para ejecutar las acciones del mando, mediante los canales auxiliares se configura el canal 6 y se adoptan 2 modos de vuelo debido al switch de 2 posiciones presente en el radiocontrol. Los modos de vuelo a utilizar son Stabilize y Loiter que fueron seleccionados con respecto a la funcionalidad del dron, en este caso una inspección de superficie terrena, figura 93.

El modo de vuelo Stabilize hace referencia a estabilizar, trabaja en conjunto con los giroscopios y acelerómetros para tener una auto estabilización luego de un movimiento brusco ayudando al dron a conservar el horizonte.

El modo de vuelo Loiter significa mantener que en este caso sería la altitud, es un vuelo utilizado para fotogrametría debido a que mantiene la altura a través del barómetro tomando como referencia el punto en el que despegó.

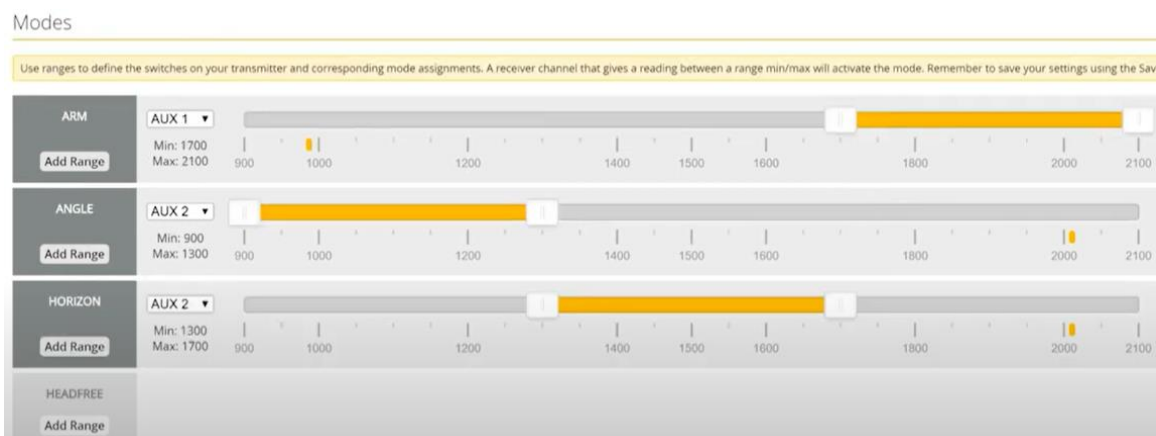


Figura 93. Configuración de Modos de vuelo Betaflight

Elaborado por el investigador

Motores

En esta pestaña se configura la funcionalidad de los motores y se visualiza el orden de posición en cada uno de sus brazos y sus giros en el sentido correcto, figura 94.

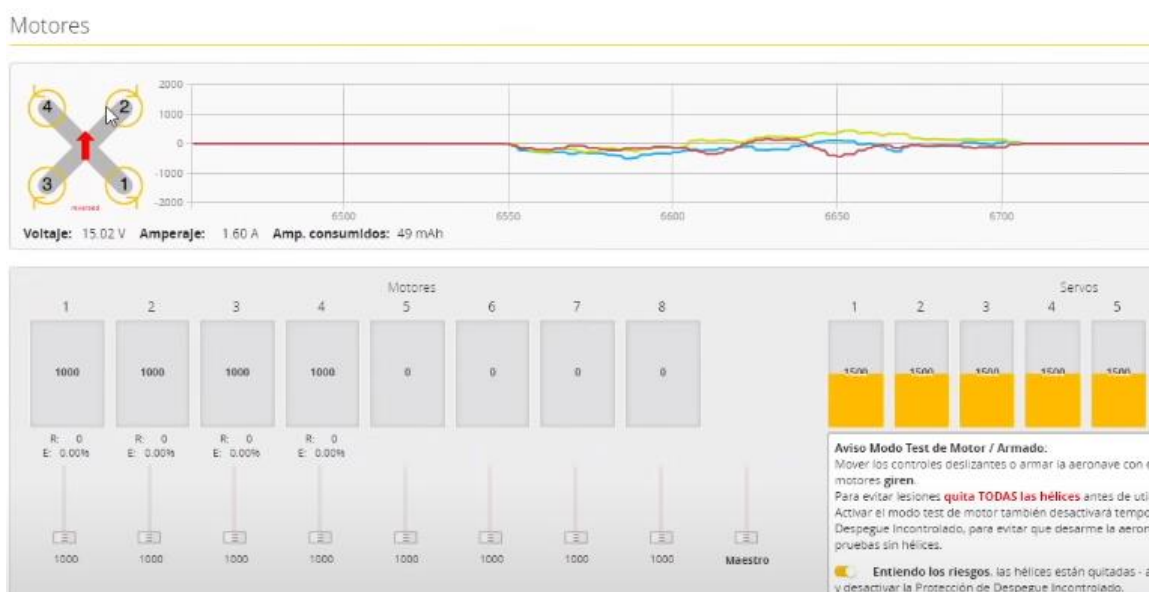


Figura 94. Motores Betaflight

Elaborado por el investigador

OSD

Por último, se configura el OSD (On Screen Display) que viene integrado en la placa controladora, su configuración nos permite observar qué es lo que se proyecta en las gafas FPV (Nivel de batería, Nombre del dron, Número de vuelos), figura 95.

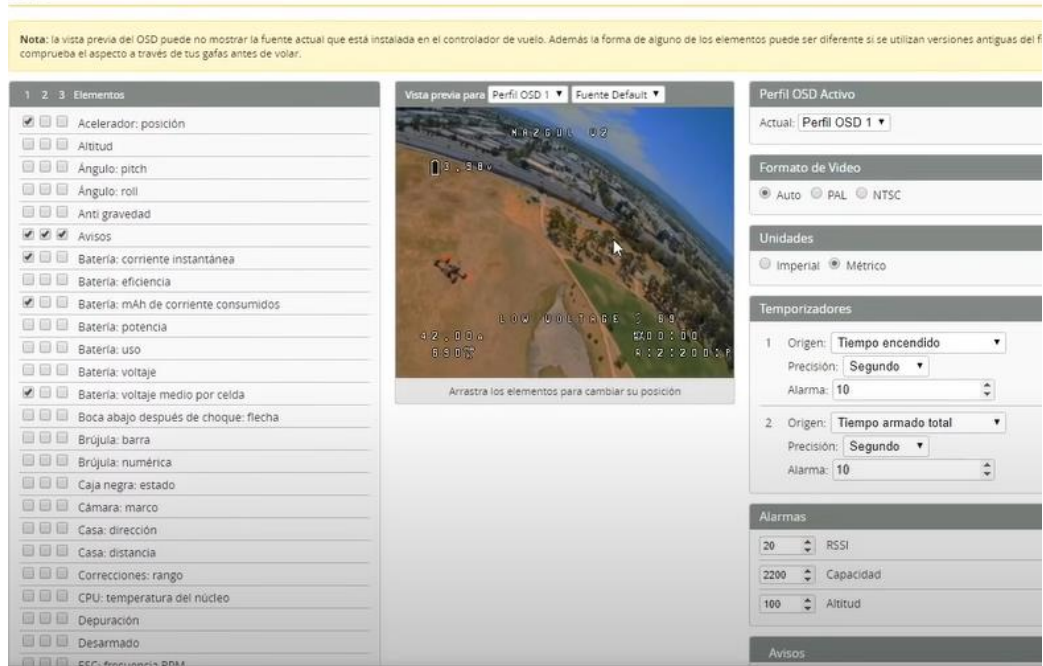


Figura 95. OSD Betaflight

Elaborado por el investigador

Una vez terminada la configuración de la placa de control y los modos de vuelos respectivos para el uso del dron, es necesario tener en cuenta cada una de sus características técnicas presentadas en la tabla 10, entre las más considerables su peso neto, peso de soporte, alcance de transmisión, duración de la batería con y sin peso adicional. Estas especificaciones técnicas nos ayudaran a elegir de mejor manera la secuencia y duración de vuelo para las inspecciones que se realizaran en los cultivos.

3.2.7.4 Simulación de modelo cinemático de un dron mediante coordenadas en software Matlab para ejecución de planes de vuelo

La simulación de un modelo cinemático para drones nos plantea la planificación de vuelos que el dron va a realizar en una inspección de superficie cultivada, representando la velocidad lineal angular con la que el dron se traslada dentro del área de inspección. En la figura 96, el dron está identificado como un punto simple dentro del plano cartesiano R que representa el marco inercial mediante las medidas X_q , Y_q y Z_q .

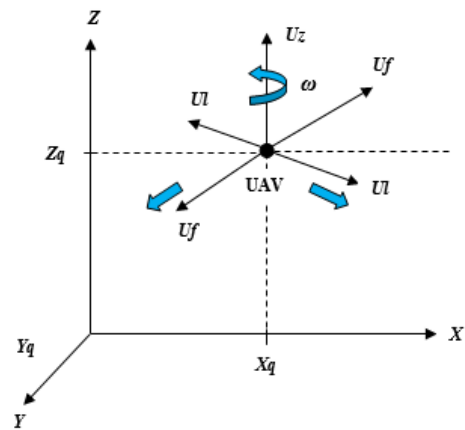


Figura 96. Marco Inercial de UAV

Elaborado por el investigador

En donde se especifica:

U_l = desplazamiento frontal del UAV

U_f = desplazamiento lateral del UAV

U_z = velocidad de elevación del UAV

w = velocidad angular del UAV

Estos son puntos claves que ayudan a la orientación del UAV en el aire.

Conocida la posición y desplazamiento de un UAV dentro de los planos cartesianos, se procede a realizar la simulación de planes de vuelo mediante el uso de coordenadas.

Para iniciar con la programación de los planes de vuelo en el entorno de Matlab, es necesario diseñar un UAV dentro del software AUTOCAD. Se diseñó un dron tipo copter de 4 alas (cuadricóptero) en una escala de 1:3, figura 97.

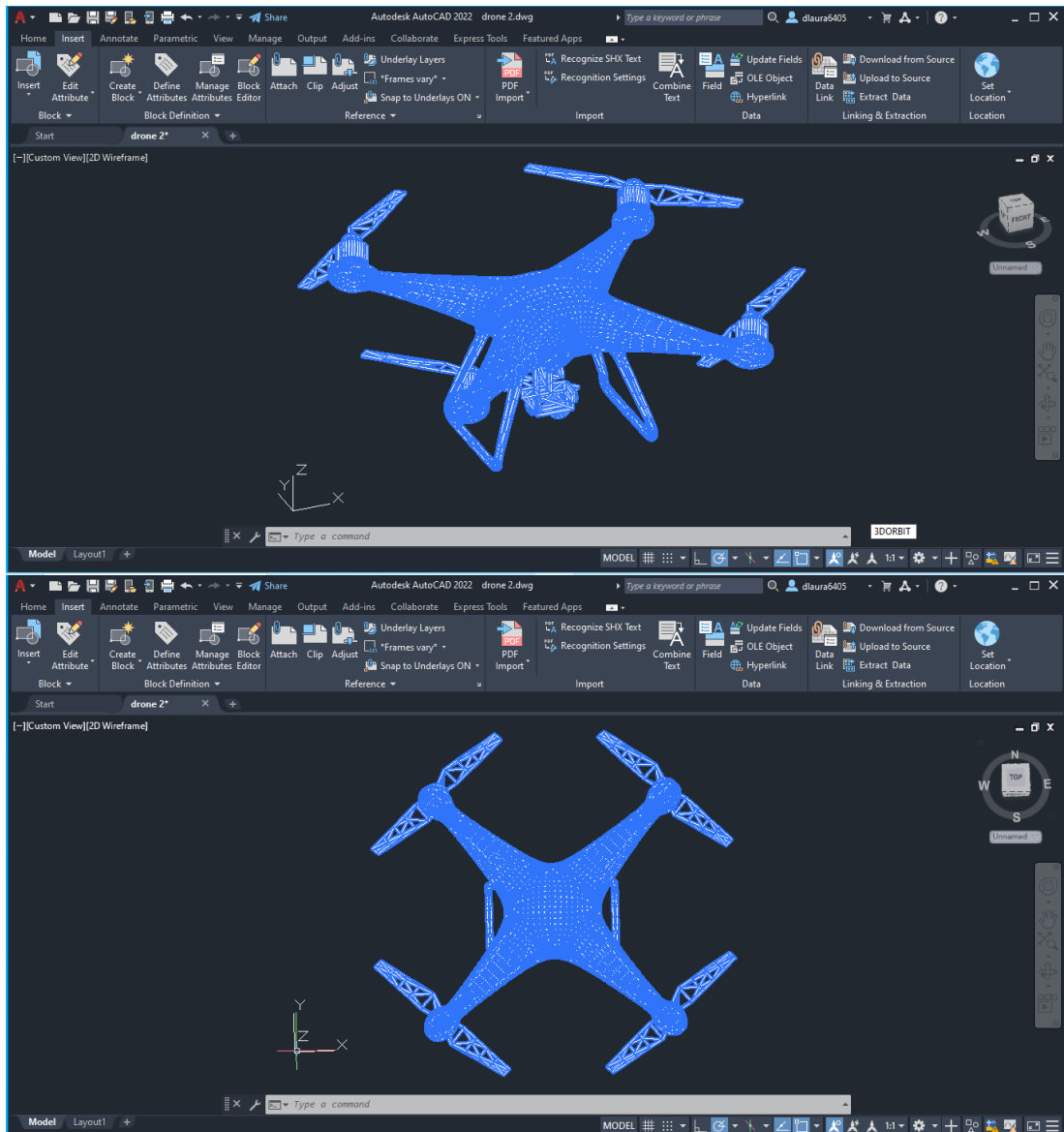


Figura 97. Diseño 3D de un dron cuadricóptero en AUTOCAD

Elaborado por el investigador

Al terminar el diseño se debe guardar el archivo en formato STL, debido a que se comprime su tamaño y el proceso del modelo es mucho más rápido en el software Matlab, debido a que se almacena solamente su superficie sin color ni textura.

En el proceso de exportación del archivo el diseño suele sufrir cambios de posición, por ende, para calibrar correctamente el parámetro de centro de masa sobre el plano que se va a trabajar se utilizó el software FUSION360, que permite centrar con facilidad el diseño, figura 98.

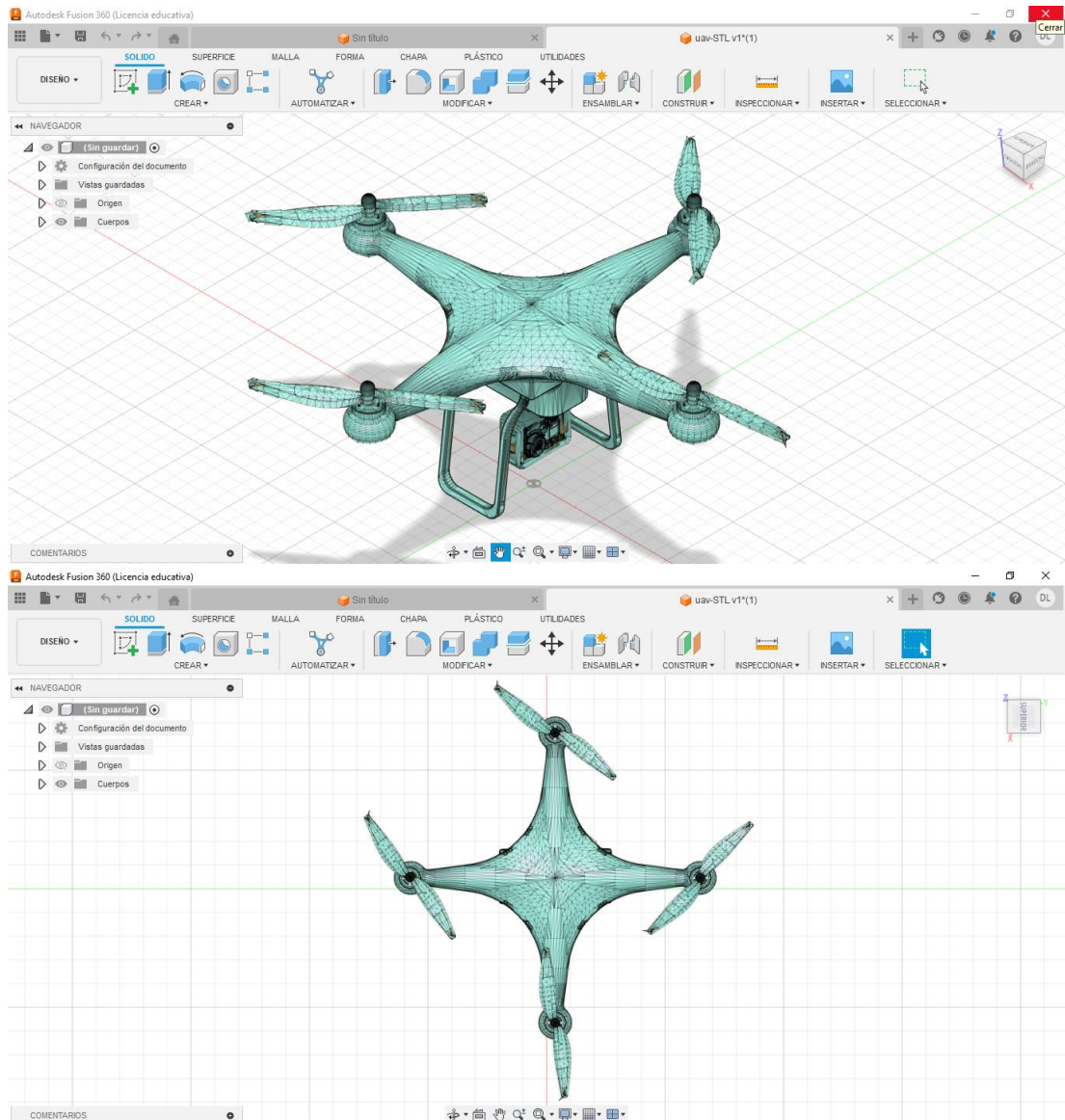


Figura 98. Diseño posicionado en software FUSION 360

Elaborado por el investigador

En el software Matlab se desarrolla el código de programación mediante el cual se ejecuta la orden de importar el diseño del UAV previamente centrado y en formato STL para la demostración de posicionamiento y trayectoria. En la codificación se crea funciones y una variable global UAV, de esta manera llamar al archivo con su nombre y en extensión STL, figura 99.

```

1 function uavRobot
2
3 global uav
4
5 parte1 = stlread('uav-STL v1.stl');
6
7 uav.parte1Vertices = parte1.vertices';
8 uav.parte1Faces = parte1.faces;
9
10 end

```

Figura 99. Declaración de variables

Elaborado por el investigador

Se crea una nueva ventana de comandos bajo el nombre de uavPlot.m, figura 100, en donde se ejecuta funciones para los movimientos de rotación y traslación del UAV todas estas en los ejes x, y, z.

```

1 function uavGraph = uavPlot(x1,y1,z1,phi,theta,psi,scale)
2
3 global uav
4 % Matrices de rotacion
5 Rz=[cos(psi), -sin(psi) 0; sin(psi) cos(psi) 0; 0 0 1]; %matriz rotación eje z
6 Ry=[cos(theta), 0, sin(theta); 0, 1, 0;-sin(theta), 0, cos(theta)]; %matriz rotación eje y
7 Rx=[1, 0, 0;0, cos(phi), -sin(phi);0, sin(phi) cos(phi)]; %matriz rotación eje x
8
9 robotPatch = Rz*Ry*Rx*uav.parte1Vertices;
10
11 robotPatch(1,:) = robotPatch(1,)*scale+x1;
12 robotPatch(2,:) = robotPatch(2,)*scale+y1;
13 robotPatch(3,:) = robotPatch(3,)*scale+z1;
14
15 uavGraph(1) = patch('Faces',uav.parte1Faces,'Vertices',robotPatch,'FaceColor','b','EdgeColor','none');
16
17 end

```

Figura 100. Declaración de funciones para los 3 ejes de coordenadas

Elaborado por el investigador

Una vez declaradas las funciones necesarias e ingresados los puntos de coordenadas en los 3 ejes de manera correcta, se ejecuta el código para importar el diseño del UAV dentro de la ventana grafica de Matlab, figura 101.

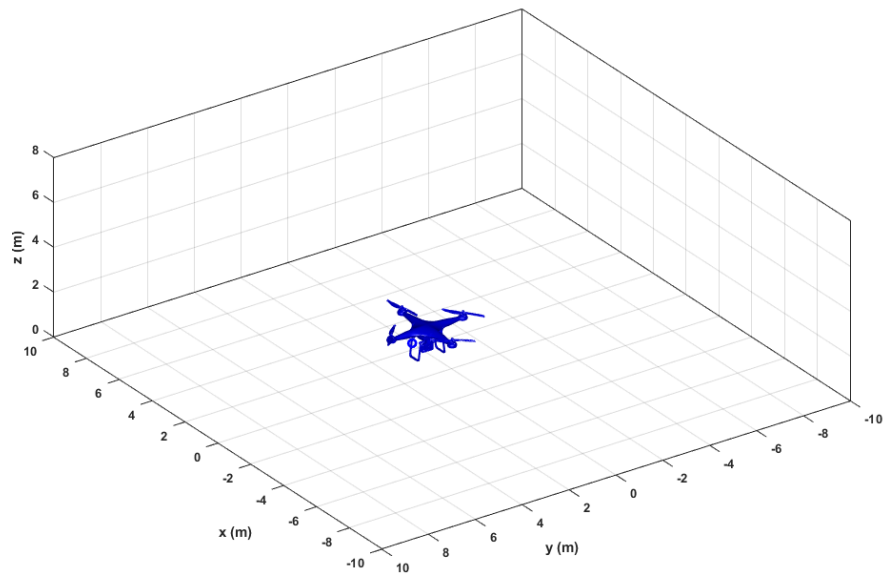


Figura 101. Posicionamiento del UAV dentro de un plano 3D en Matlab

Elaborado por el investigador

Simulación de control de posición para vuelos preprogramados

El control de posición permite ubicar el UAV en un punto de coordenadas (x,y,z) y con una orientación deseada bajo sus tres ángulos de inclinación que son el roll, pitch, yaw.

Para la codificación de este control de posición se define el tiempo de muestreo $t_s=0.1$ segundos, el vector tiempo desde cero hasta un tiempo de simulación final de 10 segundos. Se define la posición inicial del UAV en x,y,z y el ángulo de orientación inicialmente en cero y determinar las coordenadas de destino para el trayecto del UAV, figura 102.

```

Editor - C:\Users\User\Desktop\Programa matrices de vuelo\Control
uavRobot.m x uavPlot.m x Control_de_posicion.m x
4
5 % 1 TIEMPO
6 tf = 10;
7 ts = 0.1;
8 t = 0:ts:tf;
9 N = length(t);
10
11 % 2 CONDICIONES INICIALES |
12 % Centro de masa
13 hx = zeros(1,N+1);
14 hy = zeros(1,N+1);
15 hz = zeros(1,N+1);
16 yaw = zeros(1,N+1);
17
18 hx(1) = 1.5;
19 hy(1) = 0;
20 hz(1) = 0;
21 yaw(1) = 0;
22
23 % 3 POSICION DESEADA
24 hxd = -10;
25 hyd = 5;
26 hzd = 8;
27 yawd = 90 * (pi/180) ;
28

```

} Coordenadas de destino

Figura 102. Declaración de condiciones iniciales y posición final

Elaborado por el investigador

Mediante una matriz jacobiana se relacionan las posiciones del punto de control con las posiciones del UAV, utilizando una constante k que permite definir una matriz diagonal de ganancias. Seguido a esto, se realizan las configuraciones necesarias en la simulación como orientación de la figura, escala, velocidad de simulación, tamaño y etiquetado de los ejes, figura 103.

```

Editor - C:\Users\User\Desktop\Tesis Teledeteccion\Programa matrices de vuelo\Control_de_posicion.m
uavRobot.m x uavPlot.m x Control_de_posicion.m x Control_de_seguimiento_de_trayectoria.m
86 scene = figure;
87 tam=get(0,'ScreenSize');
88 set(scene,'position',tam); %Figura de posición y tamaño en la pantalla
89 set(scene,'Color','white'); % Color del fondo de la escena
90 set(gca,'FontWeight','bold');% Negrilla en los ejes y etiquetas
91 axis equal;%Establecer relaciones de aspecto de eje
92 axis([-10 10 -10 10 0 8])
93 view([-125 30]);%figura de orientación
94 grid on;% Mostrar líneas de cuadrícula de ejes
95 box on;% Mostrar contorno de ejes
96 xlabel('x (m)'),ylabel('y (m)'),zlabel('z (m)')
97 camlight('headlight');
98 scale = 0.05;
99 uavRobot;
100 M1 = uavPlot(hx(1),hy(1),hz(1),0,0,yaw(1),scale);
101 hold on
102 M2 = plot3(hx(1),hy(1),hz(1),'r','lineWidth',2);
103 M3 = plot3(hxd,hyd,hzd,'Ob','lineWidth',2);
104 step = 1;
105 for k=1:step:N
106 delete(M1)
107 delete(M2)
108 M1 = uavPlot(hx(k),hy(k),hz(k),0,0,yaw(k),scale);
109 M2 = plot3(hx(1:k),hy(1:k),hz(1:k),'r','lineWidth',2);
110 pause(ts)
111 end

```

Figura 103. Configuración de frame grafico

Elaborado por el investigador

Se ejecuta el programa según distintos puntos de destino y ángulos de inclinación configurados manualmente:

En la figura 104, se observa un control de posición obtenido con los parámetros establecidos.

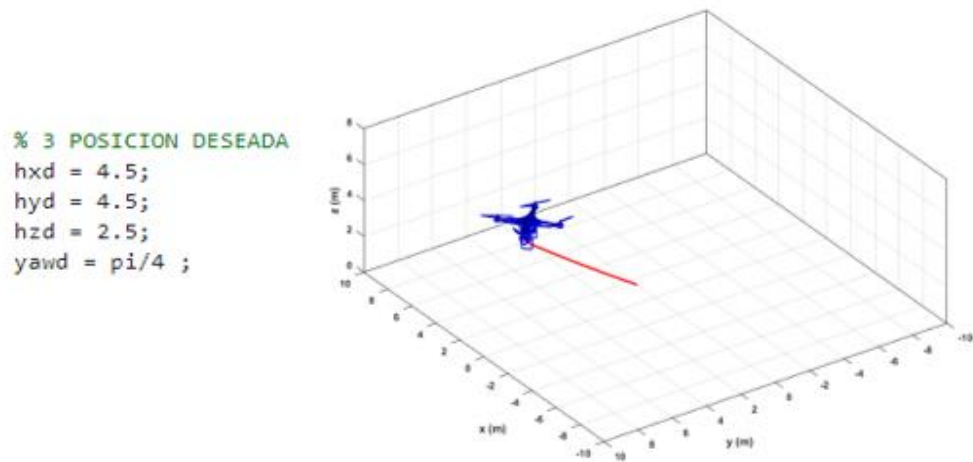


Figura 104. Trayectoria del UAV con un ángulo de 45°

Elaborado por el investigador

En la figura 105, se muestra la trayectoria del UAV con los parámetros definidos en el código de programación, así como punto de destino y ángulo de inclinación.

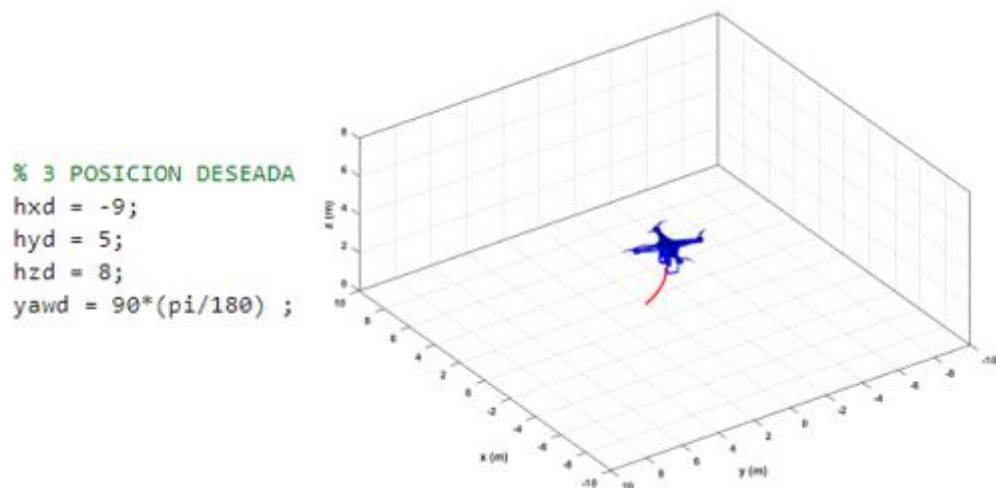


Figura 105. Trayectoria del UAV con un ángulo de 90°

Elaborado por el investigador

Control de seguimiento y trayectoria

En este panel de programación se simulan los planes de vuelo libre, octogonal, circunferencial y helicoidal, para los cuales se tiene un tiempo de muestreo de 60 segundos en pasos de 0.1, se define el punto de control inicial y control del ángulo yaw que permite al UAV elevarse en un ángulo de 90° , para la ley de control es necesario utilizar una matriz jacobiana, figura 106.

```
Editor - C:\Users\User\Desktop\Programa matrices de vuelo\Control_de_seguimiento_de_trayectoria.m
uavRobot.m x uavPlot.m x Control_de_posicion.m x Control_de_seguimiento_de_trayectoria.m
102
103 % Jacobiano
104 J = [cos(yaw(k)) -sin(yaw(k)) 0 0;...
105       sin(yaw(k))  cos(yaw(k)) 0 0;...
106       0             0         1 0;...
107       0             0         0 1];
108
109 % Parametros de control
110 K = [1 0 0 0;...
111      0 1 0 0;...
112      0 0 1 0;...
113      0 0 0 1];
114
115 % Velocidad deseada
116 hdp = [hxdp(k);hydp(k);hzdp(k);yawdp(k)];
117
118 % Ley de control
119 qpRef = pinv(J)*(hdp+K*he);
120
121 % Aplicar al robot
122 uf(k) = qpRef(1);
123 ul(k) = qpRef(2);
124 uz(k) = qpRef(3);
125 w(k) = qpRef(4);
126
127 yaw(k+1) = yaw(k)+ts*w(k);
128 % Modelo cinematico diferencial en el centro de masa
129 x1p = uf(k)*cos(yaw(k+1))-ul(k)*sin(yaw(k+1));
```

Figura 106. Matriz jacobiana para control de trayectoria del UAV

Elaborado por el investigador

La trayectoria planteada en el vuelo del UAV de forma libre se visualiza bajo los parámetros establecidos en la figura 107.

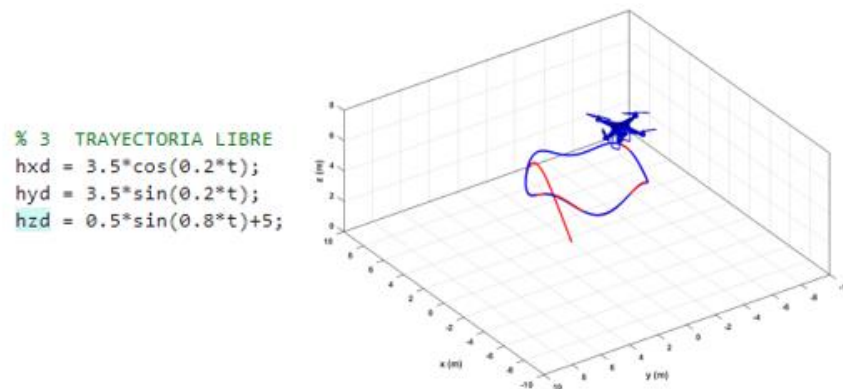


Figura 107. Trayectoria Libre

Elaborado por el investigador

Para que se ejecute la trayectoria en forma octogonal se obedecen a los parámetros que se encuentran en la figura 108.

```
% Trayectoria forma Octogonal  
hxd = 3.5*cos(0.2*t);  
hyd = 3.5*sin(0.4*t);  
hzd = 5*ones(1,N);
```

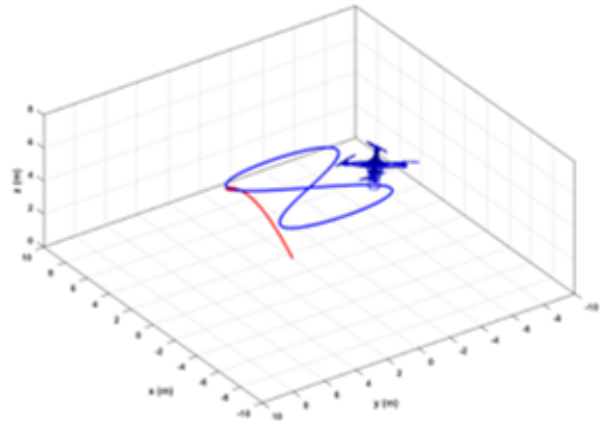


Figura 108. Trayectoria en forma octogonal

Elaborado por el investigador

En la figura 109, se muestra la trayectoria del UAV en forma circunferencial bajo los parámetros establecidos.

```
% Trayectoria forma Circunferencial  
hxd = 3.5*cos(0.2*t);  
hyd = 3.5*sin(0.2*t);  
hzd = 5*ones(1,N);
```

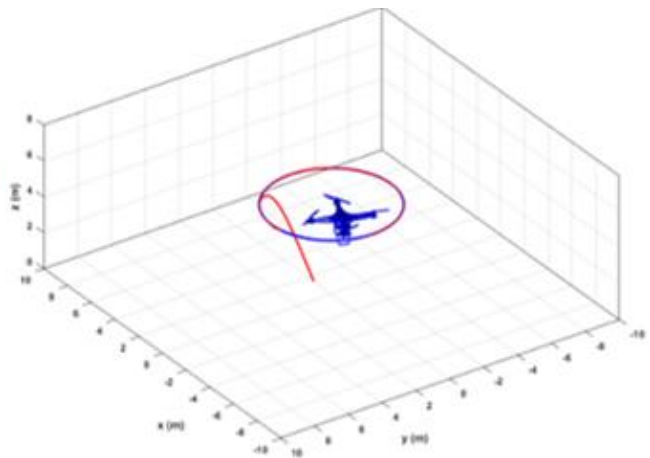


Figura 109. Trayectoria en forma circunferencial

Elaborado por el investigador

La trayectoria del UAV en forma helicoidal, se observa en la figura 110 con los parámetros necesarios para su ejecución.


```

% Trayectoria forma Helicoidal
hxd = 3.5*cos(0.2*t);
hyd = 3.5*sin(0.2*t);
hzd = 5*ones(1,N);
%
hxd = 3.5*cos(0.2*t);
hyd = 3.5*sin(0.2*t);
hzd = 0.1*t;

```

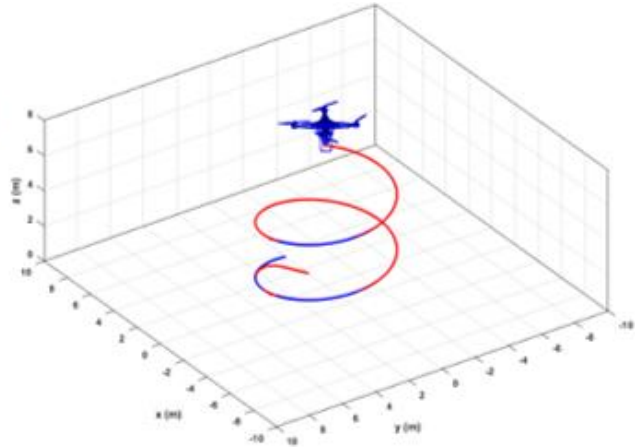


Figura 110. Trayectoria en forma helicoidal

Elaborado por el investigador

Simulación de control PID en las trayectorias del dron

Para la simulación del control PID de la trayectoria del dron se requiere definir las ecuaciones de movimiento. Solamente se moverá en los planos z e y, y se tendrá el ángulo ϕ . Además, se define la masa que es de 0,5 Kg y el momento de inercia que es igual a 0,1 unidades.

$$\ddot{z} = g - \frac{u_1}{m} \cdot \cos(\phi)$$

$$\ddot{y} = \frac{u_1}{m} \cdot \sin(\phi)$$

$$\ddot{\phi} = \frac{u_2}{I_{xx}}$$

$$m = 0,5Kg$$

$$I_{xx} = 0,1 \text{ unidades}$$

En donde:

u_1 = sumatoria de todas las fuerzas existentes

m = masa del cuadricoptero

$\ddot{\phi}$ = segunda derivada del angulo de rotacion en el eje x

I_{xx} = momento de inercia

\ddot{z} = Segunda derivada del desplazamiento lineal en el eje Z

\ddot{y} = Segunda derivada del desplazamiento lineal en el eje Y

ϕ = Angulo de rotacion en el eje X

g = Constante de aceleracion de gravedad

Trayectoria de referencia

Como se observa en la figura 111, la trayectoria de referencia del dron está compuesta por dos componentes sinusoidales: En la dirección vertical (z), el dron se elevará y descenderá con una amplitud de 5 unidades desde su punto de referencia, siguiendo una función sinusoidal en función del tiempo. En la dirección horizontal (y), el dron se moverá hacia los lados con una amplitud de 0.5 unidades desde su punto de referencia, también siguiendo una función sinusoidal más lenta en función del tiempo. Esta combinación de movimientos oscilatorios define la trayectoria de referencia que se espera que el dron siga en la simulación.

$$z = 5 \cdot \sin(t) \quad y = 0.5 \cdot \sin(0.5t)$$

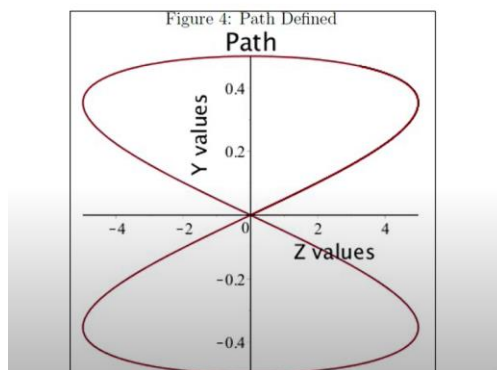


Figura 111. Trayectoria de referencia

Elaborado por el investigador

Implementación en Simulink

La simulación de control PID en las trayectorias de un dron utilizando Simulink es una forma eficaz de evaluar y afinar el rendimiento del controlador en un entorno visual y gráfico. En la figura 112 se observa el diagrama de bloques implementado en el software. Tomando en cuenta los parámetros establecidos previamente, para la primera variable de control se tiene una segunda derivada de la función Z usando un zum de polaridad para invertir los signos de la ganancia y gravedad que a su vez

mediante una doble integral se obtiene el desplazamiento de en el eje Z. En la segunda derivada de control se presenta el momento de inercia que mediante una integración doble conectada en cascada es posible obtener un bloque de salida con el desplazamiento en el eje Y. Este subsistema controla los desplazamientos del UAV en los planos Y y Z a través de las fuerzas existentes en el sistema.

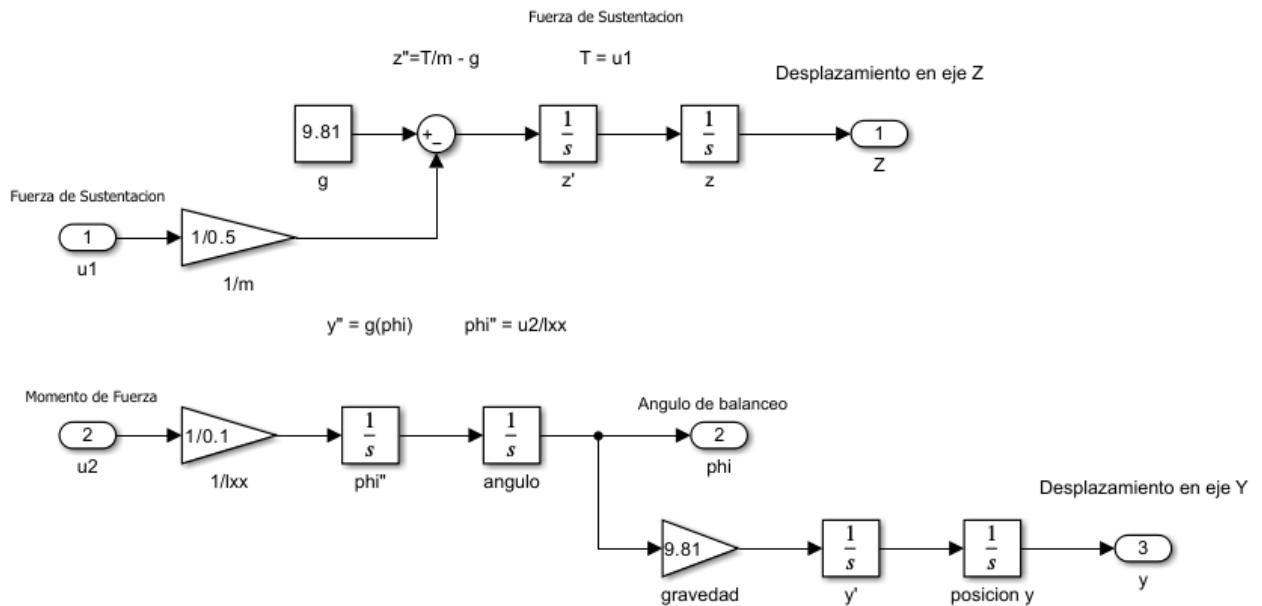


Figura 112. Modelación del subsistema

Elaborado por el investigador

Tras el subsistema desarrollado se diseñan bloques que formen las trayectorias deseadas para el dron, utilizando funciones matemáticas, generadores de señales e incluso cargando trayectorias predefinidas. En la figura 113 se visualiza el control PID diseñado, se comienza con un zum de polaridad para que el error sea la diferencia entre la entrada Z y la salida Z. Con el uso de una onda sinusoidal de amplitud 5 y frecuencia 1 en el eje Z se evidencia el control PID para el error y la salida Z. Para el desplazamiento en el eje Y es necesaria una conexión en cascada entre la onda sinusoidal de ingreso y el control PID. La salida de Y es utilizada como retroalimentación ajustar el valor del controlador. Este control es lineal debido a que no existe presencia de acción integral, por ende, se configura cada parámetro dentro de los controladores PID. Para estabilizar la señal obtenida es necesario un bloque de saturación en donde se observa la salida de Phi y su alcance definido.

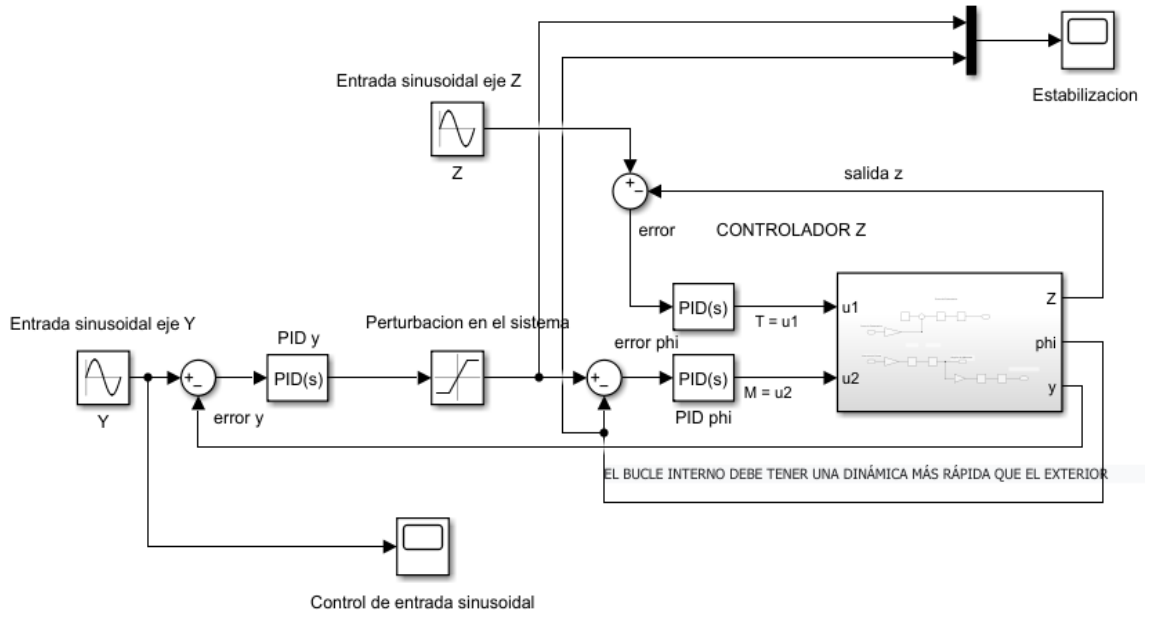


Figura 113. Control PID de la trayectoria del dron

Elaborado por el investigador

Al ejecutar el control PID diseñado en Simulink se observa como resultado la figura 114, en donde se puede apreciar que existe estabilización al alcanzar los 7 metros de altura en un tiempo de 4 segundos.

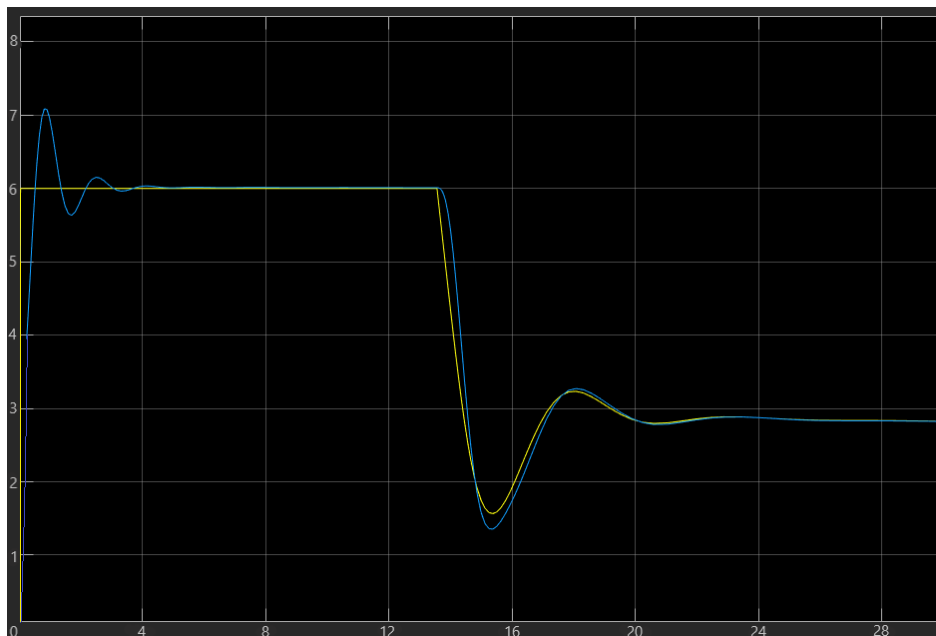


Figura 114. Estabilización

Elaborado por el investigador

Creación del controlador PID de la trayectoria del dron en Matlab

Para la creación del controlador PID de la trayectoria del dron en Matlab se debe establecer los parámetros de entrada como la masa del dron (m), el momento de inercia, las ganancias del controlador PID. Las constantes k_p , k_i y k_d del sistema son las responsables de controlar el error de posición mediante la corrección necesaria en velocidad y posición del dron. Los valores de estas constantes deben ser muy pequeños debido al error en las trayectorias de referencia, por ende, los valores están entre -1 y 1. Estos parámetros se visualizan en la figura 115.

```

% Parámetros del dron y controlador
m = 0.5; % Masa del dron en kg
Ixx = 0.1; % Momento de inercia en kg*m^2

% Parámetros de referencia de trayectoria
z_ref = @(t) 5 * sin(t);
y_ref = @(t) 0.5 * sin(0.5 * t);

% Parámetros del controlador PID
kp_z = 0.05; % Ganancia proporcional para la altura
ki_z = 0.02; % Ganancia integral para la altura
kd_z = 0.01; % Ganancia derivativa para la altura

kp_y = 0.01; % Ganancia proporcional para la posición lateral
ki_y = 0.02; % Ganancia integral para la posición lateral
kd_y = 0.05; % Ganancia derivativa para la posición lateral

```

Figura 115. Parámetros de entrada para crear controlador PID en Matlab

Elaborado por el investigador

Iniciar variables

En la figura 116 se muestra el código para establecer un intervalo de tiempo total de 20 segundos, define un tamaño de paso de tiempo de 0.01 segundos y crea un vector de tiempo t que va desde 0 hasta 20 con pasos de 0.01. Luego, calcula la cantidad de elementos en ese vector, lo que te dará el número total de pasos de tiempo en la simulación. Los vectores z , y , $u1$ y $u2$ se utilizan para almacenar valores de variables relevantes durante la simulación, mientras que t se utiliza para representar el vector de tiempo discreto que guiará la simulación.

```

% Inicialización de variables
t_total = 20; % Tiempo total de simulación en segundos
dt = 0.01; % Paso de tiempo en segundos
t = 0:dt:t_total;
N = length(t);

z = zeros(N, 1); % Altura actual del dron
y = zeros(N, 1); % Posición lateral actual del dron
u1 = zeros(N, 1); % Control de empuje
u2 = zeros(N, 1); % Control de ángulo

z(1) = 0; % Inicializar altura del dron
y(1) = 0; % Inicializar posición lateral del dron

```

Figura 116. Inicialización de variables

Elaborado por el investigador

En las líneas de código que se muestra en la figura 117 se inicializan variables que se utilizarán para llevar un seguimiento del error y la integral del error en las direcciones z (vertical) y, y (horizontal) en el proceso de control en MATLAB. Estas variables en controladores PID, para mantener un historial del comportamiento del sistema y permitir ajustes y correcciones precisas en la respuesta del sistema.

```

error_z_previo = 0;
integral_z = 0;

error_y_previo = 0;
integral_y = 0;

```

Figura 117. Definición de errores

Elaborado por el investigador

Proceso para simular el PID

Para la simulación del controlador PID se debe calcular los errores de altura y posición, calcular la integral y derivada de errores de altura, calcular la integral y derivada de errores de posición lateral, calcular señales de control utilizando el controlador PID y calcular las aceleraciones en los ejes z, y, y ángulo. La codificación se muestra en la figura 118.

```

% Simulación del controlador PID
for i = 1:N-1
    % Calcular errores de altura y posición lateral
    error_z = z_ref(t(i)) + z(i);
    error_y = y_ref(t(i)) + y(i);

    % Calcular integral y derivada de errores de altura
    integral_z = integral_z + error_z * dt;
    derivada_z = (error_z - error_z_previo) / dt;

    % Calcular integral y derivada de errores de posición lateral
    integral_y = integral_y + error_y * dt;
    derivada_y = (error_y - error_y_previo) / dt;

    % Calcular señales de control utilizando el controlador PID
    u1(i) = kp_z * error_z + ki_z * integral_z + kd_z * derivada_z;
    u2(i) = kp_y * error_y + ki_y * integral_y + kd_y * derivada_y;

    % Calcular aceleraciones en los ejes z, y y ang
    ang = atan2(u2(i) * Ixx, m * u1(i));
    az = u1(i) * cos(ang) / m;
    ay = u1(i) * sin(ang) / m;

```

Figura 118. Proceso para simular el PID

Elaborado por el investigador

El código que se presenta en la figura 119 representa una actualización de estados en un sistema dinámico en dos dimensiones (z y y) utilizando integración numérica. Está implementando un método de Euler para estimar cómo las variables z y y cambian en el tiempo en respuesta a las aceleraciones az y ay, respectivamente, en un intervalo de tiempo dt. Cada iteración del bucle calcula la nueva posición basada en la posición anterior y las aceleraciones.

```

    % Actualizar estados utilizando integración numérica
    z(i+1) = z(i) + az * dt;
    y(i+1) = y(i) + ay * dt;

    % Actualizar errores previos
    error_z_previo = error_z;
    error_y_previo = error_y;
end

```

Figura 119. Actualización de estados en Matlab

Elaborado por el investigador

Finalmente, en la figura 120 se muestra la codificación para visualizar los resultados. Este código se utiliza para crear una figura con un gráfico dividido en tres subtramas en una columna. Luego, personaliza las etiquetas de los ejes x e y, agrega una leyenda y establece un título para la subtrama. Es una forma de presentar y etiquetar

visualmente los datos en el gráfico de altura del dron en una simulación o análisis en MATLAB.

```

% Visualización de resultados
figure;

subplot(3, 1, 1);
plot(t, z, 'b', t, z_ref(t), 'r--');
xlabel('Tiempo (s)');
ylabel('Altura (m)');
legend('Altura actual', 'Altura de referencia');
title('Altura del Dron');

subplot(3, 1, 2);
plot(t, y, 'b', t, y_ref(t), 'r--');
xlabel('Tiempo (s)');
ylabel('Posición lateral (m)');
legend('Posición actual', 'Posición de referencia');
title('Posición Lateral del Dron');

sgtitle('Simulación de Controlador PID para Dron');

```

Figura 120. Visualización de resultados

Elaborado por el investigador

En la figura 121 se visualiza el resultado de la simulación de Controlador PID para trayectoria del dron. En el primer label se tiene la elevación y estabilización del dron a 50 metros de altura en un tiempo definido de 20 segundos, mientras que en el segundo label se observa la posición lateral que sigue el dron dentro de la trayectoria establecida que se desplaza 0.1 metros cada 2 segundos.

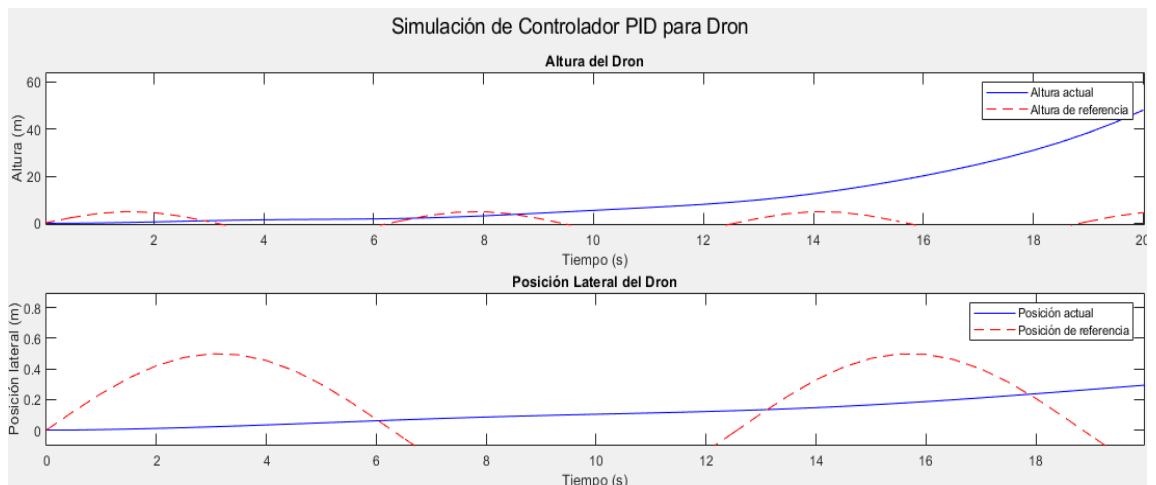


Figura 121. Resultado de la simulación de Controlador PID para trayectoria del dron

Elaborado por el investigador

- **Comportamiento de PID antes de despegar**

Los parámetros establecidos para el dron antes de despegar son de valor cero en altura, velocidad y posición, obteniendo de esta manera una gráfica constante a lo largo del valor 0 en el eje de x. Esto se debe a que el actuador dentro del dron está en reposo esperando una respuesta u acción de movimiento para iniciar su trayectoria. En la figura 122, se observa las gráficas resultantes con respecto a la acción de reposo del dron antes de su despegue, todas las señales son constantes sin ningún tipo de perturbación debido a que el dron no se encuentra en movimiento y su marco inercial no recibe ninguna respuesta de impulso.

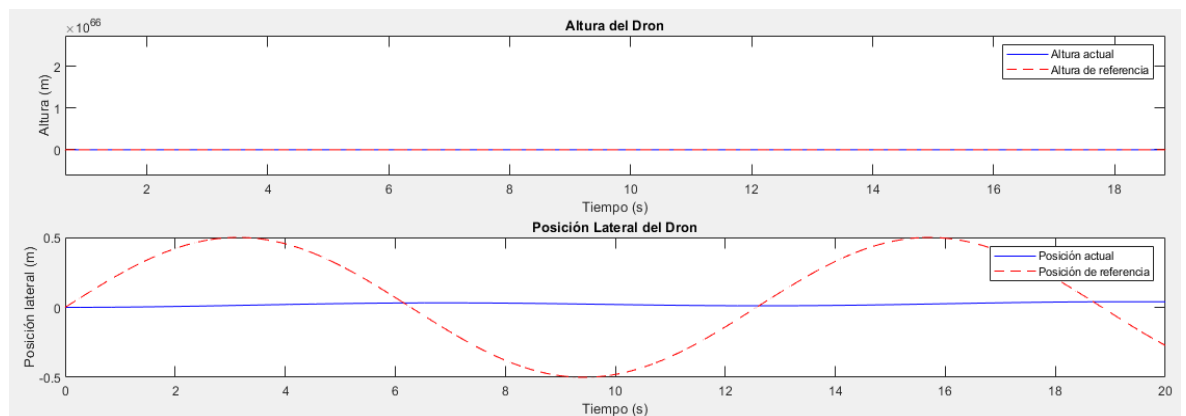


Figura 122. Comportamiento PID antes de despegue

Elaborado por el investigador

- **Comportamiento de PID en momento de despegue**

Al momento del despegue del dron los parámetros tienen cierta variación dentro de la velocidad y altura, debido a que el dron se eleva en forma vertical con un ángulo de 90 grados. En la figura 123 se observa las gráficas resultantes al momento del despegue del dron, cada una de las gráficas muestra su altura, posición lateral, empuje y ángulo de elevación. Debido a que en el instante de ascenso no se presenta ninguna perturbación, el control PID no generaliza los valores para una posible estabilización.

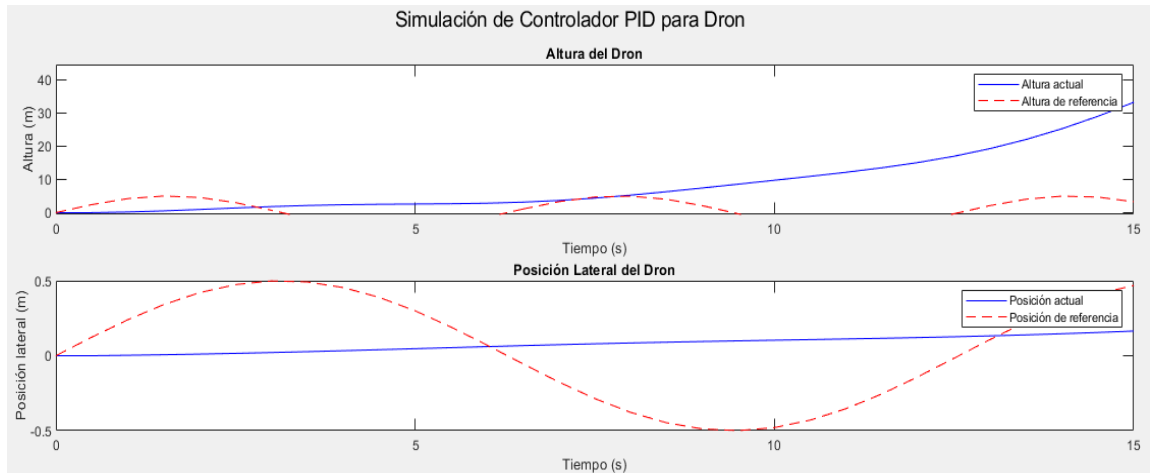


Figura 123. Comportamiento PID en ascenso de dron

Elaborado por el investigador

- **Comportamiento de PID a una altura establecida**

El dron posicionado a una altura establecida mediante una planeación de vuelo, al estar suspendido en el aire puede llegar a presentar alteraciones en su equilibrio debido a las perturbaciones existentes como es el caso del viento. Para la demostración del control PID se ha introducido una perturbación en el sistema de 20.8 rad aproximadamente unos 90 grados, en la figura 124 se observa la estabilización del sistema cada vez que se aplica esta perturbación volviendo el sistema a estabilizarse.

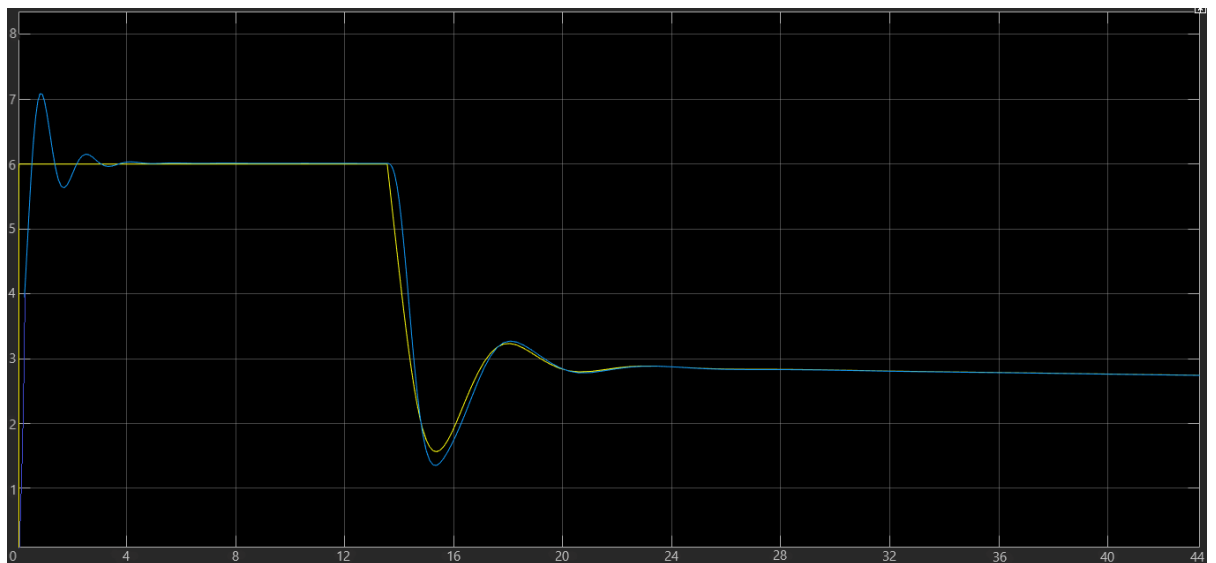


Figura 124. Comportamiento PID del dron bajo una perturbación

Elaborado por el investigador

3.2.7.5 Implementación del Módulo GPS para el dron

Se procede con la instalación del módulo AP.25E GPS/GALILEO que nos ayudara con la georreferencia del dron y además en la configuración para el armado de los planes de vuelo convirtiéndolo de esta manera en uno de tipo autónomo. Este módulo va conectado a la placa de control de vuelo y se ubica en la base superior del dron para receptor las señales satelitales, figura 125, que proporcionan la posición de los puntos seleccionados dentro de la configuración del MISSION PLANER, obligando de esta manera al dron a cumplir la ruta establecida.



Figura 125 Implementación de modulo AP.25E GPS/GALILEO en el dron

Elaborado por el investigador

3.2.7.6 Configuración de plan de vuelo para el dron autónomo en software ArduPilot

Para comenzar con la configuración del plan de vuelo programado, es necesario tener incorporado un GPS en el dron, que en este caso es el AP.25E GPS/GALILEO.

En primer lugar, se procede a abrir el programa Mission Planner de ArduPilot, figura 126, en donde se planeará cada ruta definida por el usuario mediante una serie de puntos de referencia y una dirección establecida. Se debe tomar en cuenta la conexión a internet para obtener el mapa de localización del dron en tiempo real.



Figura 126. Inicio del software Mission Planer de ArduPilot

Elaborado por el investigador

Ejecutado el programa se procede a encender el dron para conectarlo a la PC mediante cable micro USB. El puerto de conexión para el dron es COM7 y es recomendable conectarlo escuchado el segundo pitido de emergencia dado que en este instante se encuentran activos todos los elementos del sistema eléctrico y electrónico. En la figura 127 se muestra la última posición geográfica y una serie de datos recolectados en ese instante como: altitud, velocidad, etc.

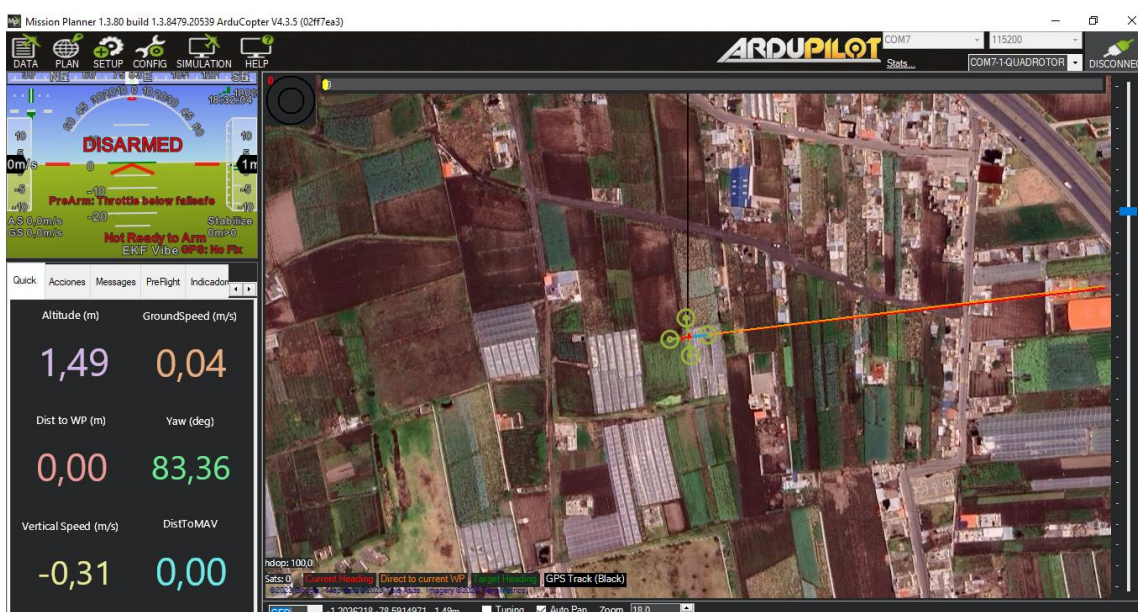


Figura 127. Ultima posición geográfica del dron

Elaborado por el investigador

Seguido de obtener la ubicación del dron, se procede a configurar el modo de vuelo automático para ejecutar los planes de vuelo programados. En la pestaña SETUP encontramos los flight modes o modos de vuelo, en donde se designará el modo automático en el estado 3 de la palanca del radiocontrol, figura 128, esto hará que el dron una vez suspendido en el aire ejecute la misión planeada.

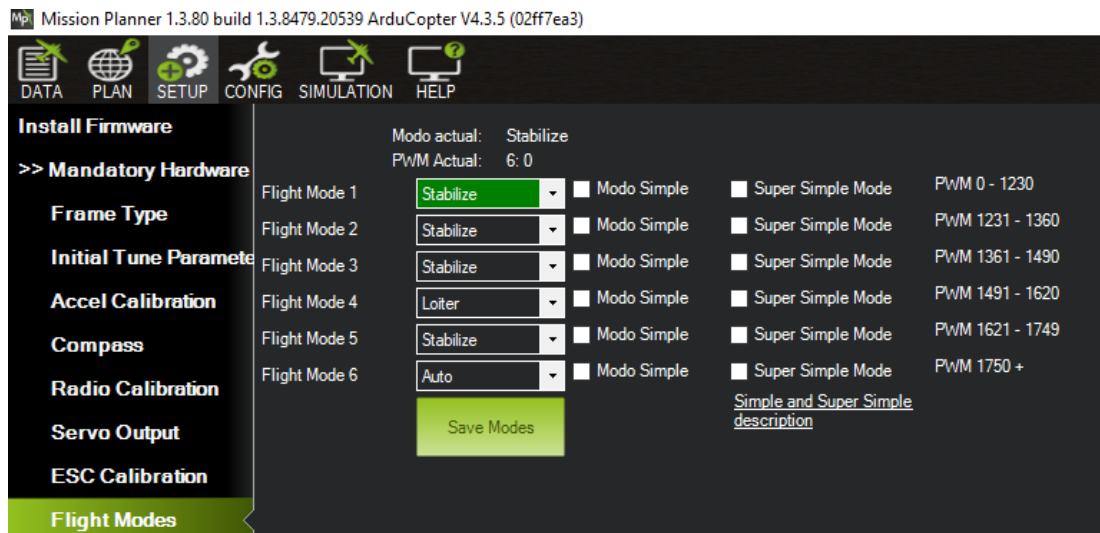


Figura 128. Modo de vuelo auto establecido

Elaborado por el investigador

En la pestaña CONFIGURACION se encuentran los parámetros de giro extendidos que se pueden modificar de acorde a las necesidades del piloto. Se configura la velocidad a la que se ejecutara cada una de las trayectorias programadas, en este caso 200 cm/s, figura 129, que son las unidades que se manejan en esta sección. Este cambio depende mucho de la potencia de los motores, debido a que ejercer una velocidad muy grande desestabilizaría al dron y no permitiría cumplir el plan de vuelo de una manera óptima.

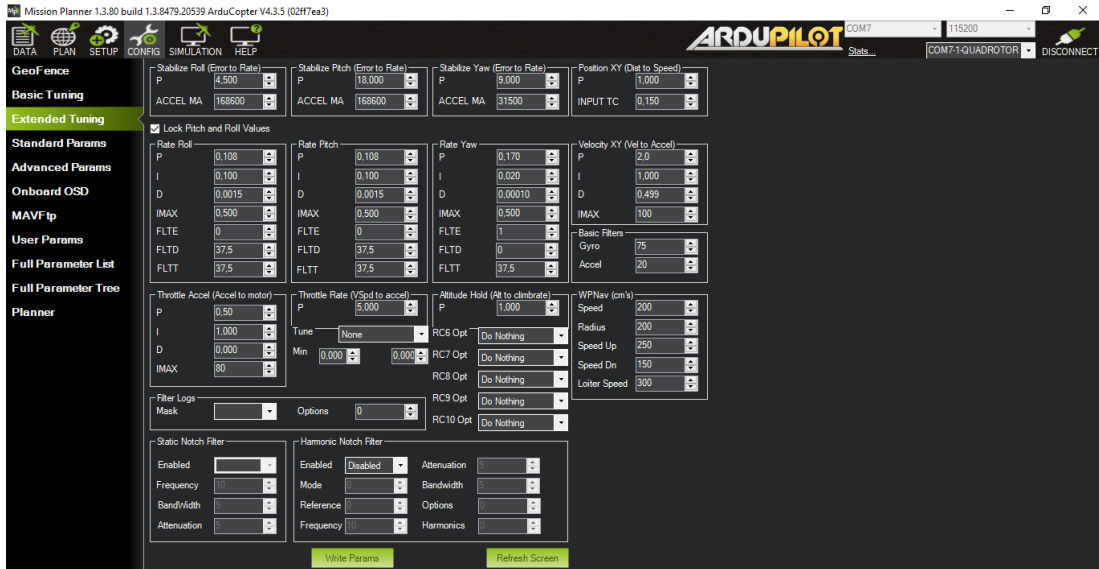


Figura 129. Parámetro de velocidad establecido en 200cm/s

Elaborado por el investigador

Establecidos cada uno de los parámetros necesario en el manejo automático, en la pestaña PLAN se procede a constituir el plan de vuelo necesario para el análisis de una superficie terrena mediante una estructura de tipo cuadrícula. En primera instancia se coloca el punto de partida “HOME” como se muestra en la figura 130, que es el sitio desde donde va a partir el dron a ejecutar la misión de vuelo luego de estar suspendido en el aire.

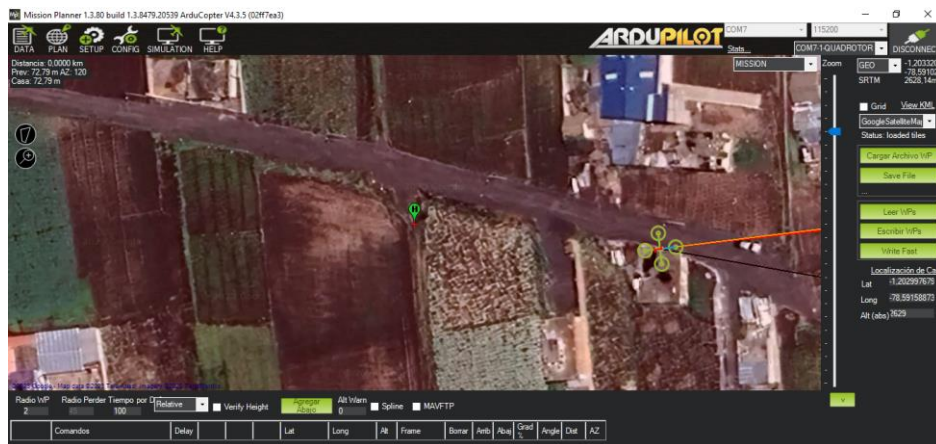


Figura 130. Punto de partida HOME

Elaborado por el investigador

A continuación, se establecen puntos estratégicos dentro de la superficie terrena que se desea inspeccionar. La ruta que se establece puede ser modificada en altura,

dirección y comando de vuelo en el recuadro de la parte inferior para cada tramo de punto a punto, como se observa en la figura 131.

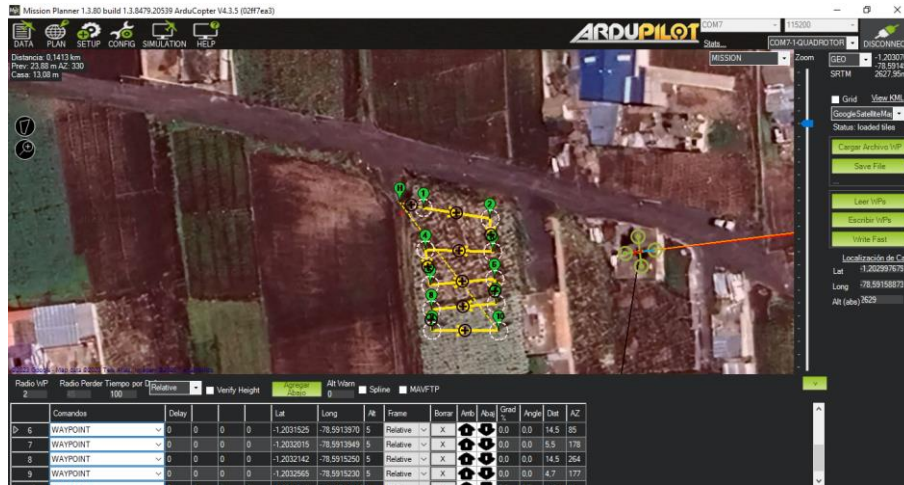


Figura 131. Puntos estratégicos establecidos a lo largo de la superficie
Elaborado por el investigador

Terminado de estructurar el plan de vuelo con los parámetros necesarios para su ejecución, se procede a grabar el archivo en la placa de control mediante el botón Escribir WPs, figura 132. Antes de realizar el grabado de la misión es recomendable revisar la ruta y los puntos en coordenadas geográficas que aparecen en el recuadro de la parte inferior, no deben variar excesivamente uno del otro.

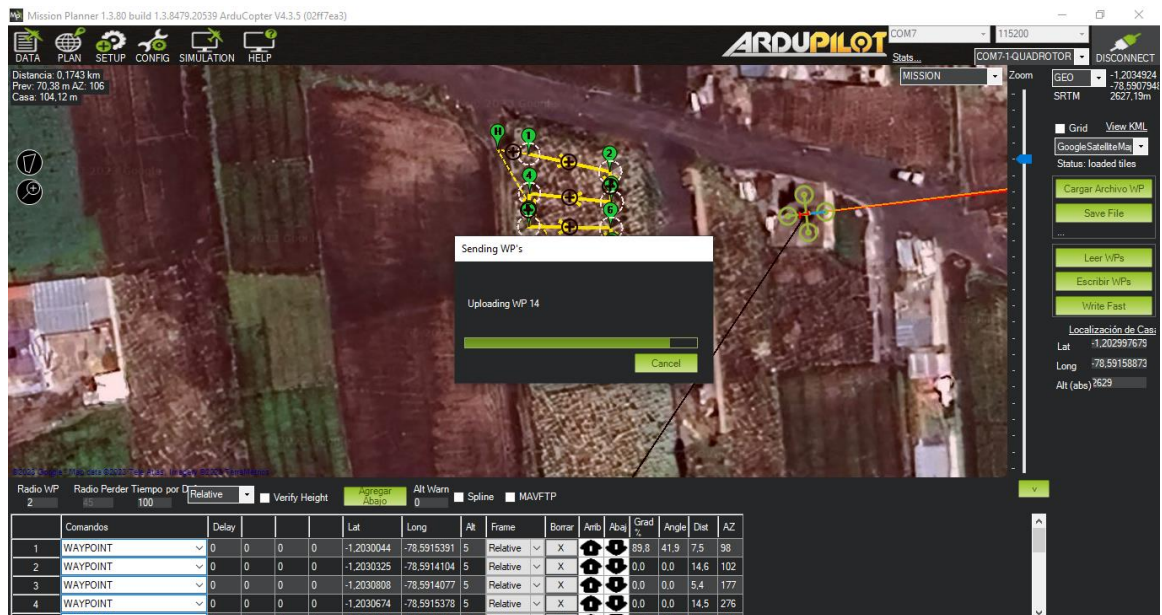


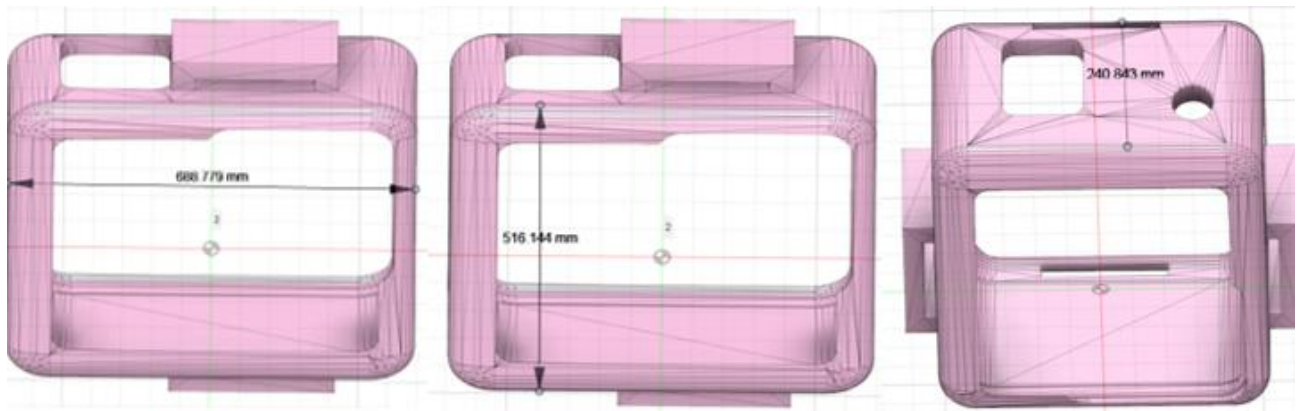
Figura 132 Plan de vuelo grabado en la placa de control
Elaborado por el investigador

3.2.7.7 Diseño e implementación del sistema de teledetección

Para el sistema de teledetección, los elementos principales que lo componen son la cámara GoPro y el microordenador Raspberry Pi 3, los cuales realizan la acción de capturar la imagen según la orden de la interfaz programada para de esta manera almacenarla y realizar el análisis respectivo en cuanto a la visión artificial.

Diseño e implementación de soporte para cámara

Para el transporte seguro de la cámara GoPro HERO 5 en el dron, se diseña un case o protector que se compacte fijamente a la base inferior de la aeronave. Este diseño se desarrolló en el software Fusion360 con una escala de 1:3 tomando en cuenta las medidas del tamaño real de la cámara. Como se observa en la figura 133, resultaron las siguientes dimensiones: 688.779 mm de largo, 516.144 mm de ancho y 240.843



mm de profundidad.

Figura 133. Diseño 3D de soporte de cámara

Elaborado por el investigador

Una vez diseñado el protector de la cámara, se procede a la impresión 3D en material plástico ABS, este proceso duro alrededor de 2 horas. En la figura 134, se observa la implementación del protector de cámara con pequeños tornillos que lo adhieren fijamente a la base inferior del dron.



Figura 134. Implementación del case protector en el dron

Elaborado por el investigador

Configuración del sistema de captura de imagen, recepción de datos GPS y envío al servidor

El código implementa un servidor que interactúa con un dron, obteniendo y procesando imágenes capturadas, así como información de ubicación, y proporcionando una interfaz gráfica para visualizar y analizar las imágenes capturadas. En la figura 135, se muestra el diagrama de flujo del proceso de captura de imagen, recepción de datos GPS y envío al servidor.

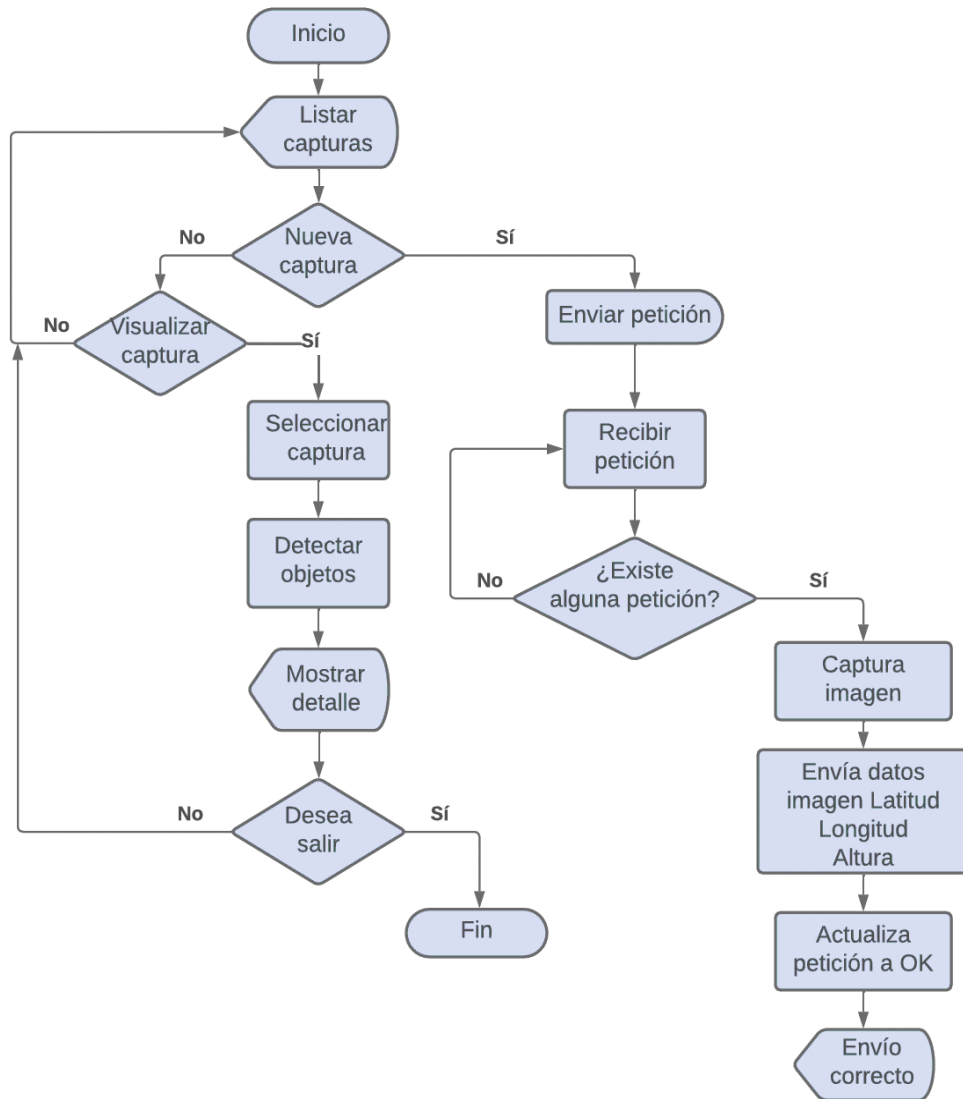


Figura 135. Diagrama de flujo del proceso de captura de imágenes
Elaborado por el investigador

El código implementa un sistema de servidor para PC que realiza las siguientes funciones:

1. **Envío y recepción de imágenes:** El servidor envía una solicitud para obtener una imagen capturada por un dron y recibe la imagen correspondiente. La imagen recibida se utiliza para su visualización.
2. **Visualización de la imagen:** El servidor muestra la imagen capturada en una interfaz gráfica para que pueda ser visualizada por el usuario.
3. **Análisis de problemas:** El servidor utiliza un modelo de detección de objetos llamado YOLOv5 para analizar la imagen capturada y detectar posibles problemas o anomalías en la misma.

4. **Obtención de datos de localización:** Junto con la imagen capturada, el servidor también obtiene datos de latitud, longitud y altura que indican la posición del dron en el momento de la captura.

A continuación, se describe la codificación en Python de la aplicación de captura y envío de imágenes del dron al servidor.

Para el desarrollo de la aplicación de captura y envío de imágenes es importante importar las librerías que se van a emplear. En la figura 136 se observa cada una de estas.

```
# Librerías usadas
import tkinter
import cv2
import io
import time
import threading
import torch
import numpy as np
import tkinter.font as font
from tkinter import ttk
from tkinter import *
import json as json
from PIL import Image, ImageTk
from urllib.parse import urlencode
from urllib.request import Request, urlopen
import base64
```

Figura 136. Librerías utilizadas

Elaborado por el investigador

Cada librería tiene una función específica dentro del sistema, la cual se describe a continuación:

`tkinter`: Es una biblioteca estándar de Python que proporciona herramientas para crear interfaces gráficas de usuario (GUI).

`cv2`: Es el módulo de OpenCV (Open Source Computer Vision Library) en Python, utilizado para el procesamiento de imágenes y videos.

`io`: Módulo de E/S de Python que proporciona clases para manejar flujos de datos.

`time`: Módulo de Python que proporciona funciones relacionadas con el tiempo.

`**threading**`: Módulo de Python que proporciona clases y funciones para trabajar con hilos (threads).

`**torch**`: Es el paquete de PyTorch, un marco de aprendizaje profundo de código abierto utilizado para tareas de aprendizaje automático.

`**numpy**`: Biblioteca de Python utilizada para trabajar con matrices y operaciones matemáticas.

`**tkinter.font**`: Módulo de Python que proporciona funciones para manejar fuentes de texto en la interfaz gráfica.

`**json**`: Módulo de Python para trabajar con datos en formato JSON.

`**PIL**`: El módulo de la biblioteca Python Imaging Library (PIL) que proporciona funciones para abrir, manipular y guardar imágenes.

`**ImageTk**`: Módulo de PIL que proporciona clases para mostrar imágenes en aplicaciones Tkinter.

`**urllib.parse**`: Módulo de Python para analizar y manipular URL.

`**urllib.request**`: Módulo de Python para abrir URL y realizar solicitudes HTTP.

`**base64**`: Módulo de Python para codificar y decodificar datos en formato Base64.

De igual manera, como se observa en la figura 137 se define variables globales, que son utilizadas en el código para almacenar y gestionar diferentes estados y valores relacionados con el funcionamiento del sistema.

```
# Variables globales
general = 0
procesoCa = 0
enviando = 0
estado1 = ""
estado2 = ""
estado3 = ""
estado4 = ""
valCamara = 0
valError = ""
```

Figura 137. Variables globales

Elaborado por el investigador

En la figura 138 se muestra el bloque de código en el que se carga el modelo YOLOv5, utilizado para la detección de objetos en imágenes. La línea de código **time.sleep(1)** introduce una pausa de 1 segundos antes de cargar el modelo. Es útil para asegurarse de que los recursos necesarios estén disponibles antes de cargar el modelo. Si ocurre alguna excepción durante la carga del modelo, se imprimirá en la consola el mensaje "Error".

```
try:
    # Cargar el modelo YOLOv5
    time.sleep(1)
    model = torch.hub.load('ultralytics/yolov5', 'custom', path='D:/TECDES/Sistemas/Dron/best (2).pt')
except:
    print("Error")
```

Figura 138. Uso de archivo entrenado en yolov

Elaborado por el investigador

Creación de ventanas principales del sistema

En el bloque de código que se muestra en la figura 139 se crean y configuran las ventanas principales y secundarias de una aplicación de interfaz gráfica utilizando la biblioteca tkinter.

```
# Crear la ventana principal
ventana = tkinter.Tk()
ventana.state("zoomed")
ventana.title("Dron")
ventana.config(bg="grey")

# Crear ventanas secundarias
proceso = tkinter.Toplevel(ventana)
proceso.state("zoomed")
proceso.title("Dron")
proceso.config(bg="#d9d9d9")

datos = tkinter.Toplevel(proceso)
datos.state("zoomed")
datos.title("Dron")
datos.config(bg="#d9d9d9")
```

Figura 139. Creación de ventanas principales del sistema

Elaborado por el investigador

A continuación, se describe su funcionamiento de las funciones empleadas en el código:

ventana = tkinter.Tk(): Se crea la ventana principal de la aplicación utilizando la clase Tk de tkinter. Esta ventana representa la ventana principal de la aplicación y es el contenedor principal para otros elementos de la interfaz gráfica. Se asigna a la variable ventana.

ventana.state("zoomed"): Se configura el estado de la ventana principal para que se maximice automáticamente al abrirse. Esto hace que la ventana ocupe toda la pantalla.

ventana.title("Dron"): Se establece el título de la ventana principal como "Dron". Este título se muestra en la barra de título de la ventana.

ventana.config(bg="grey"): Se configura el color de fondo de la ventana principal como gris. El argumento "grey" especifica el color en formato hexadecimal o como nombre de color.

Funciones de conexión con la base de datos

En el bloque de código de la figura 140 se definen varias funciones de utilidad para conexión con la base de datos.

```
# Definir funciones de utilidad

def enviar(dir):
    # Enviar una solicitud GET a la dirección especificada
    request = Request(dir)
    json = urlopen(request).read().decode()
    return json

def traerTabla():
    # Obtener la información de la tabla desde un archivo PHP
    request = Request("https://tdeteccion.000webhostapp.com/datos/lista.php")
    json = urlopen(request).read().decode()
    return json

def traerImagen(id):
    # Obtener la información de una imagen específica desde un archivo PHP
    request = Request("https://tdeteccion.000webhostapp.com/datos/seleccionar.php?id="+str(id)+"")
    json = urlopen(request).read().decode()
    return json

def traerEstado():
    # Obtener el estado desde un archivo PHP
    request = Request("https://tdeteccion.000webhostapp.com/datos/captura.php")
    json = urlopen(request).read().decode()
    return json

def registrarBDD():
    # Enviar una solicitud POST para registrar en la base de datos
    enviar("https://tdeteccion.000webhostapp.com/datos/crear.php")
    return
```

Figura 140. Funciones de conexión con la base de datos

Elaborado por el investigador

A continuación, se proporciona una descripción general de cada función:

enviar(dir): Esta función envía una solicitud GET a la dirección especificada por el parámetro dir. Utiliza la biblioteca urllib para realizar la solicitud y devuelve la respuesta en formato JSON.

traerTabla(): Esta función obtiene la información de una tabla desde un archivo PHP. Realiza una solicitud GET a la dirección especificada en el archivo PHP y devuelve la respuesta en formato JSON.

traerImagen(id): Esta función obtiene la información de una imagen específica desde un archivo PHP. Utiliza el parámetro id para construir la URL de la solicitud y obtener la imagen correspondiente. Devuelve la respuesta en formato JSON.

traerEstado(): Esta función obtiene el estado desde un archivo PHP. Realiza una solicitud GET a la dirección especificada en el archivo PHP y devuelve la respuesta en formato JSON.

registrarBDD(): Esta función envía una solicitud POST para registrar datos en una base de datos. Utiliza la función enviar() para enviar la solicitud a la dirección especificada en el archivo PHP.

Abrir página principal y obtener registros de la bdd

La función abrirProceso() se encarga de abrir la ventana de proceso y ocultar las demás ventanas. En la figura 141 se muestra la codificación de dicha función.

```
def abrirProceso():
    # Función para abrir la ventana de proceso y ocultar las demás ventanas
    global datos
    global dat
    tDatos = traerTabla()
    dat.delete(*dat.get_children())
    a = np.array(json.loads(tDatos))
    contar=0
    for x in a:
        dat.insert("", END, text='', values=(
            x['id'],x['fecha'], x['latitud'], x['longitud'], x['altura']))
        contar+=1
    ejecutar_control_estado()

    proceso.state(newstate="zoomed")
    ventana.state(newstate="withdraw")
    datos.state(newstate="withdraw")
```

Figura 141. Abrir página principal y obtener registros de la bdd

Elaborado por el investigador

Cada función empleada realiza una actividad específica, a continuación, se detalla cada una de estas:

- Obtiene los datos de una tabla mediante la función traerTabla() y los almacena en la variable tDatos.
- Limpia el contenido del widget dat utilizando el método delete() con el argumento *dat.get_children(). Esto borra todas las filas existentes en el widget.
- Convierte los datos obtenidos en formato JSON a un array NumPy utilizando la función json.loads(tDatos) y np.array(). Los datos se almacenan en la variable a.
- Itera sobre los elementos del array a y agrega las filas correspondientes al widget dat utilizando el método insert() con los valores de cada elemento. Cada fila contiene los valores de las columnas "id", "fecha", "latitud", "longitud" y "altura".
- Llama a la función ejecutar_control_estado() para controlar el estado.
- Establece el estado de la ventana proceso como "zoomed" para maximizarla.
- Establece el estado de la ventana ventana y datos como "withdraw" para ocultarlas.

Verificación de existencia de alguna petición en el servidor

Las funciones controlarEstado() y ejecutar_control_estado() están relacionadas con el control del estado y la actualización del botón de captura. En la figura 142 se observa la codificación para verificar si esta alguna petición en el servidor.

```
def controlarEstado():
    # Función para controlar el estado y actualizar el botón de captura
    global btnIrCapturar
    tDatos = traerEstado()
    a = np.array(json.loads(tDatos))
    print(len(a))
    if len(a) > 0:
        btnIrCapturar.configure(bg="#FF0000", fg="#FFFFFF")
        btnIrCapturar['text'] = "Capturando"
    else:
        btnIrCapturar.configure(bg="#FFFFFF", fg="#000000")
        btnIrCapturar['text'] = "Capturar"

def ejecutar_control_estado():
    # Ejecutar la función de control de estado periódicamente
    controlarEstado()
    threading.Timer(10, ejecutar_control_estado).start()
```

Figura 142. Verificación de existencia de alguna petición en el servidor

Elaborado por el investigador

La descripción general de cada una de las funciones es la siguiente:

controlarEstado(): Esta función se encarga de controlar el estado y actualizar el botón de captura. A grandes rasgos, realiza lo siguiente:

- Obtiene el estado mediante la función traerEstado() y lo almacena en la variable tDatos.
- Convierte los datos obtenidos en formato JSON a un array NumPy utilizando la función json.loads(tDatos) y np.array(). Los datos se almacenan en la variable a.
- Imprime la longitud del array a en la consola.
- Si la longitud de a es mayor que cero, configura el botón btnIrCapturar con un fondo rojo (#FF0000) y texto blanco (#FFFFFF). Además, cambia el texto del botón a "Capturando".
- Si la longitud de a es cero, configura el botón btnIrCapturar con un fondo blanco (#FFFFFF) y texto negro (#000000). Además, cambia el texto del botón a "Capturar".

ejecutar_control_estado(): Esta función se encarga de ejecutar la función controlarEstado() periódicamente. A grandes rasgos, realiza lo siguiente:

- Llama a la función controlarEstado() para controlar el estado y actualizar el botón de captura.
- Utiliza la clase threading.Timer() para programar la ejecución de la función ejecutar_control_estado() después de un intervalo de tiempo específico (en este caso, 10 segundos).

Actualización de la tabla de datos

La función actualizar(), figura 143, se encarga de actualizar la tabla de datos en la ventana de proceso.

```

def actualizar():
    # Función para actualizar la tabla de datos
    global datos
    global dat
    tDatos = traerTabla()
    a = np.array(json.loads(tDatos))
    dat.delete(*dat.get_children())
    contar=0
    for x in a:
        dat.insert("", END, text=contar, values=(
            x['id'],x['fecha'], x['latitud'], x['longitud'], x['altura']))
        contar+=1

```

Figura 143. Actualización de la tabla de datos

Elaborado por el investigador

actualizar(): Esta función realiza los siguientes pasos para actualizar la tabla de datos:

- Obtiene los datos de la tabla utilizando la función traerTabla() y los almacena en la variable tDatos.
- Convierte los datos obtenidos en formato JSON a un array NumPy utilizando la función json.loads(tDatos) y np.array(). Los datos se almacenan en la variable a.
- Borra todos los elementos existentes en la tabla (dat) utilizando el método delete() con el parámetro *dat.get_children().
- Inicializa una variable contar para llevar un conteo de los elementos insertados en la tabla.
- Recorre el array a en un bucle for y, en cada iteración, inserta un nuevo elemento en la tabla con los valores correspondientes a x['id'], x['fecha'], x['latitud'], x['longitud'] y x['altura']. Los elementos se insertan utilizando el método insert() de la tabla.
- Incrementa el valor de contar en cada iteración para mantener un conteo correcto de los elementos insertados.

Abrir la ventana de datos y cargar la imagen

La función abrirDatos() se encarga de abrir la ventana de datos y cargar la imagen correspondiente. La codificación de esta función se visualiza en la figura 144.

```

def abrirDatos():
    # Función para abrir la ventana de datos y cargar la imagen correspondiente
    global lblImgCam1
    global dat
    seleccion = dat.selection()
    primer_item = seleccion[0]
    valores = dat.item(primer_item, option='values')
    primer_valor_columna_0 = valores[0]
    tDatos = traerImagen(primer_valor_columna_0)

    a = np.array(json.loads(tDatos))
    for x in a:
        base64_data = x['contenido']
        image_bytes = base64.b64decode(base64_data)
        image = Image.open(io.BytesIO(image_bytes))
        new_width = 1280
        new_height = 960
        resized_image = image.resize((new_width, new_height), Image.ANTIALIAS)
        resized_image.save('imagenServidor.png')

    image_path = "imagenServidor.png"
    frame1 = cv2.imread(image_path)
    detect = model(frame1)
    info = detect.pandas().xyxy[0]
    arr = np.asarray(info.confidence.array)
    detector = len(arr)
    if detector > 0:
        arr = info.to_numpy()
        cv2.imwrite("resultado.png", np.squeeze(detect.render()))
        nueva_imagen = Image.open('resultado.png')
        nueva_imagen = nueva_imagen.resize((1280, 500))
        nueva_imgCam1 = ImageTk.PhotoImage(nueva_imagen)
        lblImgCam1.configure(image=nueva_imgCam1)
    else:
        try:
            nueva_imagen = Image.open('imagenServidor.png')
            nueva_imagen = nueva_imagen.resize((1280, 500))
            nueva_imgCam1 = ImageTk.PhotoImage(nueva_imagen)
            lblImgCam1.configure(image=nueva_imgCam1)
            lblImgCam1.image = nueva_imgCam1
        except Exception as e:
            print("Error al procesar la imagen:", str(e))
    datos.state(newstate="zoomed")
    proceso.state(newstate="withdraw")
    ventana.state(newstate="withdraw")

```

Figura 144. Abrir la ventana de datos y cargar la imagen

Elaborado por el investigador

Se realiza una descripción detallada del funcionamiento de la función `abrirDatos()`:

abrirDatos(): Esta función realiza los siguientes pasos para abrir la ventana de datos y cargar la imagen correspondiente:

- Obtiene la selección actual en la tabla (`dat`) utilizando el método `selection()`. Suponemos que se selecciona el primer ítem en este caso.
- Obtiene los valores del ítem seleccionado utilizando el método `item()` con el parámetro `option='values'`. Los valores se almacenan en la variable `valores`.
- Extrae el primer valor de la columna 0 de los valores obtenidos, que suponemos es el identificador de la imagen a cargar. El valor se almacena en la variable `primer_valor_columna_0`.

- Obtiene la información de la imagen correspondiente utilizando la función `traerImagen(primer_valor_columna_0)`. La información se almacena en la variable `tDatos`.
- Convierte los datos obtenidos en formato JSON a un array NumPy utilizando la función `json.loads(tDatos)` y `np.array()`. Los datos se almacenan en la variable `a`.
- Itera sobre los elementos del array `a`. En cada iteración, decodifica los datos base64 de la imagen utilizando `base64.b64decode()` y crea una imagen a partir de los bytes utilizando `Image.open(io.BytesIO(image_bytes))`. Luego, redimensiona la imagen a un nuevo tamaño utilizando el método `resize()` y la guarda en un archivo llamado "imagenServidor.png".
- Carga la imagen redimensionada y guardada utilizando `Image.open('imagenServidor.png')`. Luego, verifica si se detectaron objetos en la imagen utilizando el modelo YOLOv5 (`model`). Si se detectaron objetos, guarda una imagen con los objetos detectados en un archivo llamado "resultado.png". A continuación, redimensiona la imagen de resultados y la muestra en el widget `lblImgCam1` de la ventana de datos utilizando `ImageTk.PhotoImage()` y el método `configure()` del widget.
- Si no se detectaron objetos en la imagen, intenta cargar y mostrar la imagen original redimensionada en el widget `lblImgCam1`. Si ocurre algún error al procesar la imagen, se imprime un mensaje de error.
- Cambia el estado de la ventana de datos a "zoomed" para mostrarla, y oculta las ventanas de proceso y principal cambiando sus estados a "withdraw".

Obtención de datos de velocidad, altura del dron y velocidad del viento proporcionados por el software ArduPilot.

Mediante el software ArduPilot se puede identificar y modificar diferentes parámetros que conllevan al uso del dron en el instante del pilotaje. El software presenta datos de relevancia tales como velocidad y altura del dron como se observa en la figura 145.



Figura 145. Datos mostrados en el software ArduPilot

Elaborado por el investigador

Los datos que se obtienen mediante el software se consideran de gran importancia al momento del vuelo del dron, por esta razón se los presenta dentro del interfaz de Python mediante una puerta de enlace para la conexión a un servidor remoto, de la siguiente manera:

En la figura 146 se visualiza el código para ubicar en la interfaz del sistema la velocidad adecuada dentro del software para el vuelo del dron.

```
lblVelocidad = tkinter.Label(
    seccionCapturas, text="Velocidad Promedio: 2 m/s", font=('Arial 18 '), anchor='w')
lblVelocidad.place(relx=0.5, y=5, relwidth=0.5, height=20)
```

Figura 146. Código para ubicar en la interfaz la velocidad del dron

Elaborado por el investigador

De la misma manera, en la figura 147 se observa las líneas de código para la obtención de los de altura del dron los cuales son definidos dentro del software en la placa de control de vuelo.

```
lblAlturaPromedio = tkinter.Label(  
    seccionCapturas, text="Altura: 3 m", font=('Arial 18'), anchor='w')  
lblAlturaPromedio.place(x=10, y=65, relwidth=0.5, height=20)
```

Figura 147. Obtención de datos de altura del dron

Elaborado por el investigador

La velocidad del viento que se requiere obtener corresponde a la actual magnitud del flujo de aire en la localidad de Ambato en tiempo real. Para llevar a cabo esta medición, se emplean funciones que posibilitan la búsqueda y extracción del código HTML de la página oficial de Microsoft, en donde se presentan estadísticas ambientales. Como se observa en la figura 148, en dicha página se obtiene el valor de 14 km/h, en el tiempo que se realizó el ejemplo.

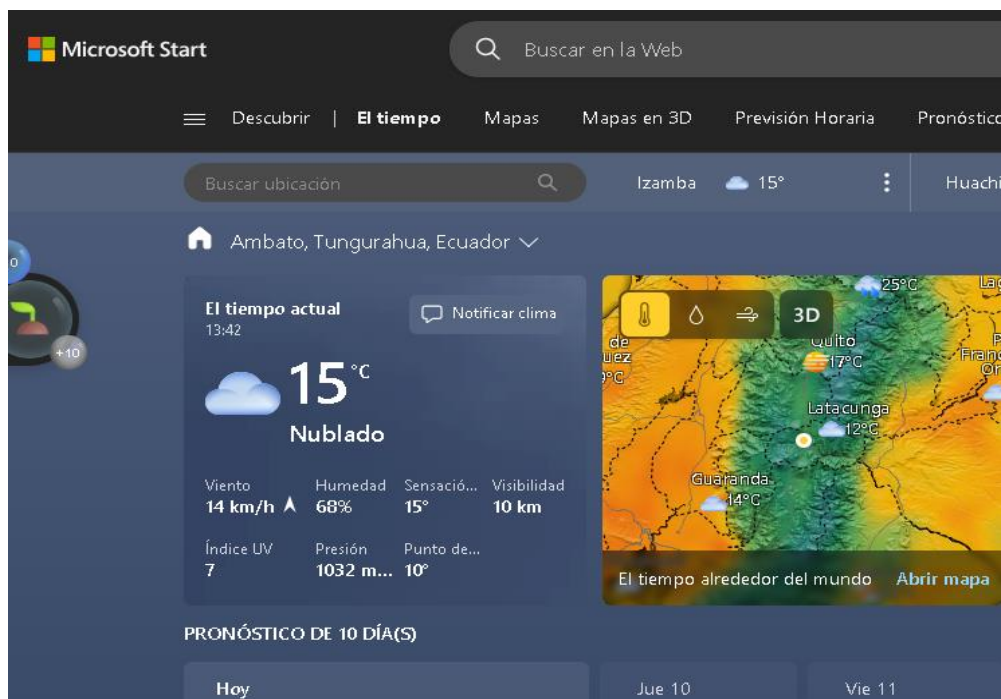


Figura 148. Obtención del dato de la velocidad del viento de la página de Microsoft

Elaborado por el investigador

En la implementación, se utiliza la línea de código "response=requests.get(url)" para realizar una solicitud a través de la biblioteca requests. Esta solicitud tiene como

objetivo obtener el contenido de una URL específica, lo que resulta esencial para adquirir los datos actuales de la velocidad del viento en la ubicación determinada, en este caso, en la ciudad de Ambato. El código empleado se visualiza en la figura 149.

```
def obtener_codigo_html(url):
    try:
        response = requests.get(url)
        if response.status_code == 200:
            return response.text
        else:
            print(f"Error al obtener el código HTML. Código de estado: {response.status_code}")
            return None
    except requests.exceptions.RequestException as e:
        print(f"Error al hacer la solicitud HTTP: {e}")
        return None

def buscar_pagina_en_html(html, pagina_buscada):
    soup = BeautifulSoup(html, 'html.parser')
    # Realiza la búsqueda del elemento deseado, por ejemplo, un título o una etiqueta específica.
    elemento = soup.find('title', text=pagina_buscada)
    if elemento:
        return True
    else:
        return False
```

Figura 149. Obtención del código html de una página determinada

Elaborado por el investigador

De igual forma se realiza la codificación para ubicar en la interfaz del sistema la velocidad del viento, la misma que se visualiza en la figura 150.

```
lblViento = tkinter.Label(
    seccionCapturas, text="Velocidad del Viento: 0 km/h", font=('Arial 18'), anchor='w')
lblViento.place(relx=0.5, y=35, relwidth=0.5, height=20)
```

Figura 150. Código para ubicar en la interfaz la velocidad del viento

Elaborado por el investigador

Función para cambiar el valor de la velocidad el viento

Se ubica la URL deseada y se llama a las funciones de búsqueda de código, se compara con el componente deseado, en este caso el div CurrentDetailLineWinValue, y así se muestra la velocidad del viento, figura 151.

```

def cambiar_viento():
    global lblViento
    url_pagina = 'https://www.msn.com/es-xl/el-tiempo/pronostico/in-Ambato,Tungurahua?loc=eyJ5IjoiQW1iY2'

    # Obtener el código HTML de la página
    codigo_html = obtener_codigo_html(url_pagina)

    if codigo_html:
        # Buscar si la página específica está presente en el HTML
        # Parsear el código HTML con BeautifulSoup
        soup = BeautifulSoup(codigo_html, 'html.parser')

        # Encontrar el elemento que contiene el valor deseado
        # elemento = soup.find('div', id='CurrentDetailLineWindValue')
        elemento = soup.find('div', id='CurrentDetailLineWindValue')
        if elemento:
            # Obtener el texto dentro del elemento y limpiarlo para obtener solo el valor numérico
            valor_numerico = elemento.get_text().split(' ')[0]
            lblViento.config(text=f"Velocidad del Viento: {valor_numerico} km/h")
        else:
            lblViento.config(text=f"Velocidad del Viento: 0 km/h")

    time.sleep(60) # Esperar 1 minuto

```

Figura 151. Función para cambiar el valor de la velocidad el viento

Elaborado por el investigador

Además, se realizó la codificación correspondiente para mostrar la velocidad del viento cada minuto en la interfaz. Esto puede ser útil para realizar operaciones en segundo plano mientras el programa principal sigue ejecutándose en el hilo principal. El código es el que se muestra en la figura 152.

```

hilo_viento = threading.Thread(target=cambiar_viento)
hilo_viento.start()

```

Figura 152. Código para ejecutar el proceso de mostrar la velocidad del viento cada minuto

Configuración de un widget de ttk.Treeview

El bloque de código que se muestra en la figura 153 configura y muestra un widget de ttk.Treeview en la ventana de proceso. La línea de código "style = ttk.Style()" crea una instancia del objeto Style de ttk en Tkinter. El objeto Style proporciona una forma de configurar y personalizar la apariencia de los widgets en una aplicación Tkinter. Permite modificar los colores, fuentes, tamaños y otros atributos visuales de los widgets.


```

style = ttk.Style()
style.theme_use("default")
style.configure("Treeview.Heading", font=('Arial', 14,'bold'))
style.configure("Treeview",background="#D3D3D3",
    foreground="black",
    rowheight=25,
    fieldbackground="#D3D3D3",
    font=('Arial', 14))

style.map('Treeview',
    background=[('selected', 'blue')])

dat = ttk.Treeview(proceso, columns=("col5","col1", "col2", "col3", "col4"))
dat.column("#0",width=0, stretch=NO)
dat.column("col5", width=20, anchor="center")
dat.column("col1", width=80, anchor="center")
dat.column("col2", width=80, anchor="center")
dat.column("col3", width=80, anchor="center")
dat.column("col4", width=80, anchor="center")

dat.heading("#0", text="", anchor=W)
dat.heading("col5", text="id", anchor="center")
dat.heading("col1", text="Fecha", anchor="center")
dat.heading("col2", text="Latitud", anchor="center")
dat.heading("col3", text="Longitud", anchor="center")
dat.heading("col4", text="Altura", anchor="center")

dat.pack()
dat.place(x=55, y=220, relwidth=0.92, relheight=0.42)

```

Figura 153. Configuración de un widget de tk.Treeview

Elaborado por el investigador

A continuación, se proporciona una descripción general del código empleado:

style: Se crea una instancia de `ttk.Style()` para definir el estilo del widget `tk.Treeview`.

style.theme_use("default"): Se establece el tema predeterminado para el estilo del widget.

style.configure("Treeview.Heading", font=('Arial', 14,'bold')): Se configura el estilo de las cabeceras de las columnas del `Treeview` para usar la fuente Arial de tamaño 14 y negrita.

style.configure("Treeview", ...): Se configura el estilo general del `Treeview` con varios atributos, como el color de fondo, el color del texto, la altura de las filas y el color de fondo de los campos.

style.map('Treeview', ...): Se asigna un mapeo de estilos al `Treeview`, especificando que el fondo de los elementos seleccionados debe ser de color azul.

dat: Se crea una instancia de `tk.Treeview` en la ventana de proceso, con columnas etiquetadas como "col5", "col1", "col2", "col3" y "col4".

dat.column(): Se configuran las propiedades de cada columna, como su ancho y alineación.

dat.heading(): Se configuran los encabezados de las columnas.

dat.pack(): Se empaqueta el widget `Treeview` en la ventana de proceso.

dat.place(): Se posiciona el widget `Treeview` en una ubicación específica dentro de la ventana de proceso.

Interfaz gráfica del sistema de teledetección aérea

La pantalla inicial o carátula que se muestra en la figura 154 muestra los datos principales de la universidad, el estudiante que desarrolla y el tutor. También incluye un botón con el texto "Ingresar", el cual permite acceder a la ventana principal del programa.



Figura 154. Interfaz del sistema de teledetección aérea

Elaborado por el investigador

La pantalla principal del sistema de teledetección aérea muestra un título que indica su función. Además, incluye una sección con los detalles del dron, como su nombre, nivel de batería, velocidad del dron, velocidad del viento y altura promedio como se observa en la figura 155.

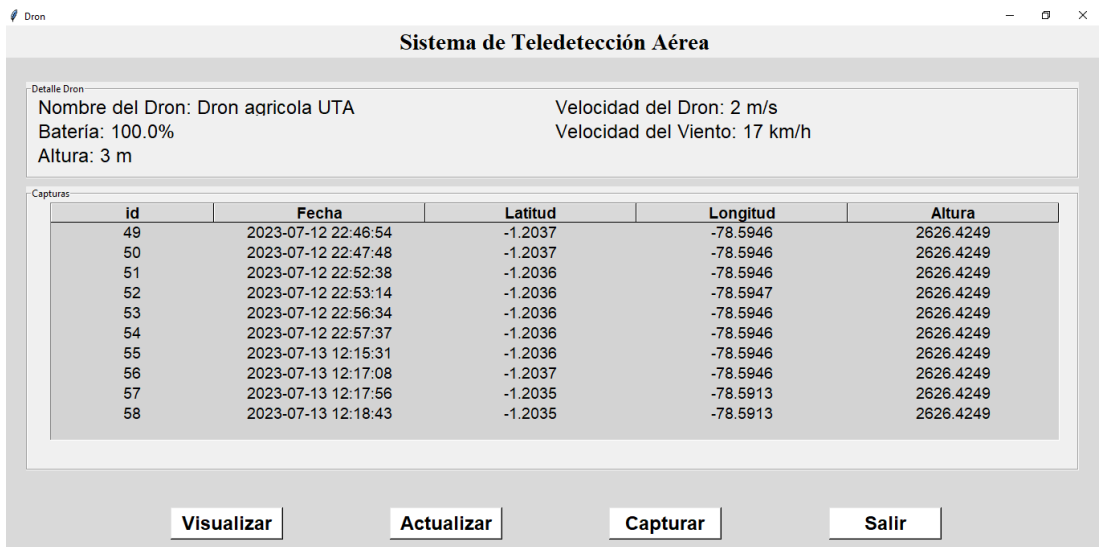


Figura 155. Pantalla principal del sistema de teledetección aérea

Elaborado por el investigador

Otra sección importante es la tabla de capturas, donde se lista la información de las capturas almacenadas en la base de datos, como el ID, la fecha de captura, la latitud, la longitud y la altura, figura 156.

id	Fecha	Latitud	Longitud	Altura
49	2023-07-12 22:46:54	-1.2037	-78.5946	2626.4249
50	2023-07-12 22:47:48	-1.2037	-78.5946	2626.4249
51	2023-07-12 22:52:38	-1.2036	-78.5946	2626.4249
52	2023-07-12 22:53:14	-1.2036	-78.5947	2626.4249
53	2023-07-12 22:56:34	-1.2036	-78.5946	2626.4249
54	2023-07-12 22:57:37	-1.2036	-78.5946	2626.4249
55	2023-07-13 12:15:31	-1.2036	-78.5946	2626.4249
56	2023-07-13 12:17:08	-1.2037	-78.5946	2626.4249
57	2023-07-13 12:17:56	-1.2035	-78.5913	2626.4249
58	2023-07-13 12:18:43	-1.2035	-78.5913	2626.4249

Figura 156. Tabla con lista de capturas

Elaborado por el investigador

Para interactuar con las capturas, se presentan botones con opciones adicionales, como visualizar, actualizar, capturar y salir. Estas opciones permiten realizar acciones específicas dentro del sistema. El botón visualizar permite abrir la ventana donde se muestra el resultado del detalle de la imagen que se ha enviado a comparar con el modelo YOLOv5x, figura 157.

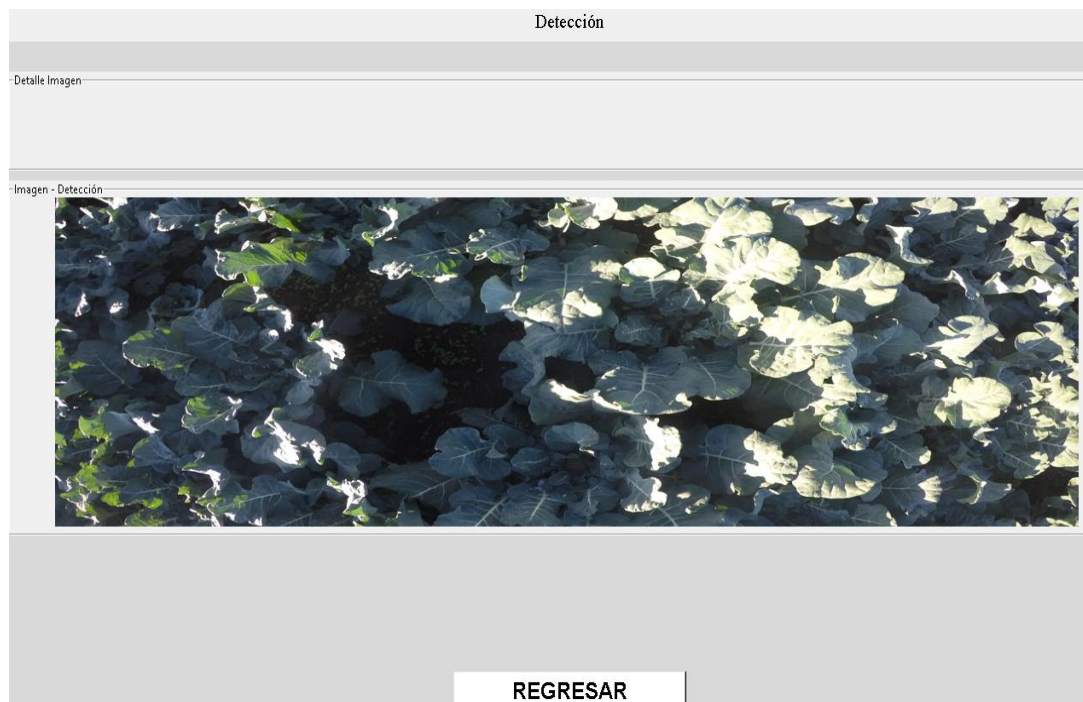


Figura 157. Botón visualizar
Elaborado por el investigador

El botón capturar permite enviar una solicitud al servidor para que el cliente o el código desde el dron capture la imagen y la suba al servidor, cuando no hay petición de captura el botón se encuentra de color plomo con el texto capturar, y si hay una petición el color cambia a rojo y el texto dice capturando. El botón salir regresa a la ventana de caratula.

Entrenamiento del algoritmo de visión artificial YOLO v5x

Para iniciar con el entrenamiento del algoritmo de visión artificial, fue necesaria la obtención de una gran cantidad de información gráfica de cada etapa de crecimiento del cultivo en diversas condiciones climáticas, alturas, ángulos de visión y cada parámetro externo que pueda afectar al aprendizaje y reconocimiento de objetos por parte del sistema de teledetección. Se recolecto alrededor de 2400 imágenes para iniciar con el entrenamiento del algoritmo YOLO v5x, figura 158.

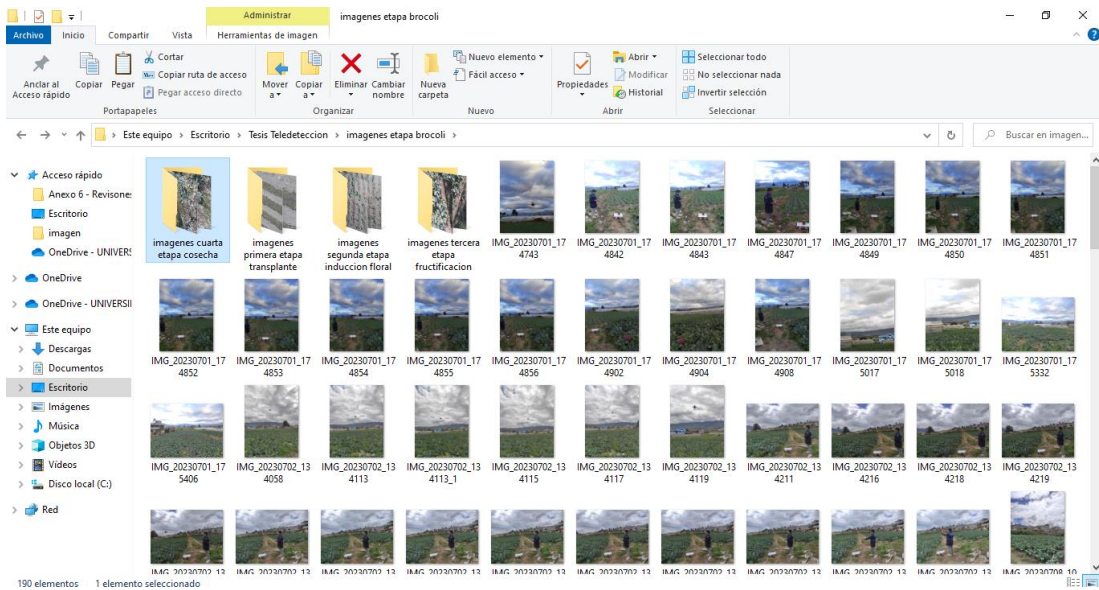


Figura 158. Imágenes recolectadas para el entrenamiento del algoritmo YOLO v5x

Elaborado por el investigador

Etiquetado de imágenes

Como se observa en la figura 159, para etiquetar las imágenes se utiliza la página online Make Sense para generar el dataset, con la finalidad de construir un conjunto de datos etiquetados.

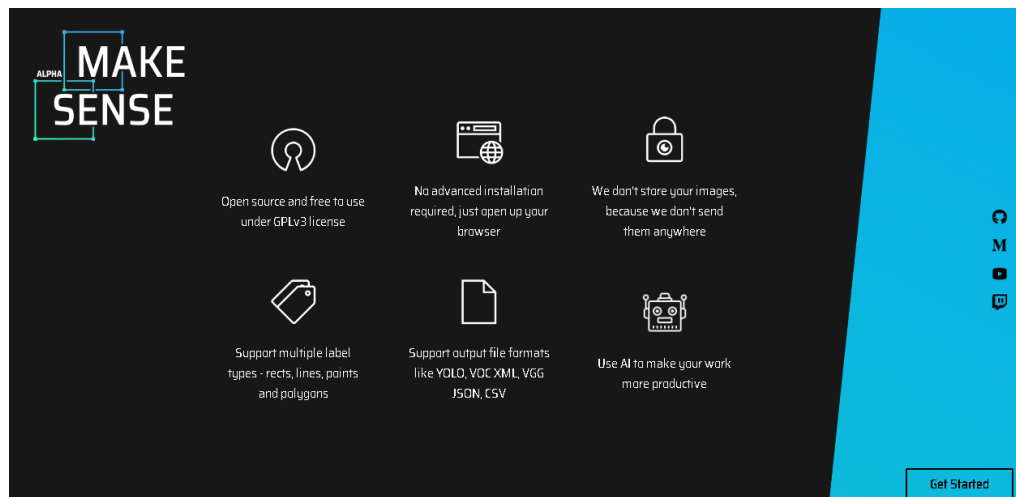


Figura 159. Etiquetado de imágenes con Make Sense

Elaborado por el investigador

Para agregar una nueva etiqueta, se debe hacer clic en el signo + en la esquina superior izquierda del cuadro de mensaje e ingresar la etiqueta en el campo de texto

"Insertar etiqueta". Como se puede observar en la figura 160 para el presente estudio se crearon 3 etiquetas, Saludable, Mala Hierba y Estrés hídrico.

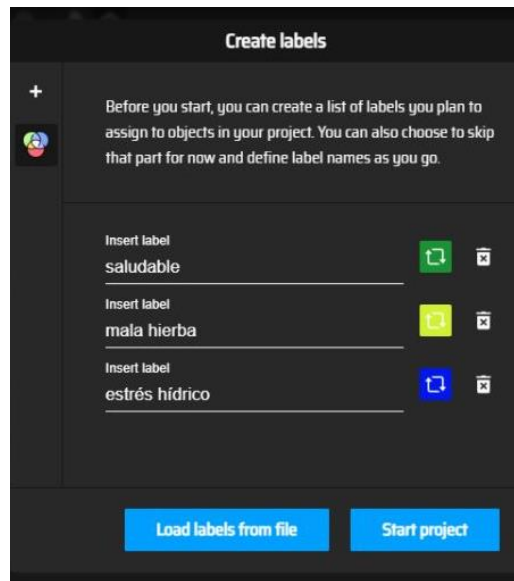


Figura 160. Establecimiento de etiquetas
Elaborado por el investigador

Cada una de las etiquetas fue establecida de acuerdo a las siguientes características

Etiqueta 1-Brócoli saludable: se caracteriza por tener hojas de color verde oscuro, unos floretes compactos y de igual manera de color verde oscuro, con un tallo duro y grueso, figura 161. Entre las etapas de fructificación y cosecha su tamaño oscila entre 10 a 20 cm [58].



Figura 161. Etiqueta del brócoli saludable
Elaborado por el investigador

Etiqueta 2-Estrés Hídrico: Un cultivo en estrés hídrico cerrará sus estomas, enrolla sus hojas reduciendo el crecimiento de sus partes, afectando notablemente al

rendimiento. La falta de agua limita la capacidad de la planta para realizar la fotosíntesis y producir energía, como resultado, el crecimiento de las hojas y tallos puede verse inhibido, lo que afecta el desarrollo vegetativo general. De igual forma puede disminuir el tamaño y peso de las cabezas de brócoli, así como afectar su calidad general, figura 162. Las cabezas pueden volverse más pequeñas, menos densas y presentar una menor calidad organoléptica, provocando a futuro la aparición de agentes patógenos en la planta [59].

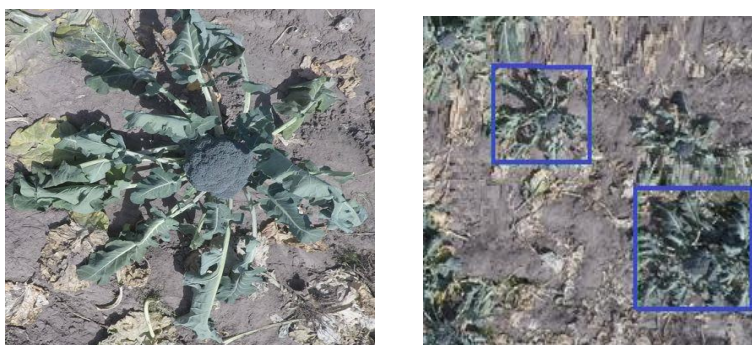


Figura 162. Etiqueta 2-Estrés Hídrico

Elaborado por el investigador

Etiqueta 3- Mala hierba: el cultivo de brócoli suele enfrentarse a la presencia de hierbas no deseadas. Estas plantas indeseables compiten con los brotes jóvenes por espacio, luz solar, agua y nutrientes. Esta competencia puede afectar el crecimiento y desarrollo del brócoli, ya que las malas hierbas pueden absorber estos recursos de manera más eficiente, dejando al brócoli con menos disponibilidad de ellos, figura 163. Como resultado, el brócoli puede mostrar un crecimiento más lento y un desarrollo más tardío, consecuentemente la presencia de enfermedades en el cultivo [60].



Figura 163. Etiqueta 3- Mala hierba

Elaborado por el investigador

Como se puede observar en las Figuras 161, 162 y 163 se encuadra la planta en la imagen añadiendo la etiqueta y guardando en cuadro delimitador en el formato que se seleccionó, que para el desarrollo del presente proyecto es Yolo v5x, por lo que la información es guardada en un documento de texto con extensión txt.

Código para crear y entrenar el modelo

Para realizar el proyecto se hace uso de la herramienta en línea Google Colab en el que se tiene disponible una GPU que debe ser activado para ejecutar los procesos de entrenamiento de manera rápida. De igual manera se emplea Gogle drive para almacenar los códigos, modelos, imágenes y otros archivos relevantes.

Para cargar etiquetas desde Google drive se emplea la codificación que se observa en la figura 164. Este importa el módulo "drive" de la biblioteca "google.colab" en el entorno de Google Colab. El módulo "drive" proporciona funciones y métodos para interactuar con Google Drive desde el entorno de Colab. Una vez que Google Drive está montado en el entorno de Colab, se puede acceder a los archivos almacenados en Google Drive utilizando rutas relativas a `"/content/drive/"`. Esto facilita la carga y el acceso a los datos almacenados en Google Drive durante el desarrollo y ejecución del código en Google Colab.

```
from google.colab import drive
drive.mount('/content/drive')
```

Figura 164. Cargar etiquetas desde Google drive
Elaborado por el investigador

Para descomprimir la carpeta que contiene las etiquetas se lo realiza mediante el comando unzip, se ejecuta la codificación que se muestra en la figura 165. Esta línea de código descomprime un archivo ZIP llamado "data.zip" que se encuentra en `"/content/drive/MyDrive/"` y extrae los archivos en la carpeta `"/content/"`.

```
!unzip -q /content/drive/MyDrive/data.zip -d /content/
```

Figura 165. Descomprimir carpeta que contiene las etiquetas
Elaborado por el investigador

Antes de llevarse a cabo el entrenamiento es importante instalar los repositorios de Google Colab. Esta acción implica la descarga de la carpeta principal de YOLOv5 desde GitHub, con el fin de instalar los requisitos necesarios que se encuentran dentro de estos archivos. Además, se debe utilizar la biblioteca "torch" durante el proceso de entrenamiento como se observa en la figura 166.

```
!git clone https://github.com/ultralytics/yolov5 # clone
%cd yolov5
%pip install -qr requirements.txt # install

import torch
import utils
display = utils.notebook_init() # checks
```

Figura 166. Clonar e instalar requerimientos yolov

Elaborado por el investigador

Para la ejecución del entrenamiento se emplea la línea de código que se observa en la figura 167, en donde se establece que el entrenamiento del modelo en donde se ejecuta el archivo de Python es denominado Train.py, se entrena 544 imágenes, pasándole como argumento el batch.size, lo que significa que toma las fotos en lotes de 35 en 35. De igual manera epochs indica el número de épocas o iteraciones completas a través del conjunto de datos durante el entrenamiento, en este caso, se realizarán 200 épocas. El comando data1.yaml especifica el archivo de configuración YAML que contiene la información sobre el conjunto de datos, como las rutas de los archivos de imágenes y etiquetas, el número de clases, etc. Así también contiene los pesos pre-entrenados de la arquitectura Yolov5 en su versión x .

```
!python train.py --img 544 --batch 35 --epochs 200 --data data1.yaml --weights yolov5x.pt --cache
```

Figura 167. Ejecución del entrenamiento

Elaborado por el investigador

3.2.7.8 Diseño e implementación del sistema de geolocalización

Para la implementación del sistema de geolocalización, los elementos que lo componen son modulo SIM 808, Raspberry Pi 3, antena GPS y Mobile Wifi HUAWEI E586 los cuales proporcionan los datos de geolocalización según la orden de captura de imagen dentro del interfaz. En la figura 168, se muestra la

implementación del sistema con cada uno de sus componentes ubicados respectivamente.

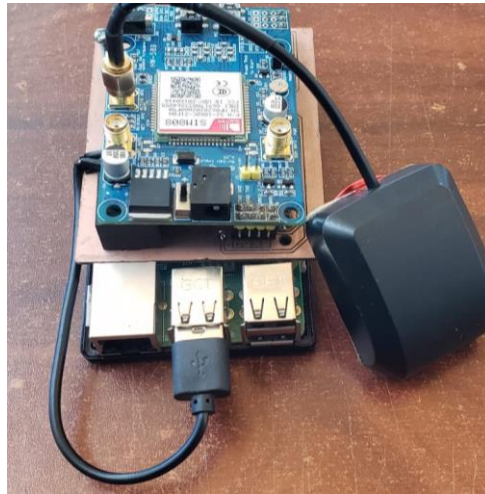


Figura 168. Implementación del sistema de geolocalización
Elaborado por el investigador

Diseño e implementación de frame para Sistema de geolocalización

Para la ubicación y transporte del sistema de geolocalización sobre el dron, fue necesario el diseño de un frame con las medidas respectivas de la tarjeta Raspberry Pi, debido a que es el componente base que sostiene el sistema. En la figura 169 se observan las dimensiones en una escala de 1:3 del diseño: 456.209mm de ancho, 796.308 mm de largo y 39.846 mm de profundidad.

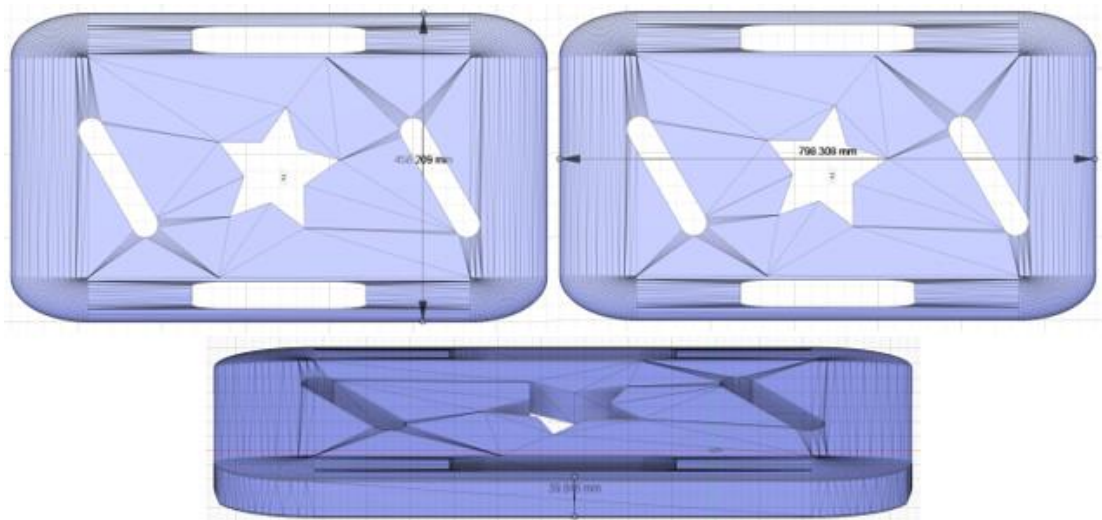


Figura 169. Diseño 3D de frame para sistema de geolocalización
Elaborado por el investigador

El diseño se imprimió con material plástico ABS en un tiempo aproximado de 55 minutos. En la figura 170 se observa la implementación de este frame sobre la base superior del dron para el transporte seguro del sistema de geolocalización.

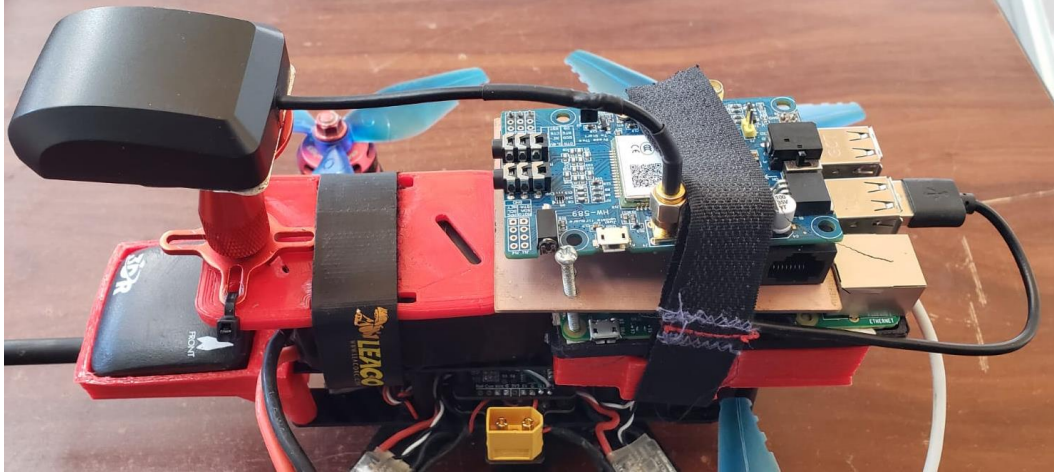


Figura 170. Implementación de frame sobre el dron

Elaborado por el investigador

Configuración del interfaz para el sistema de geolocalización

En esta etapa se desarrolla el código en Python del sistema de geolocalización mediante el cual se obtiene los datos de posicionamiento del dron al momento de realizar la inspección. Los datos proporcionados por el sistema de geolocalización se almacenan de igual manera en la base de datos de phpMyAdmin.

Capturar imagen del dron

Verifica si existe petición, si es así, se conecta a la cámara y realiza la captura de la imagen la almacena en el dispositivo, la muestra por interfaz y la envía por base de datos, figura 171.

```

def controlarEstado():
    global lblDetalle
    global lat
    global lon
    global alt
    global valorEstado
    global lblImgCam1
    tDatos = traerEstado()
    a = np.array(json.loads(tDatos))
    print(len(a))
    if len(a) > 0:
        for x in a:
            valorEstado=x['id']
            lblDetalle.configure(bg="#FF0000", fg="#FFFFFF")
            lblDetalle['text'] = "Capturando"

            cap = cv2.VideoCapture(0)
            leido0, frame0 = cap.read()
            cv2.imwrite("captura.png", frame0)
            nueva_imagen = Image.open('captura.png')
            nueva_imagen = nueva_imagen.resize((900, 450))
            nueva_imgCam1 = ImageTk.PhotoImage(nueva_imagen)
            lblImgCam1.configure(image=nueva_imgCam1)

```

Figura 171. Capturar imagen del dron

Elaborado por el investigador

Trae el estado actual, o petición para capturar imagen como se muestra en la figura 172.

```

def traerEstado():
    request = Request("https://tdeteccion.000webhostapp.com/datos/captura.php")
    json = urlopen(request).read().decode()
    return json

```

Figura 172. Petición para capturar imagen

Elaborado por el investigador

En la figura 173 se muestra el código para la conexión a la cámara y tomar la captura de la foto Cv2. permite conectarse a la cámara y toma la captura de la foto, crea la foto de manera local. Se conecta por serial al sim808 para obtener los datos de gps, y almacenarlos en variables locales

```

a = np.array(json.loads(tDatos))
print(len(a))
if len(a) > 0:
    for x in a:
        valorEstado=x['id']
        print('Capturando')
        cap = cv2.VideoCapture('http://192.168.1.100:4747/video')
        ret, frame = cap.read()
        cv2.imwrite('captura.png', frame)
        cap.release()

    try:
        comando = serial.Serial("/dev/serial0",baudrate=9600,timeout=1)
        comando.write(b'AT+CGNSINF\r\n')
        response=comando.read(130).decode('utf-8')
        infor=response.split(",")
        lat=float(infor[3])
        lon=float(infor[4])
        alt=float(infor[5])
    except:
        print("No hay conexion Serial")

```

Figura 173. Conexión a la cámara y toma la captura de la foto

Elaborado por el investigador

En la figura 174 se muestra el código para seleccionar la URL para subir los datos de imagen con la localización, convierte la imagen en base64 para enviarlos a mysql, envía los datos por método post, si la respuesta o actualización es correcta muestra un mensaje satisfactorio caso contrario error.

```

url_php = 'https://tdeteccion.000webhostapp.com/datos/imagen.php'
imagen = 'captura.png'

fecha="2023-06-23"
with open(imagen, 'rb') as file:
    imagen_bytes = file.read()
    imagen_base64 = base64.b64encode(imagen_bytes).decode('utf-8')
    #print(imagen_base64)
    print(valorEstado)
    data = {'id': str(valorEstado),'alt': str(alt),'lat': str(lat),'lon': str(lon),'fecha': str(fecha),'contenido': str(imagen_base64)}
    #data = {'id': str(valorEstado)}
    response = requests.post(url_php, data=data)
    # Verificar la respuesta del servidor
if response.status_code == 200:
    print('La imagen se ha subido correctamente.')
    print(response.text)
else:
    print('Error al subir la imagen.')
    #actualizarBDD()

```

Figura 174. URL para subir los datos de imagen con la localización

Elaborado por el investigador

Cálculo de peso de soporte para el dron

Es necesario tener en cuenta el peso de carga que soporta el dron, en la tabla 11 se especifica que su soporte máximo es de 500 gramos. En la tabla 14 se muestra la descripción de los accesorios y dispositivos externos con sus pesos respectivos.

Tabla 14. Peso de los componentes externos acoplados al dron

Peso de componentes externos acoplados al dron	
Modulo M8N GPS	14 gramos
Soporte para cámara	50 gramos
Frame para Sistema de Geolocalización	22 gramos
Cámara GoPro Hero 5	153 gramos
SIM 808	31 gramos
Raspberry PI 4	45 gramos
Antena GPRS	1 gramo
Mobile Wifi HUAWEI E586	80 gramos
Placa PCB	0.5 gramos
Fuente de alimentación	2 gramos

Fuente: Elaborado por el investigador

Definidos cada uno de los pesos respectivos, se procede a realizar la sumatoria para el cálculo del peso neto y comparar con lo establecido en las especificaciones del dron para su soporte.

$$\mathbf{Peso\ Neto} = 14g + 50g + 22g + 153g + 31g + 45g + 1g + 80g + 0.5g + 2g$$

$$\mathbf{Peso\ Neto} = 398.5\ \text{gramos}$$

El sistema implementado tiene un peso neto de 398.5 gramos, por ende, es factible su acoplamiento para la movilidad en el dron debido a que es un peso menor al rango definido en sus especificaciones, como se observa en la figura 175.

Mientras que el peso total del sistema en conjunto con el dron es de:

$$\mathbf{Peso\ Total} = 398.5g + 98g$$

$$\mathbf{Peso\ Total} = 496.5\ \text{gramos}$$

$$\mathbf{Peso\ Total} \approx 0.5\ \text{kilogramos}$$

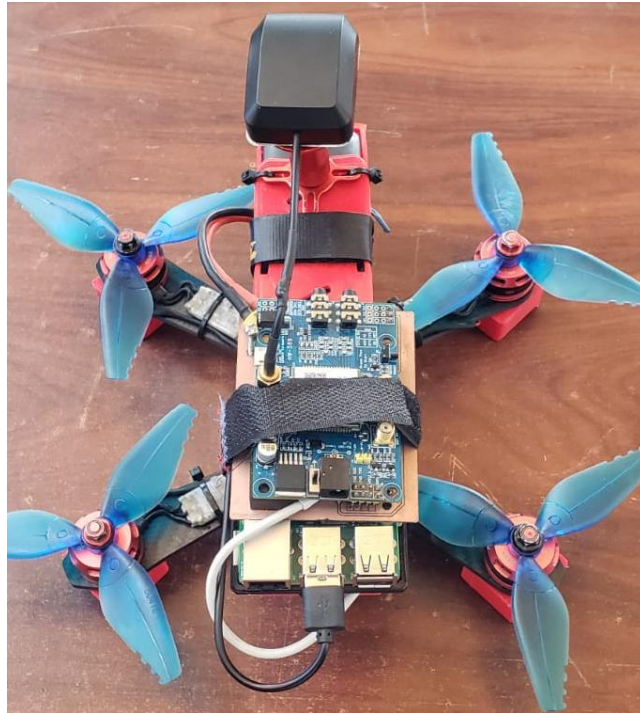


Figura 175. Dron implementado con el sistema
Elaborado por el investigador

Calibración del dron con peso extra

Es necesario realizar la configuración de calibración del dron nuevamente debido al peso extra que este va a transportar durante la inspección del cultivo de brócoli. Para este proceso se utiliza el software ArduPilot.

Se conecta el dron a la PC mediante cable USB para que pueda reconocer la tarjeta de control, en la pestaña INSTALL SETUP se procede a instalar el firmware necesario para el reconocimiento del tipo de vehículo que se va a utilizar dentro del software, el firmware instalado fue AC 3.1.5, figura 176.

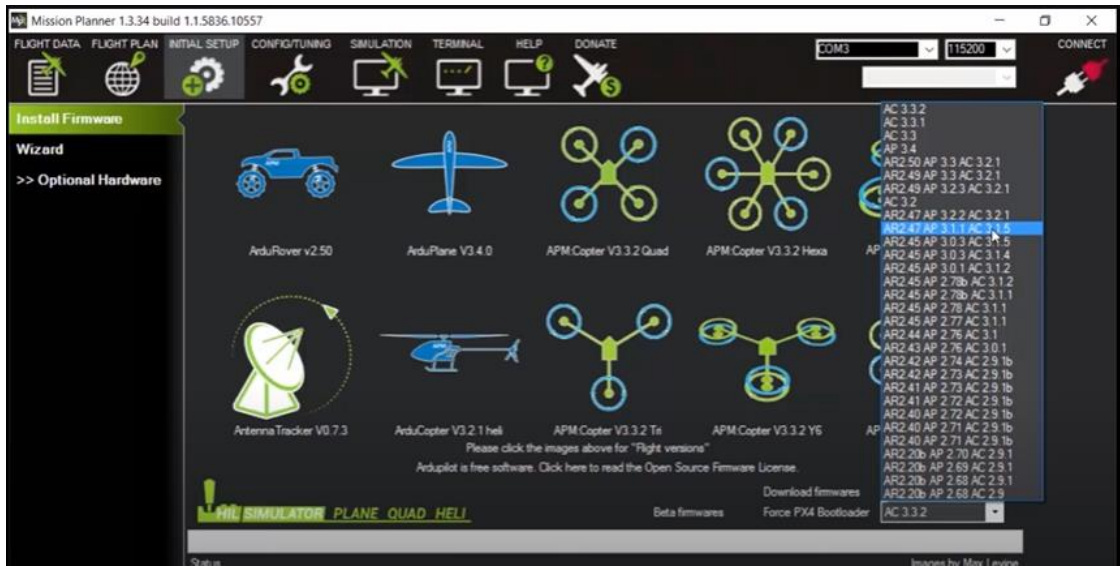


Figura 176. Instalación de Firmware

Elaborado por el investigador

Se procede a elegir en el software el modelo de configuración tipo COPTER, figura 177, mediante esto la tarjeta reconoce el número de motores por los que está compuesta la aeronave.



Figura 177. Selección de tipo de vehículo, en este caso tipo COPTER

Elaborado por el investigador

Posterior a esto, se procede a calibrar el cuerpo del dron mediante la configuración del compás en donde se da giros al dron de forma manual en cada uno de sus ejes (x,y,z) de esta manera se obtiene una serie de puntos que se reflejan en el software

marcados por los giros efectuados. Llegada la cantidad necesaria de puntos para calibrar el compás magnético, automáticamente el programa da por finalizado este proceso como se observa en la figura 178.

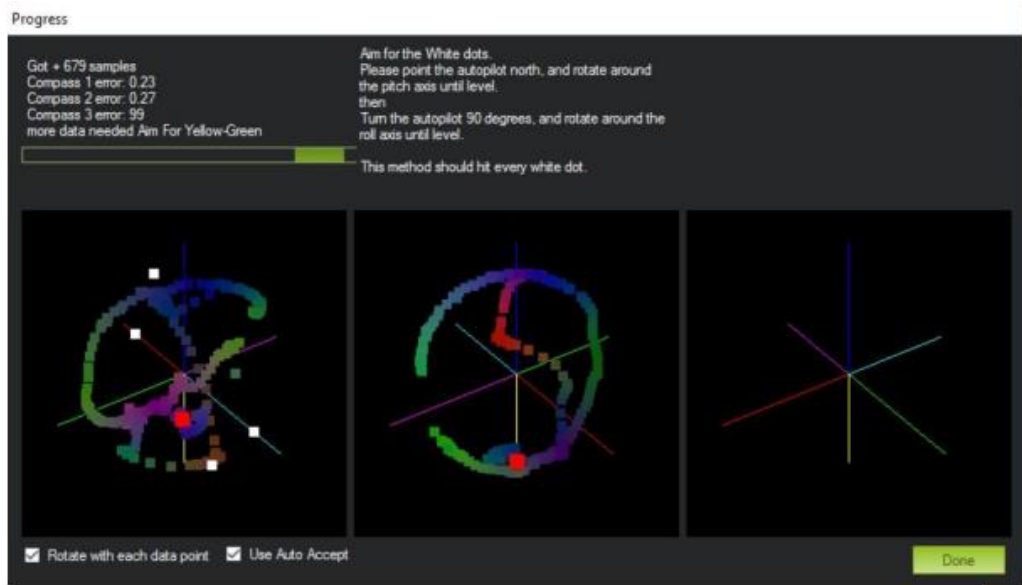


Figura 178. Calibración de ejes x,y,z

Elaborado por el investigador

A continuación, se procede a calibrar los acelerómetros mediante las posiciones mostradas en la pantalla como se muestra en la figura 179. Luego de efectuar cada posición se debe guardar el proceso presionando el botón CONTINUAR.

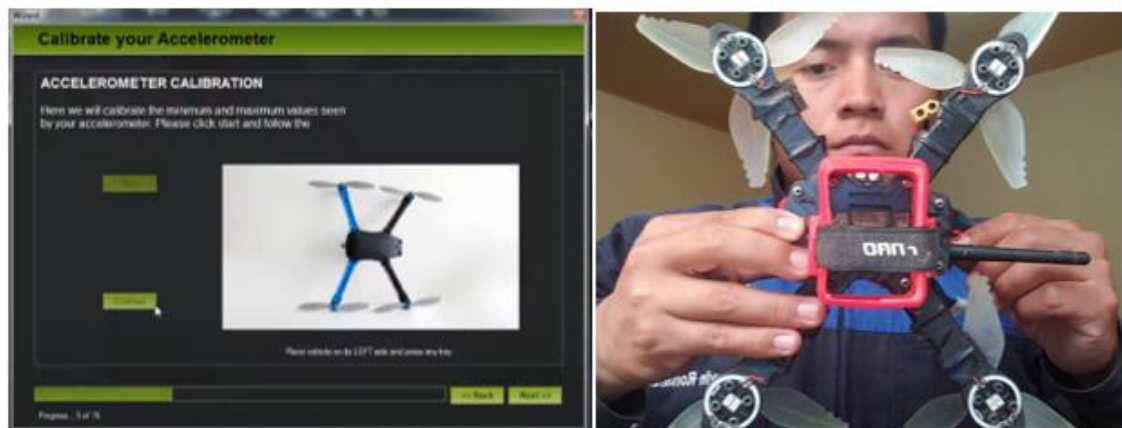


Figura 179. Calibración del dron mediante posiciones mostradas en el programa

Elaborado por el investigador

Finalmente, se configura el nivel posicionando el dron en una superficie nivelada que servirá de referencia para su estabilización de nivel en el aire, terminada la

calibración el nivel en el software debe posicionarse en 0, esto quiere decir que el dron esta calibrado correctamente, figura 180.



Figura 180. Calibración de nivel
Elaborado por el investigador

Cálculo de autonomía de vuelo requerido

Para la inspección del cultivo con el dron cuadricóptero es necesario realizar un vuelo autónomo preprogramado de tipo cuadrícula como se observa en la figura 181, esto facilitara la captura de imágenes en todo el perímetro del cultivo para su procesamiento posterior. Además, para prevenir riesgos durante la inspección del cultivo se integraron al sistema, un control de nivel de batería y la protección necesaria para los accesorios utilizados.



Figura 181. Planeación de vuelo tipo cuadrícula
Elaborado por el investigador

Para el cálculo de la autonomía de vuelo se toma como referencia una extensión de terreno de 100 metros de largo por 100 metros de ancho, con el cual se procede a obtener los datos de la distancia total recorrida en la planeación del vuelo del dron como se observa en la figura 182, la velocidad con la que se ejecuta esta acción y el tiempo de retorno.

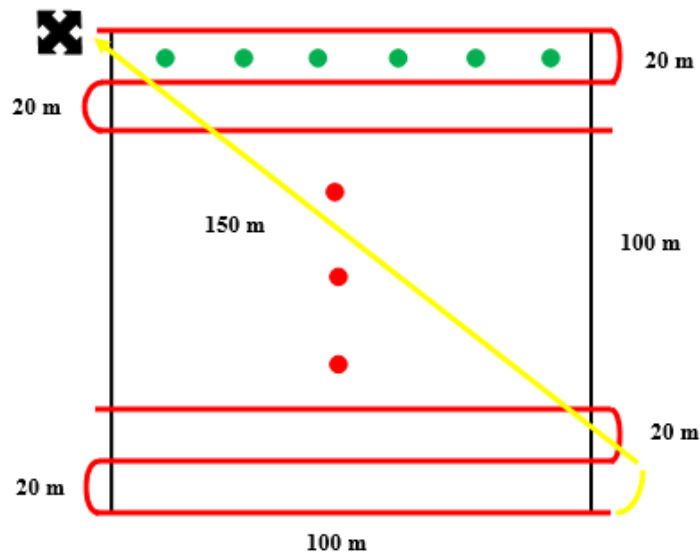


Figura 182. Distancias recorridas en un plan de vuelo

Elaborado por el investigador

- Ruta horizontal 100 metros por línea = 10
- Ruta curva en giro del dron 20 metros por vuelta = 11
- Ruta curva para retorno del dron 25 metros = 1
- Ruta de retorno del dron 150 metros = 1
- Ruta vertical de aterrizaje del dron 4 metros = 1

Con cada uno de los datos obtenidos se construye la siguiente tabla 15 de resultados:

Tabla 15. Resultados de distancias obtenidas

Detalle	Longitud	Cantidad	Total por ruta
Ruta horizontal del dron	100 metros	10	1000 metros
Ruta curva en giro del dron	20 metros	11	220 metros
Ruta curva para retorno del dron	25 metros	1	25 metros
Ruta de retorno del dron	150 metros	1	150 metros

Ruta vertical de aterrizaje del dron	4 metros	1	4 metros
		TOTAL	1400 metros

Fuente: Elaborado por el investigador

El dron cuadricóptero debe recorrer una distancia total de 1400 metros para cubrir una hectárea de terreno, con este dato se logra calcular el tiempo autónomo de vuelo mediante la siguiente formula:

$$Tiempo\ autonomo\ de\ vuelo = \frac{Distancia\ total}{Velocidad * 60}$$

En la configuración de vuelo preprogramado se coloca una velocidad de desplazamiento de 6m/s, entonces:

$$Tiempo\ autonomo\ de\ vuelo = \frac{1400m}{6m/s * 60}$$

$$Tiempo\ autonomo\ de\ vuelo = 3.888\ min$$

Efectuados los cálculos anteriores se obtiene que el dron cuadricóptero cumplirá el plan de vuelo establecido para una hectárea en aproximadamente 4 minutos.

3.2.8 Pruebas de funcionamiento y resultados

Se realizaron un total de 3 pruebas de funcionamiento, acorde a lo realizado en la implementación del sistema. Cada una de estas pruebas ayudaran a verificar el funcionamiento correcto del prototipo y a definir un porcentaje de confiabilidad con respecto a otros sistemas dirigidos al análisis de cultivos.

Prueba 1: Evaluación del plan de vuelo para dron autónomo en los tiempos calculados

Se evalúa la ejecución del plan de vuelo elaborado para el dron antes de realizar la inspección de la superficie del cultivo y tiempo de autonomía calculado para un terreno de $10000m^2$.

Resultados:

Para el desarrollo de la primera prueba es necesario conocer físicamente el lugar seleccionado para la demostración, debido a la posible presencia de obstáculos en el entorno como postes, arboles, estructuras entre otras. Para que de esta manejar se pueda estructurar el plan de vuelo con las evasiones respectivas y no se produzca

ningún accidente durante su ejecución. En la figura 183 se muestra el lote de terreno de $10000m^2$ seleccionado y su ubicación dentro del software Mission Planer para la ejecución del plan de vuelo.



Figura 183. Lote de terreno seleccionado para la prueba de vuelo

Elaborado por el investigador

Definida la posición de la superficie de terreno, se procede a conectar el dron a la PC para identificar mediante el módulo GPS el último dato de ubicación de la aeronave y dirigirlo a su nueva posición para estructurar el plan de vuelo en tipo cuadrícula. Como se muestra en la figura 184, se ubica en primera instancia el punto de partida HOME y posterior a esto los puntos estratégicos para el plan de vuelo que en este caso son 11, se plantea una altura de 5 metros para toda la trayectoria a excepción del tramo 4 al 5 en donde la altura es de 10 metros para demostrar el alza del vuelo en el dron ante un posible obstáculo presente, toda la trayectoria se desplazara a una velocidad de 600 cm/s o 6m/s que es la velocidad a la que se calculó el tiempo de vuelo dentro de un terreno de $10000m^2$.

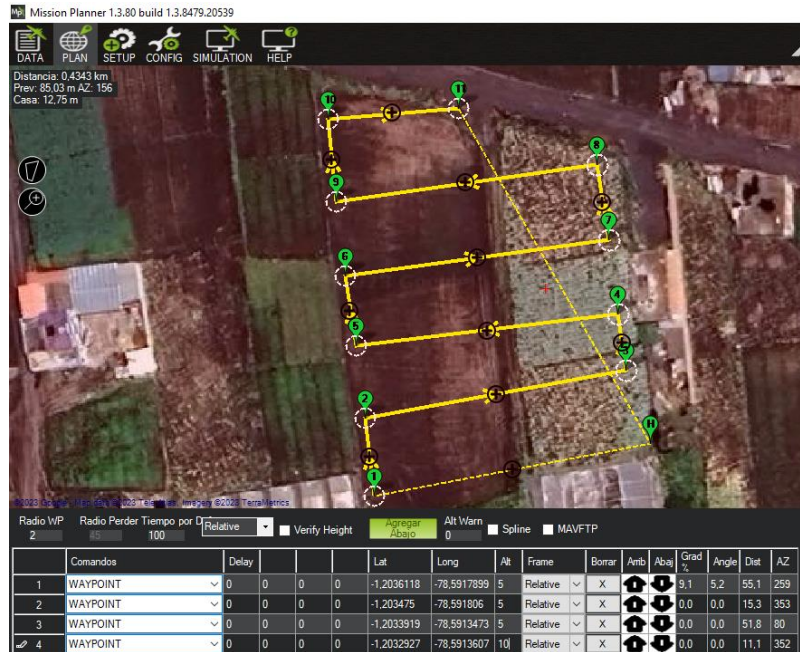


Figura 184. Plan de vuelo totalmente estructurado

Elaborado por el investigador

Estructurado el plan de vuelo en su totalidad y revisado el correcto planteamiento de cada parámetro con sus coordenadas geográficas, se graba el archivo en la placa de control del dron, para dirigirnos al sitio de vuelo seleccionado. Es necesario ubicarse lo más cerca del punto HOME establecido en el plan de vuelo para que el dron ejecute la misión al instante. En la figura 185 se observa la elevación del dron mediante el mando y el modo de vuelo Stabilize, seguido se procede al cambio de modo de vuelo Loiter que nos permite tener mayor estabilidad del dron a una altura definida. Luego de estabilizar el dron, se procede a ejecutar la misión en el modo de vuelo AUTO que se encuentra en la tercera posición del switch del radiocontrol.



Figura 185. Elevación del dron y ejecución de la misión de vuelo

Elaborado por el investigador

En el desarrollo de la misión de vuelo, se estableció una altura de 10 metros en el tramo de los puntos 4 al 5. El dron obedece esta variación de altura como se puede observar en la figura 186, terminado este tramo vuelve a su altura definida para el resto de la ruta que es de 5 metros.



Figura 186. Variación de altura de 10 metros en tramos 4 al 5

Elaborado por el investigador

Luego de recorrer cada uno de los puntos estratégicos definidos en el programa, el dron regresa a su punto HOME mediante la acción definida en el punto 11 como RETURN TO LAUNCH para su retorno seguro. En la figura 187, se observa el instante de arribo del dron y su tiempo de vuelo autónomo empleado en esta ruta que es de 3 minutos con 57 segundo, tiempo aproximado al calculado de 4 minutos para una extensión de terreno de $10000m^2$.

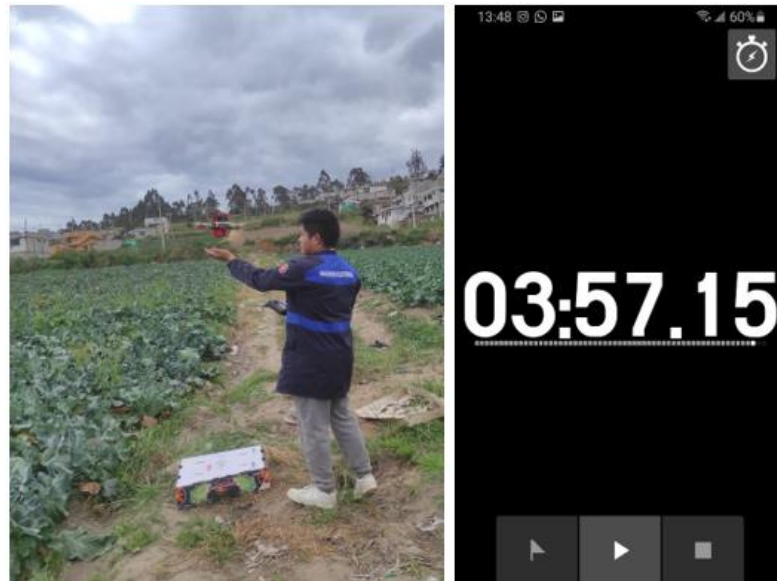


Figura 187. Arribo del dron al punto HOME y tiempo empleado

Elaborado por el investigador

Verificado el tiempo de vuelo calculado con el tiempo empleado en la misión programada a una velocidad de $6m/s$, se definen misiones progresivas con el aumento de la extensión del terreno como se muestra en la tabla 16, las misiones programadas dependen de la duración total de la batería que es de 15 minutos con peso extra en vuelo libre y el alcance de transmisión de señal desde el radio control. El mando utilizado tiene una cobertura de 800 metros a su alrededor sin ningún obstáculo a la vista por ende la inspección de los cultivos tendría una limitante hasta $640000m^2$.

Tabla 16. Relación entre terreno inspeccionado vs tiempo de vuelo

Terreno Inspeccionado	Tiempo de vuelo empleado
$10000m^2$	3.57 minutos
$20000m^2$	7.18 minutos
$40000m^2$	14.33 minutos

Fuente: Elaborado por el investigador

Prueba 2: Evaluación de imágenes capturadas a diferentes alturas para precisar el reconocimiento del algoritmo de visión artificial

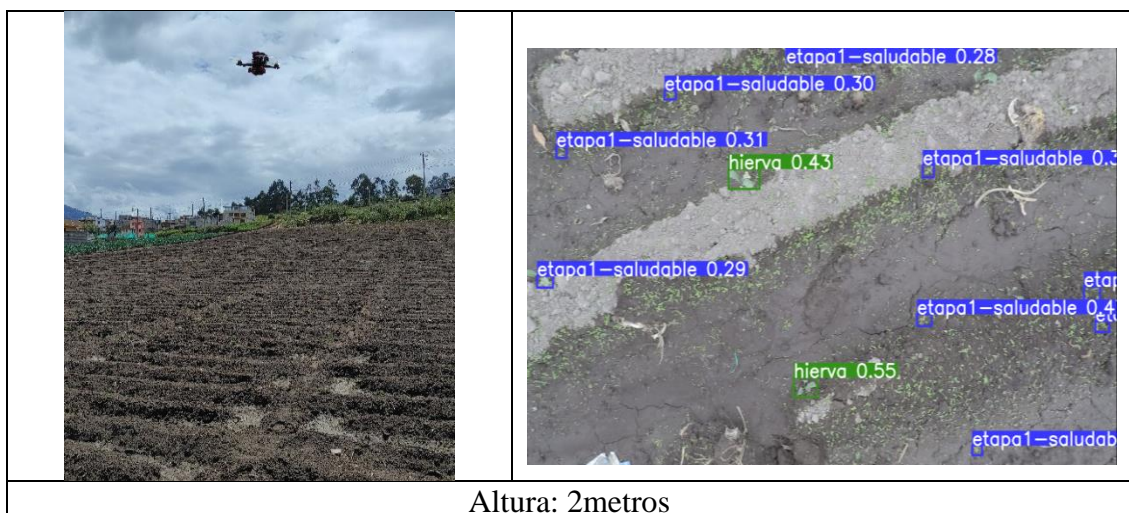
Se evalúa la captura de imágenes a diferentes alturas para determinar la altura propicia y obtener un óptimo análisis en cada etapa del cultivo de brócoli.

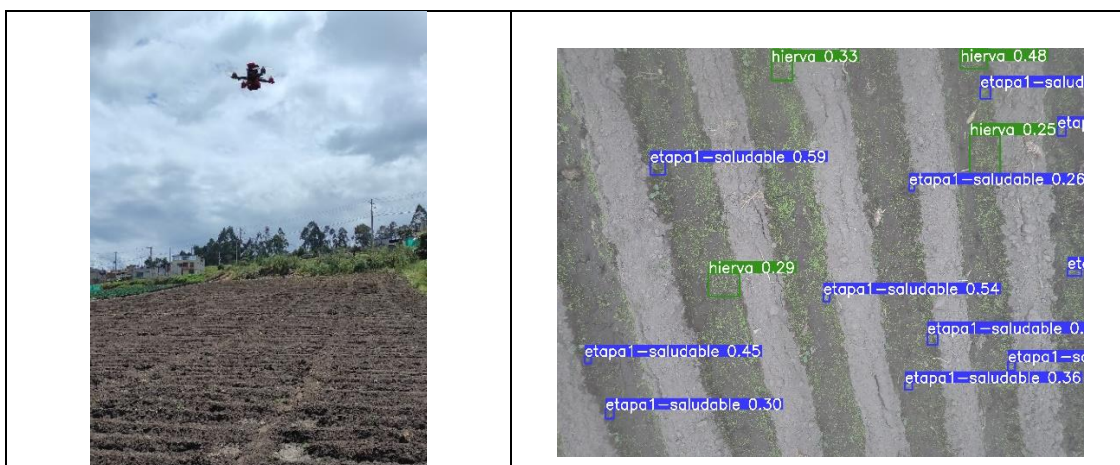
Resultados:

Las imágenes capturadas anteriormente para el entrenamiento del algoritmo de visión artificial fueron tomadas a alturas de 1 a 15 metros. Por esta razón, ya implementado el sistema sobre el dron, se procedió elevarlo a alturas de 2, 3, 5 y 10 metros respectivamente sobre cada etapa de cultivo del brócoli y de esta manera evaluar la altura propicia para la detección de anomalías dentro del terreno cultivado, además durante el vuelo se verifico la estabilidad del dron en modo de vuelo loiter y la trasmisión de los datos recolectados hacia el ordenador. Para la valoración de la altura óptima y la estabilidad presentada por el dron al momento del vuelo se utilizaron los criterios: Malo, Aceptable y Excelente.

En las tablas 17, 19, 21 y 23 se observa el funcionamiento del sistema sobre cada una de las etapas de cultivo del brócoli.

Tabla 17. Etapa 1 de cultivo de brócoli - Trasplante





Altura: 3 metros



Altura: 5 metros



Altura: 10 metros

Fuente: Elaborado por el investigador

La valoración de los resultados en esta primera etapa del cultivo dictamina una altura propicia de 2 metros para el funcionamiento óptimo del sistema de teledetección, como se observa en la tabla 18 esto se debe al tamaño minúsculo de la plántula que es de aproximadamente 4cm en su etapa de trasplante, que a una mayor altura logra

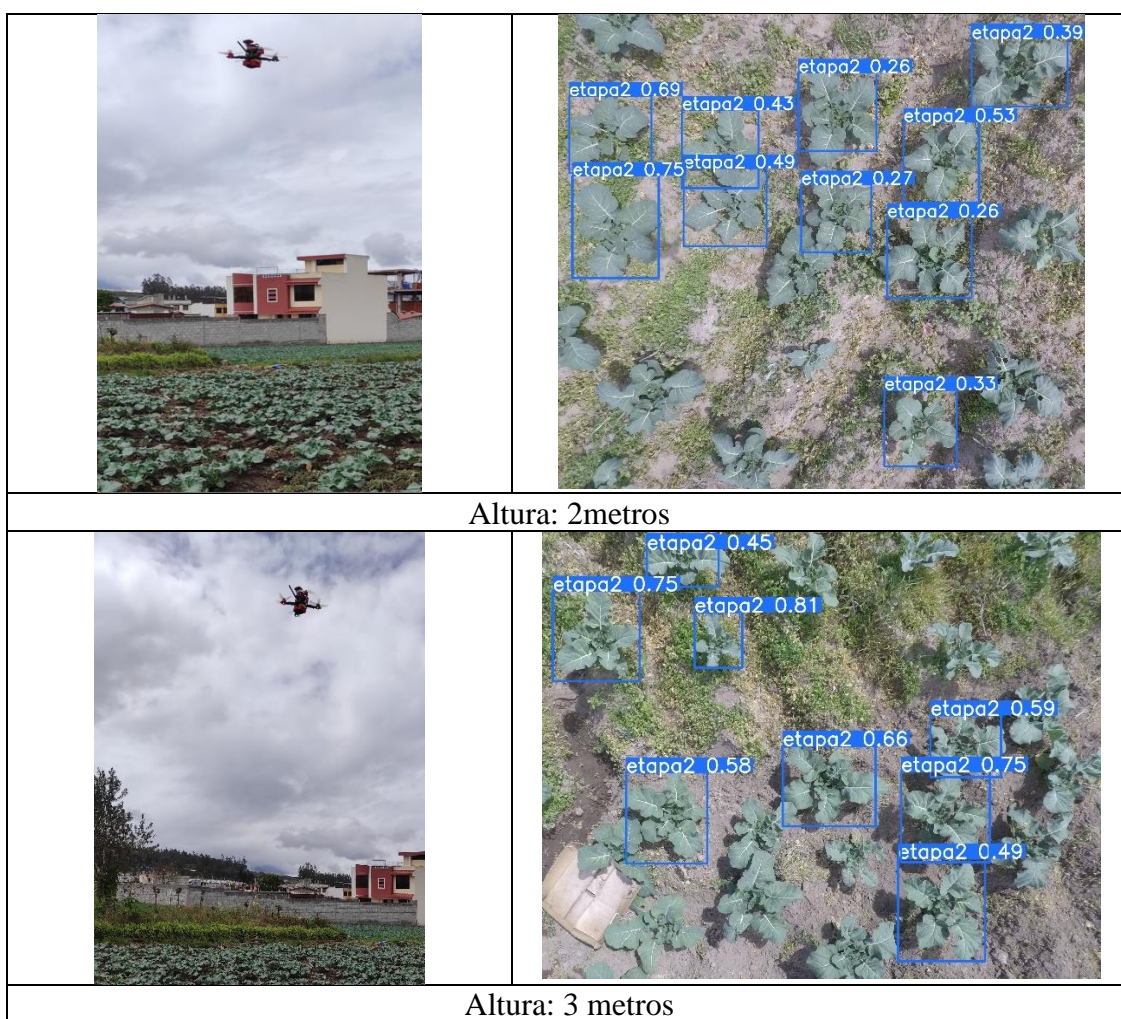
ser confundida como una mala hierba por el algoritmo de visión artificial y obteniendo de esta manera falsos positivos en el análisis.

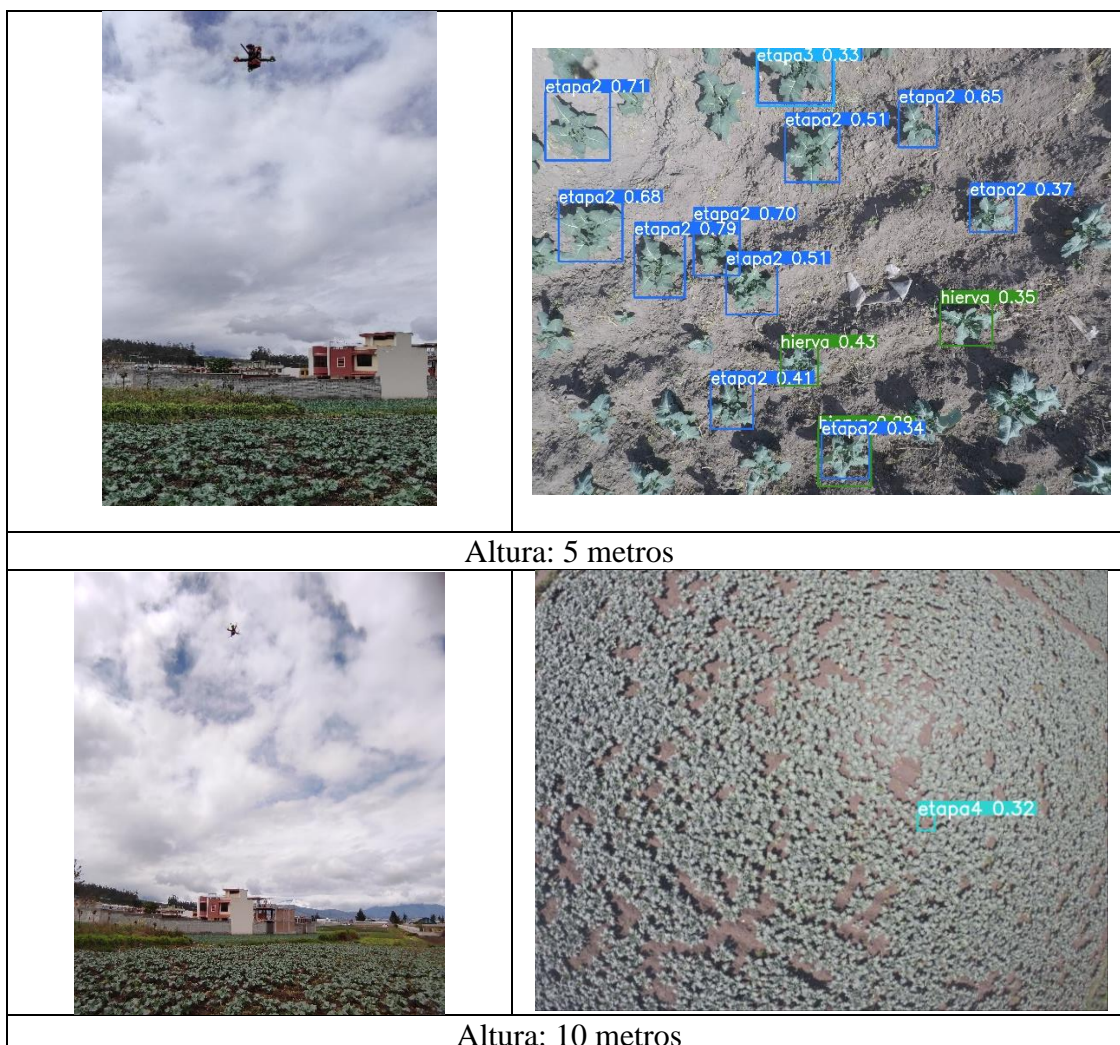
Tabla 18. Valoración de los resultados obtenidos en la etapa 1

Prueba	Altura	Evaluación de calidad
1	2 metros	Excelente
2	3 metros	Aceptable
3	5 metros	Mala
4	10 metros	Mala

Fuente: Elaborado por el investigador

Tabla 19. Etapa 2 de cultivo de brócoli – Inducción floral





Fuente: Elaborado por el investigador




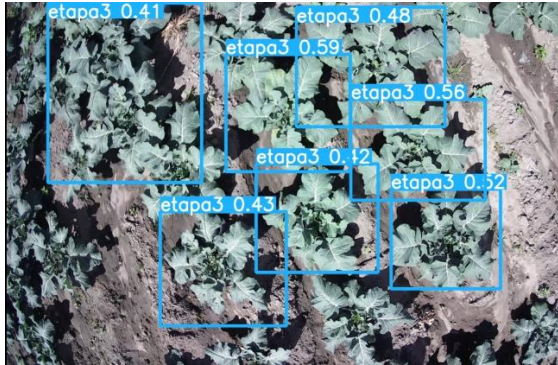

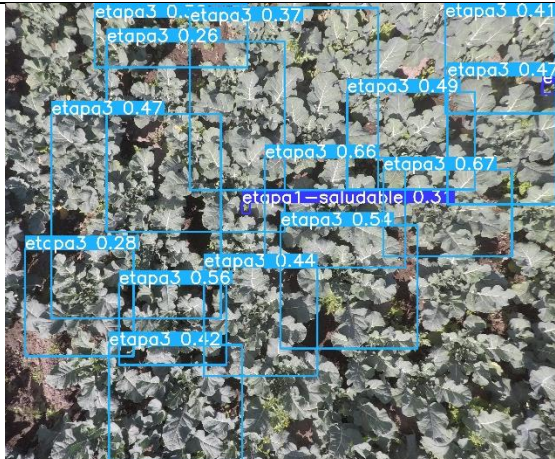
En la etapa 2 de cultivo del brócoli, la planta se encuentra a un tamaño promedio de 10cm obteniendo que a una altura de 3 metros el funcionamiento del sistema es excelente, tabla 20, debido a que el análisis del cultivo es identificado correctamente en cada imagen capturada. A una mayor altura no se pierde la estabilidad de vuelo, pero el sistema si pierde la detección correcta del cultivo, dado que se obtiene lecturas de falsos negativos y falsos positivos durante el proceso.

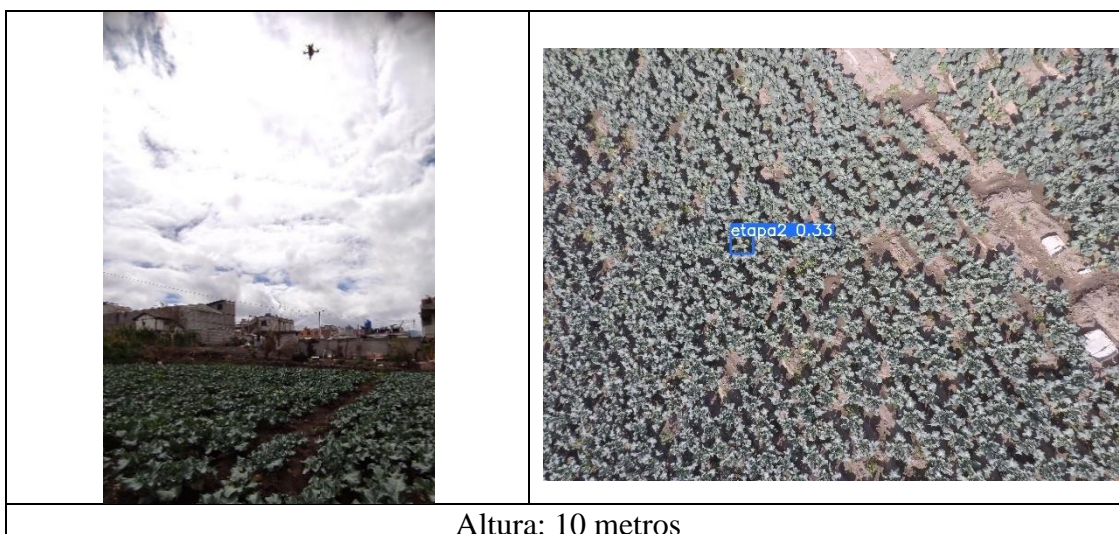
Tabla 20. Valoración de los resultados obtenidos en la etapa 2

Prueba	Altura	Evaluación de calidad
1	2 metros	Excelente
2	3 metros	Excelente
3	5 metros	Aceptable
4	10 metros	Mala

Fuente: Elaborado por el investigador

Tabla 21. Etapa 3 de cultivo de brócoli – Fructificación

	
<p>Altura: 2metros</p>	
	
<p>Altura: 3 metros</p>	
	
<p>Altura: 5 metros</p>	



Altura: 10 metros

Fuente: Elaborado por el investigador

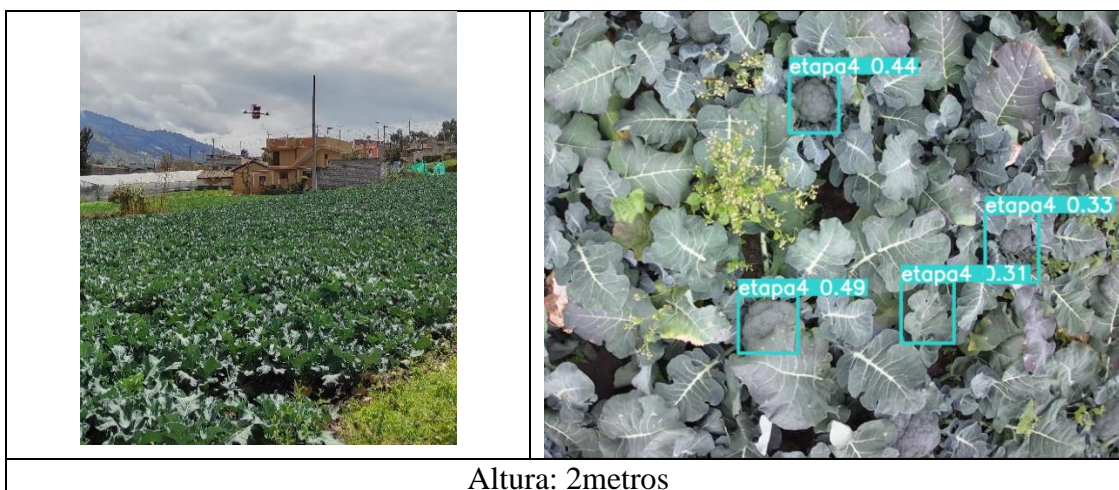
En la tabla 22 se concluye que la altura óptima para la tercera etapa del cultivo es de 5 metros, debido a que el brócoli posee una altura de 45 cm que resulta visible para que el sistema funcione sin falencias y detecte las anomalías dentro de la superficie cultivada. A esta altura la plataforma de transporte para el sistema sigue siendo estable y se encuentran menos lecturas de falsos positivos y falsos negativos.

Tabla 22. Valoración de los resultados obtenidos en la etapa 3

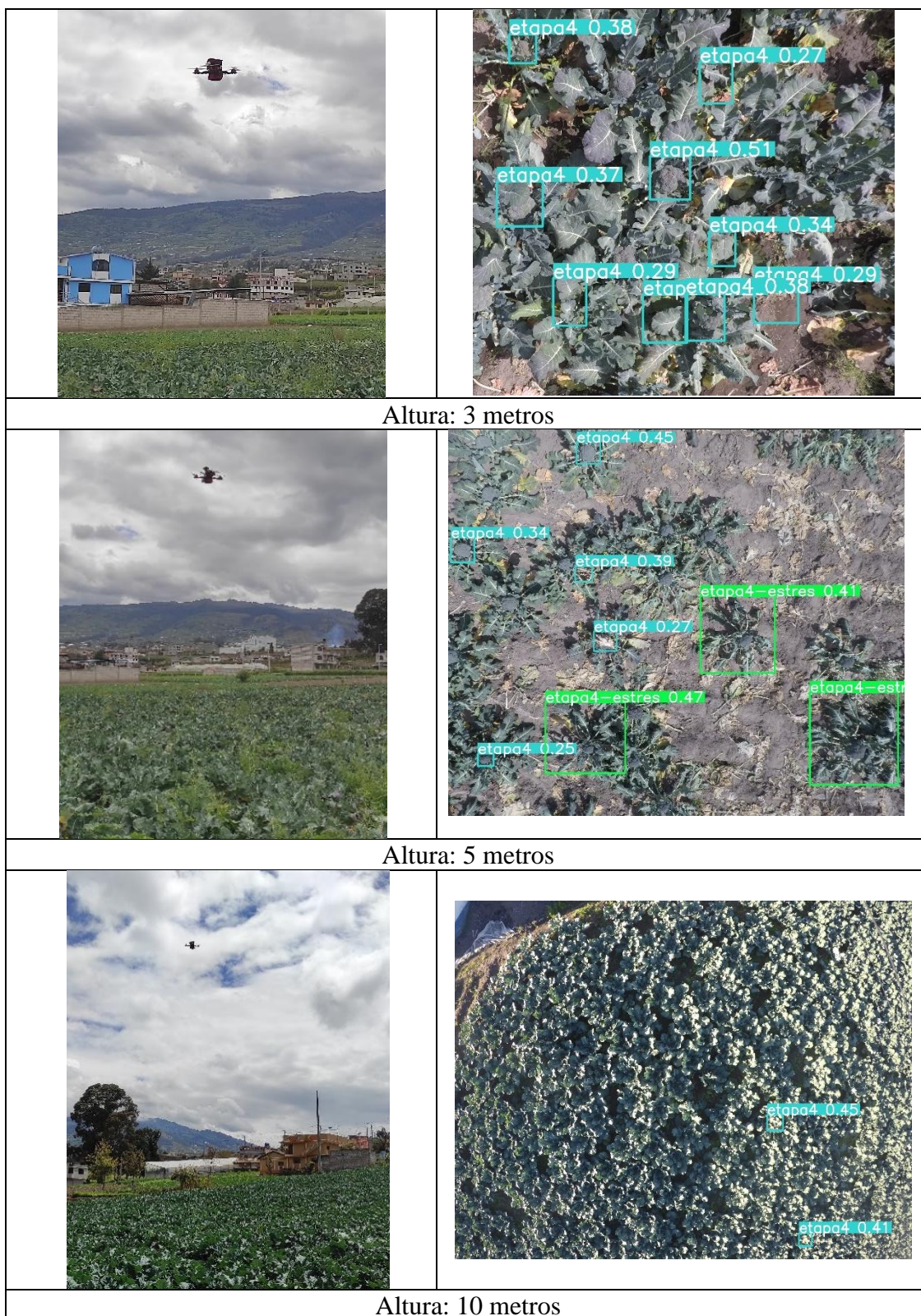
Prueba	Altura	Evaluación de calidad
1	2 metros	Excelente
2	3 metros	Excelente
3	5 metros	Excelente
4	10 metros	Mala

Fuente: Elaborado por el investigador

Tabla 23. Etapa 4 de cultivo de brócoli - Cosecha



Altura: 2metros



Fuente: Elaborado por el investigador

En la última etapa de cultivo se obtiene que la altura ideal para un adecuado funcionamiento del sistema es de 5 metros, tabla 24, gracias al amplio enfoque de visibilidad presente en esta etapa debido a que el cultivo posee un tamaño de 60cm

aproximadamente. La estabilidad de vuelo del dron a esta altura es satisfactoria, al igual que el análisis del cultivo con una baja latencia de lecturas falsas negativas y falsas positivas.

Tabla 24. Valoración de los resultados obtenidos en la etapa 4

Prueba	Altura	Evaluación de calidad
1	2 metros	Excelente
2	3 metros	Excelente
3	5 metros	Excelente
4	10 metros	Mala

Fuente: Elaborado por el investigador

Prueba 3: Evaluación del plan de vuelo en conjunto con el funcionamiento del sistema de teledetección implementado

Se evalúan los diferentes resultados obtenidos en el análisis del cultivo de brócoli tras la inspección realizada sobre las superficies cultivadas.

Resultados:

En primera instancia se desarrollaron los planes de vuelo en el software Mission Planer para cada una de las etapas del cultivo como se muestra en la figura 188. Cada plan de vuelo debe tener un tiempo de autonomía de 4 minutos aproximadamente, una velocidad de 6m/s y una altura acorde a lo concluido en los resultados de la prueba 2, además de una velocidad de viento que ronda los 10 a 15 km/h.



Figura 188. Planes de vuelo programados: A) Etapa 1, B) Etapa 2, C) Etapa 3 y D) Etapa 4

Elaborado por el investigador

Para cada etapa del cultivo, el proceso de teledetección se realiza bajo los parámetros necesarios y el entrenamiento correcto del algoritmo de visión artificial. La base de datos entrenada para el reconocimiento de agentes patógenos dentro de los cultivos de brócoli constó en un inicio de 100 épocas, obteniendo un bajo nivel de precisión y sensibilidad como se observa en la tabla 25, por consiguiente, se procedió a entrenar el algoritmo aumentando a 200 épocas por etapa para obtener un mayor rendimiento dentro de los valores de precisión y sensibilidad sin la presencia masiva de falsos positivos y falsos negativos. En la tabla 26 se muestran los valores obtenidos luego del entrenamiento del algoritmo con 200 épocas.

Tabla 25. Entrenamiento del algoritmo con 100 épocas

Clases	Imágenes	Precisión	Sensibilidad
Primera Etapa	544	0.820	0.880
Segunda Etapa	544	0.866	0.901
Tercera Etapa	544	0.909	0.836
Cuarta Etapa	544	0.913	0.893
Todas	2176	0.877	0.878

Fuente: Elaborado por el investigador

Tabla 26. Entrenamiento del algoritmo con 200 épocas

Clases	Imágenes	Precisión	Sensibilidad
Primera Etapa	544	0.895	0.907
Segunda Etapa	544	0.901	0.915
Tercera Etapa	544	0.922	0.933
Cuarta Etapa	544	0.958	0.941
Todas	2176	0.919	0.932

Fuente: Elaborado por el investigador

Los resultados obtenidos en la última fila de cada tabla hacen referencia a los valores porcentuales del progreso en el entrenamiento del algoritmo de visión artificial YOLO v5x. En las 200 épocas se obtiene una precisión de 91.90% y una sensibilidad de 93.20% dentro del análisis de cada imagen, posicionando al sistema en un nivel alto de precisión, debido a su incremento porcentual tras el aumento de épocas en el entrenamiento del algoritmo. Además, la evaluación realizada para la visión artificial consta de una muestra de la base de datos de 60% para entrenamiento y 40% para validación.

Estructurado cada uno de los planes de vuelo y validado el entrenamiento del algoritmo de visión artificial, se procede al funcionamiento del sistema mediante la

captura de imágenes en un lapso de tiempo de 2 segundos para su respectivo procesamiento y obtención de datos en conjunto con las coordenadas de geolocalización respectivas. Para comprobar el funcionamiento del sistema de geolocalización implementado en el dron, en la tabla 27 se adecuan cada uno de los datos de posicionamiento obtenidos durante cada captura de imagen ejecutada por el sistema de teledetección, esto con el fin de localizar el punto al que pertenecen los resultados obtenidos luego de cada procesamiento de la imagen.

Tabla 27. Datos de geolocalización recolectados al momento de cada captura de imagen

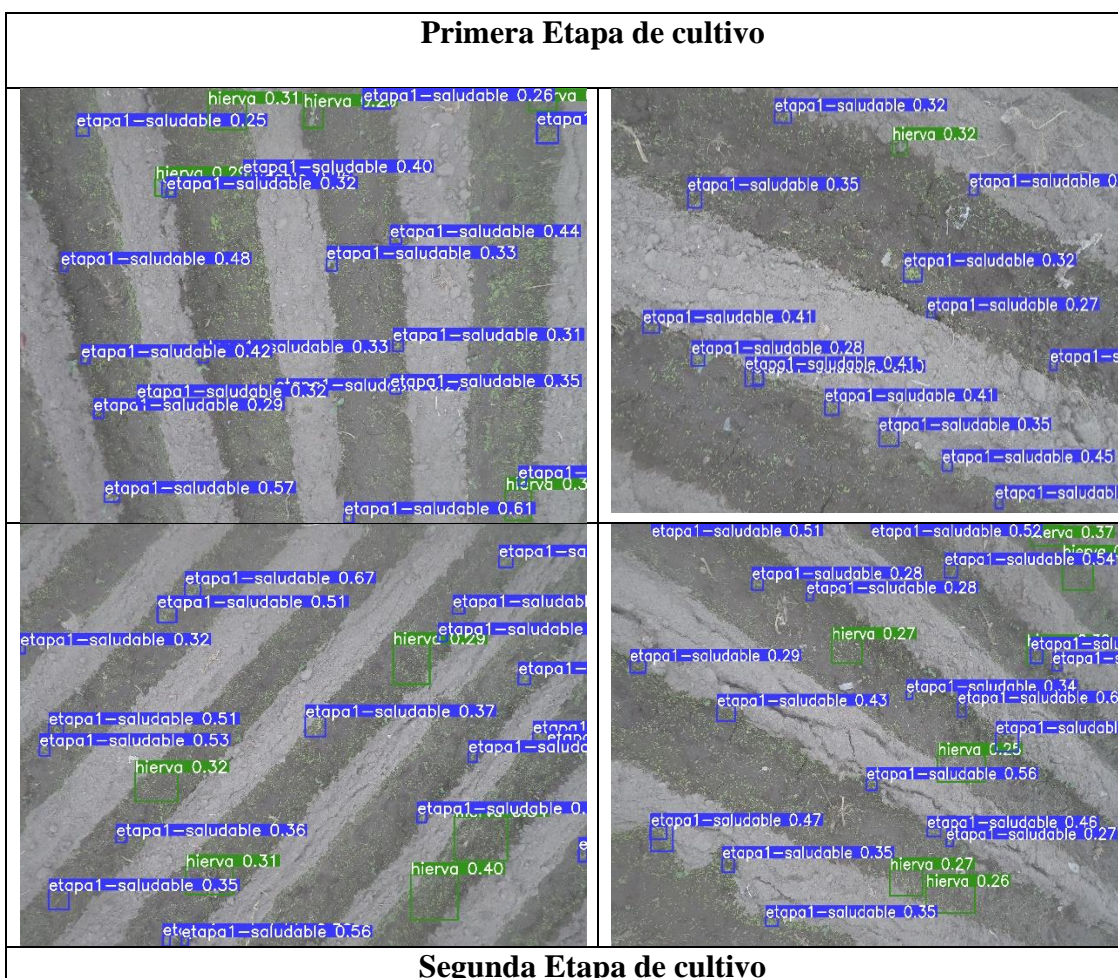
	Etapa 1 Trasplante		Etapa 2 Inducción floral		Etapa 3 Fructificación		Etapa 4 Cosecha	
	Latitud	Longitud	Latitud	Longitud	Latitud	Longitud	Latitud	Longitud
1	-1.2037	-78.5946	-1.2027	-78.5917	-1.2035	-78.5913	-1.2029	-78.5939
2	-1.2037	-78.5946	-1.2027	-78.5918	-1.2035	-78.5913	-1.2028	-78.5939
3	-1.2036	-78.5946	-1.2027	-78.5918	-1.2035	-78.5914	-1.2027	-78.5939
4	-1.2036	-78.5947	-1.2026	-78.5918	-1.2035	-78.5914	-1.2026	-78.5940
5	-1.2036	-78.5946	-1.2027	-78.5917	-1.2034	-78.5915	-1.2026	-78.5939
6	-1.2036	-78.5946	-1.2026	-78.5917	-1.2035	-78.5914	-1.2026	-78.5939
7	-1.2036	-78.5946	-1.2027	-78.5918	-1.2034	-78.5913	-1.2028	-78.5938
8	-1.2037	-78.5946	-1.2026	-78.5917	-1.2035	-78.5913	-1.2029	-78.5938
9	-1.2037	-78.5946	-1.2026	-78.5917	-1.2034	-78.5913	-1.2029	-78.5938
10	-1.2037	-78.5946	-1.2026	-78.5917	-1.2034	-78.5913	-1.2026	-78.5937
11	-1.2036	-78.5946	-1.2027	-78.5917	-1.2034	-78.5913	-1.2028	-78.5937
12	-1.2036	-78.5946	-1.2025	-78.5917	-1.2035	-78.5914	-1.2029	-78.5938
13	-1.2035	-78.5945	-1.2025	-78.5917	-1.2034	-78.5914	-1.2028	-78.5937
14	-1.2036	-78.5945	-1.2025	-78.5919	-1.2033	-78.5915	-1.2028	-78.5937
15	-1.2035	-78.5945	-1.2024	-78.5919	-1.2033	-78.5915	-1.2027	-78.5937
16	-1.2037	-78.5945	-1.2024	-78.5918	-1.2034	-78.5915	-1.2027	-78.5938
17	-1.2036	-78.5946	-1.2024	-78.5917	-1.2034	-78.5915	-1.2027	-78.5938
18	-1.2036	-78.5946	-1.2023	-78.5917	-1.2034	-78.5916	-1.2026	-78.5937
19	-1.2036	-78.5946	-1.2023	-78.5917	-1.2033	-78.5916	-1.2026	-78.5938
20	-1.2035	-78.5945	-1.2023	-78.5918	-1.2033	-78.5916	-1.2026	-78.5938
21	-1.2035	-78.5945	-1.2023	-78.5917	-1.2033	-78.5916	-1.2025	-78.5937
22	-1.2035	-78.5945	-1.2024	-78.5917	-1.2032	-78.5916	-1.2025	-78.5936
23	-1.2036	-78.5944	-1.2024	-78.5917	-1.2033	-78.5915	-1.2027	-78.5935
24	-1.2036	-78.5944	-1.2023	-78.5917	-1.2034	-78.5916	-1.2028	-78.5935
25	-1.2036	-78.5944	-1.2024	-78.5917	-1.2033	-78.5916	-1.2029	-78.5934
26	-1.2036	-78.5944	-1.2023	-78.5918	-1.2032	-78.5915	-1.2027	-78.5934
27	-1.2035	-78.5944	-1.2023	-78.5919	-1.2033	-78.5916	-1.2026	-78.5935
28	-1.2035	-78.5944	-1.2021	-78.5919	-1.2033	-78.5916	-1.2028	-78.5935
29	-1.2035	-78.5944	-1.2021	-78.5918	-1.2032	-78.5916	-1.2026	-78.5935
30	-1.2035	-78.5943	-1.2021	-78.5917	-1.2032	-78.5914	-1.2025	-78.5934

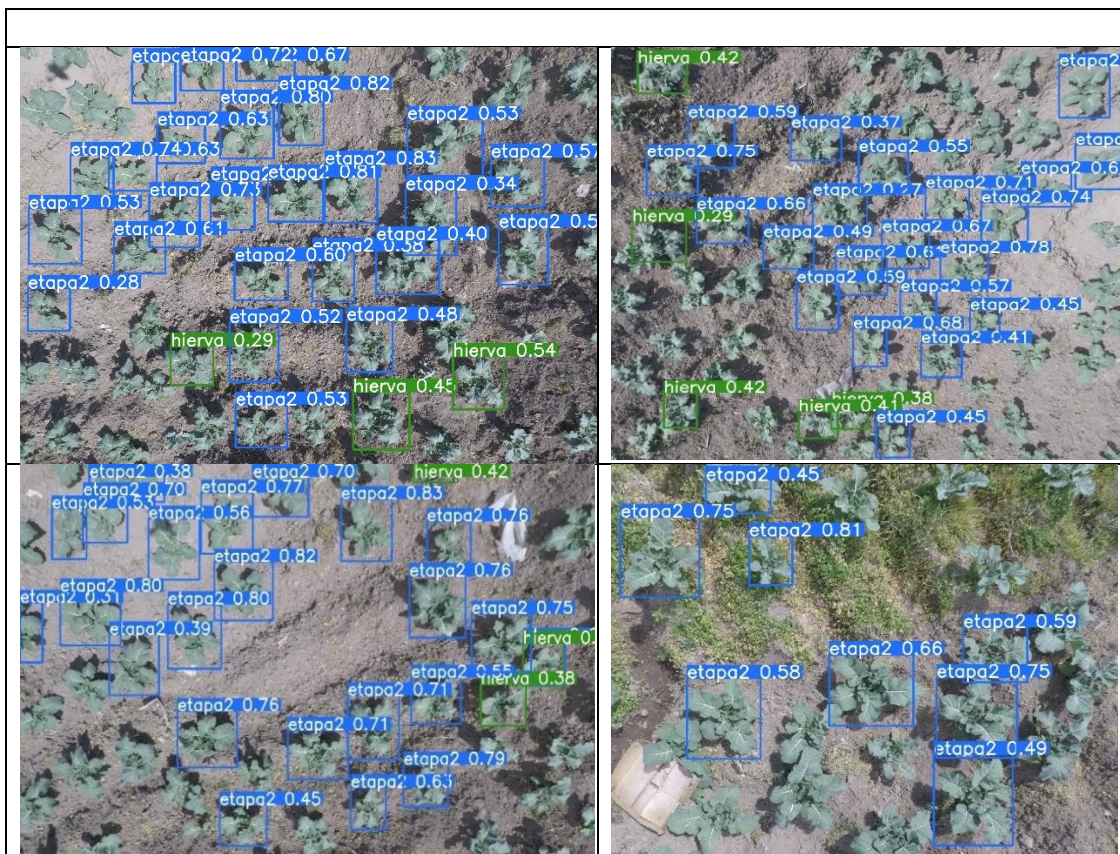
31	-1.2036	-78.5943	-1.2020	-78.5917	-1.2031	-78.5913	-1.2025	-78.5935
32	-1.2036	-78.5943	-1.2020	-78.5918	-1.2031	-78.5913	-1.2024	-78.5933
33	-1.2036	-78.5943	-1.2019	-78.5919	-1.2031	-78.5913	-1.2026	-78.5932
34	-1.2034	-78.5942	-1.2019	-78.5919	-1.2031	-78.5914	-1.2025	-78.5932
35	-1.2035	-78.5942	-1.2019	-78.5919	-1.2031	-78.5914	-1.2024	-78.5932
36	-1.2035	-78.5942	-1.2019	-78.5918	-1.2030	-78.5915	-1.2023	-78.5931
37	-1.2035	-78.5941	-1.2019	-78.5918	-1.2030	-78.5915	-1.2023	-78.5931
38	-1.2035	-78.5941	-1.2019	-78.5919	-1.2030	-78.5915	-1.2025	-78.5930
39	-1.2034	-78.5941	-1.2018	-78.5920	-1.2029	-78.5914	-1.2023	-78.5930
40	-1.2034	-78.5940	-1.2018	-78.5919	-1.2030	-78.5914	-1.2026	-78.5930
41	-1.2034	-78.5940	-1.2019	-78.5918	-1.2030	-78.5913	-1.2027	-78.5930
42	-1.2034	-78.5940	-1.2019	-78.5917	-1.2030	-78.5915	-1.2029	-78.5930

Fuente: Elaborado por el investigador

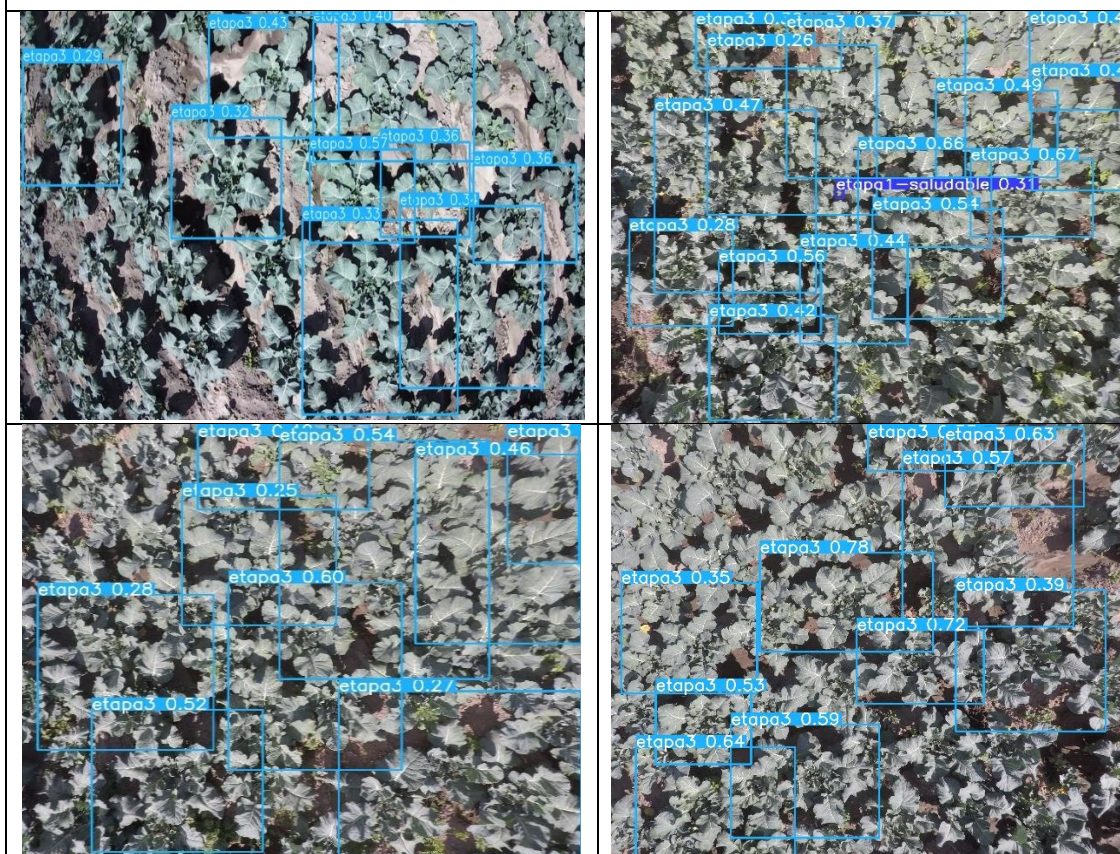
La evaluación para comprobar la detección de agentes patógenos dentro de los cultivos, parte del porcentaje de precisión obtenido en el entrenamiento de visión artificial, que en este caso fue de 91.90%. En la tabla 28 se muestran el estado de salud y anomalías encontradas en cada una de las etapas del cultivo.

Tabla 28. Resultados de detección obtenidos en cada etapa del cultivo

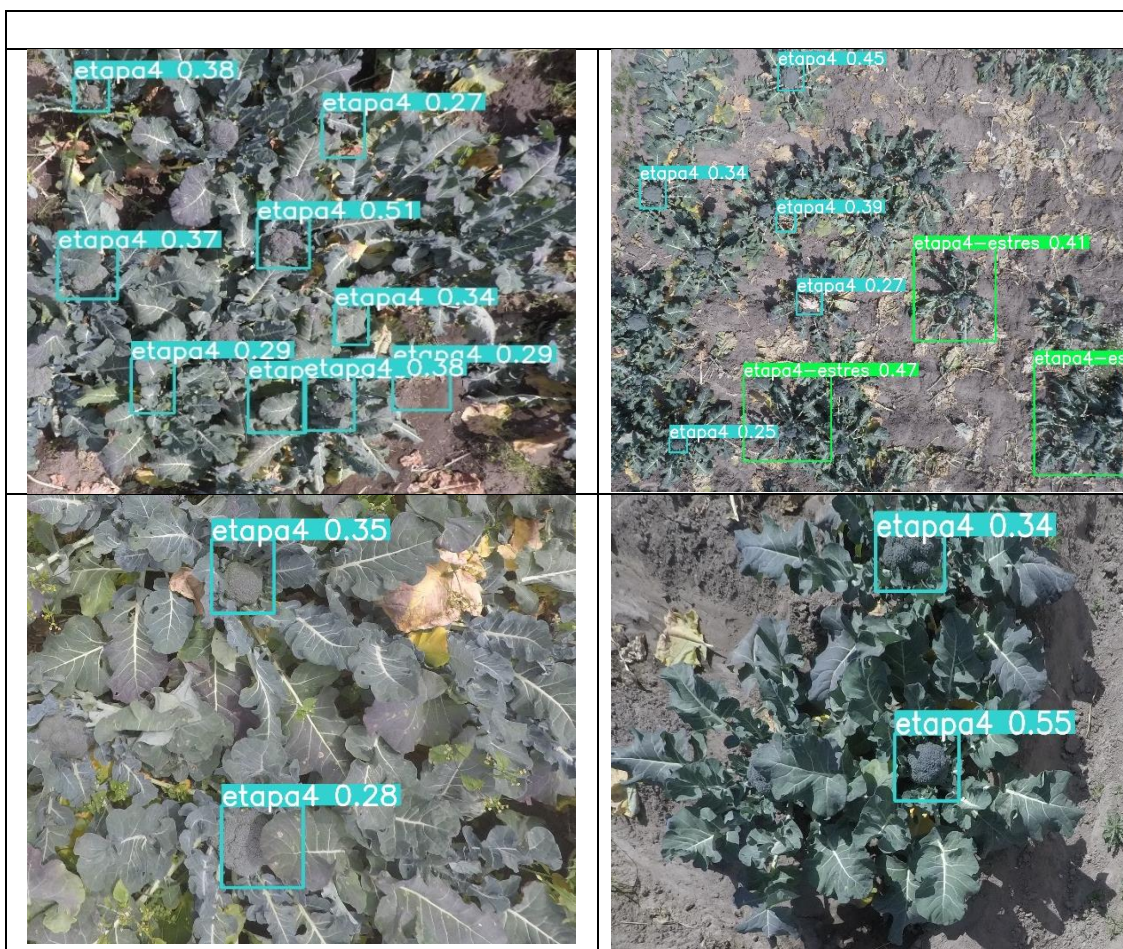




Tercera Etapa de cultivo



Cuarta Etapa de cultivo



Fuente: Elaborado por el investigador

Matriz de confusión

La matriz de confusión hace referencia a los falsos negativos y falsos positivos encontrados durante el rendimiento del algoritmo de visión artificial YOLO v5x. Esta matriz de confusión se desarrolla para cada una de las etapas presentes en el cultivo de brócoli a través de Google Colaboraty, debido a que el entrenamiento y renderización de cada uno de los datos extraídos de las imágenes se encuentran grabados en esta página web. En la figura 189 se puede observar el valor porcentual del número de aciertos efectuados en la validación de cada etiqueta dentro del algoritmo de visión artificial, así como también se muestra el porcentaje de las falencias obtenidas al momento de ejecutar el algoritmo para el reconocimiento de las etiquetas.

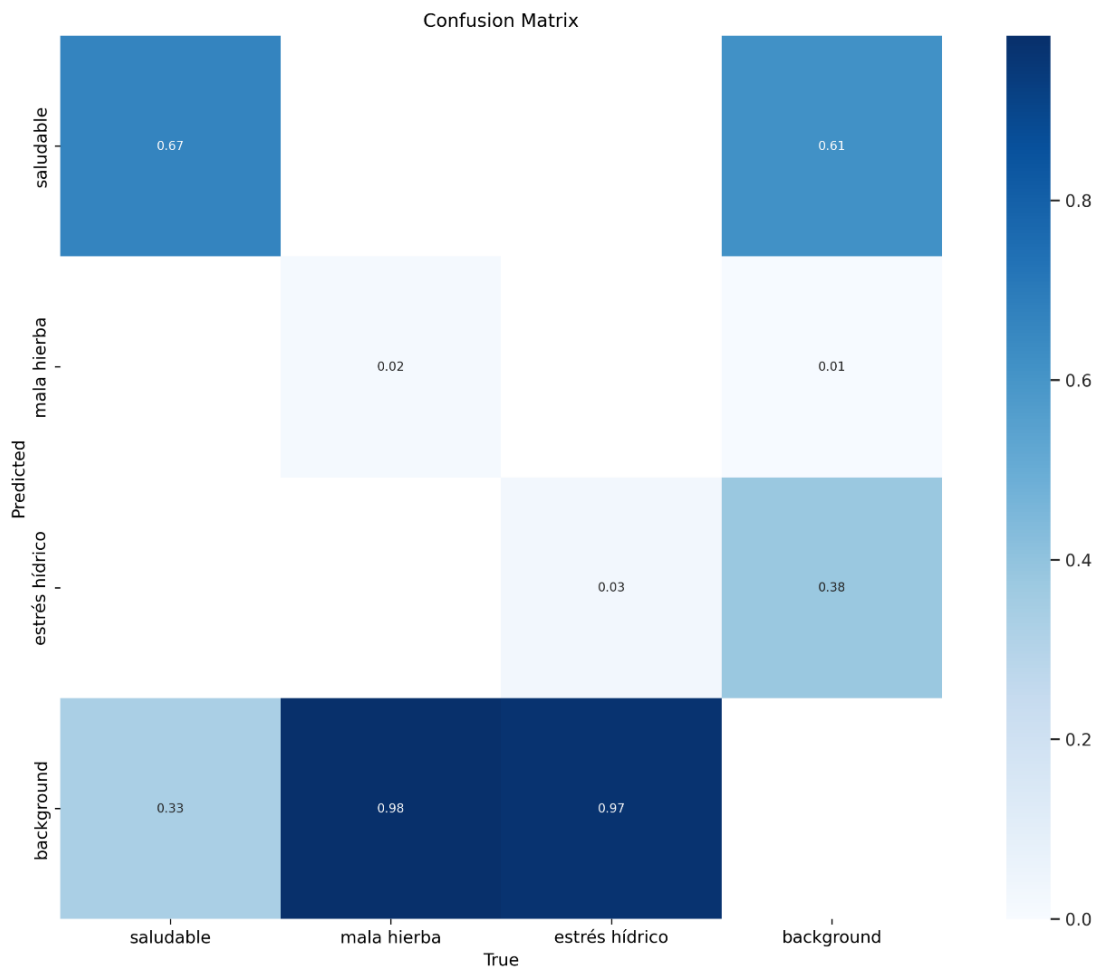


Figura 189. Matriz de confusión: Background horizontal – Falsos positivos y Background vertical – Falsos negativos

Elaborado por el investigador

- **Muestras positivas**

Con un nivel de precisión de 91.90% en el entrenamiento del algoritmo de visión artificial, se obtuvo resultados satisfactorios acorde a la detección de agentes patógenos y enfermedades en cultivos de brócoli, en la figura 190 se muestran los resultados idóneos para cada una de las cuatro etapas en el desarrollo del cultivo.

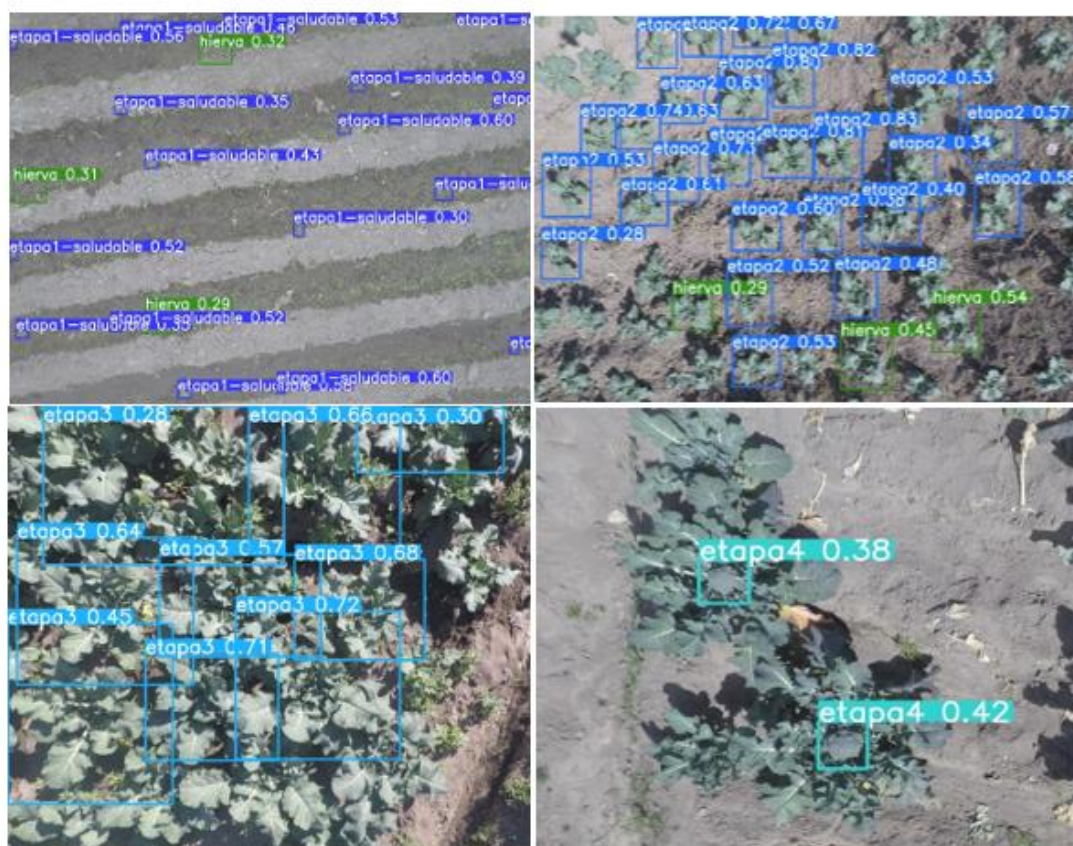


Figura 190. Muestras positivas en el sistema

Elaborado por el investigador

- **Muestras negativas**

Las muestras negativas dentro del entrenamiento del algoritmo se presentan en un 8.10%. Este porcentaje de errores se puede mejorar mediante el aumento de la base de datos para el entrenamiento del algoritmo, en la figura 191 se muestran los errores presentes en cada una de las etapas del cultivo, como la detección de dos etiquetas sobre la misma anomalía.

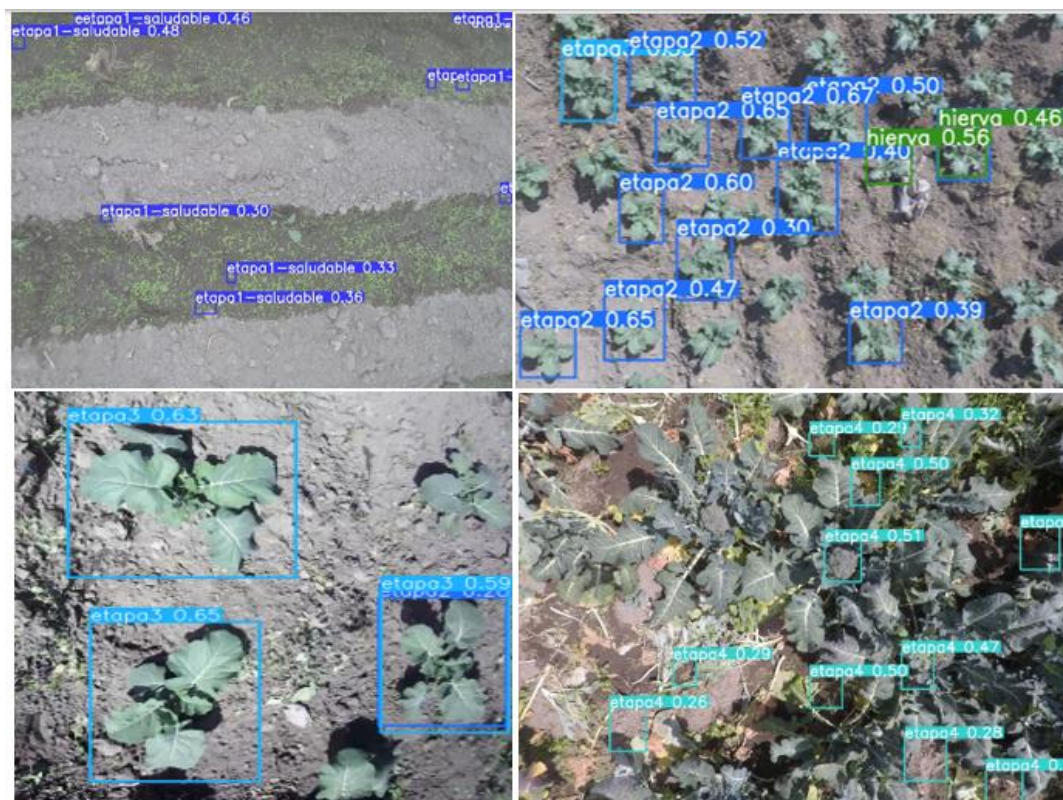


Figura 191. Muestras negativas en el sistema

Elaborado por el investigador

3.2.9 Confiabilidad del prototipo

La confiabilidad para el prototipo parte del entrenamiento y respuesta de precisión obtenidos por el algoritmo de visión artificial YOLO v5x aplicado a la detección de agentes patógenos y enfermedades en cultivos de brócoli. En la validación del algoritmo de visión artificial se concluye que el sistema consta de un 91.90% de precisión de anomalías dentro de los cultivos, adecuando de esta manera al prototipo con una confiabilidad alta dentro de las investigaciones realizadas en el campo de la Inteligencia Artificial aplicada a la agricultura de precisión, por el ejemplo el entramiento de la red neuronal Feed Forward con 200 épocas en cada una de las etapas obtuvieron una precisión en fase de validación de 91.67%, que en comparación al prototipo implementado y entrenado mediante el algoritmo de visión artificial YOLO v5x con una red neuronal convolucional, es menor al índice porcentual de precisión obtenido.

Por otra parte, para el sistema de geolocalización implementado en el dron se procede hacer una comparativa entre los datos de posicionamiento obtenidos por el

prototipo y la cámara GoPro Hero 5 Black que cuenta con su propio sistema GPS integrado en el dispositivo. En la tabla 29 se puede observar los datos de posicionamiento recolectados en la etapa 3 por el sistema y la cámara GoPro.

Tabla 29. Comparación de datos de posicionamiento obtenidos por el prototipo y el GPS de cámara GoPro

	Etapa 3 Fructificación		GPS Cámara GoPro	
	Latitud	Longitud	Latitud	Longitud
1	-1.20349	-78.59133	-1.2034	-78.5913
2	-1.20352	-78.59137	-1.2035	-78.5917
3	-1.20351	-78.59144	-1.2035	-78.5914
4	-1.20353	-78.59142	-1.2035	-78.5914
5	-1.20345	-78.59153	-1.2034	-78.5915
6	-1.20352	-78.59149	-1.2035	-78.5914
7	-1.20341	-78.59136	-1.2034	-78.5913
8	-1.20332	-78.59131	-1.2033	-78.5913
9	-1.20314	-78.59132	-1.2031	-78.5913
10	-1.20343	-78.59133	-1.2034	-78.5913
11	-1.20322	-78.59131	-1.2032	-78.5913
12	-1.20331	-78.59144	-1.2031	-78.5914
13	-1.20344	-78.59145	-1.2344	-78.5914
14	-1.20331	-78.59152	-1.2033	-78.5915
15	-1.20332	-78.59153	-1.2033	-78.5915
16	-1.20343	-78.59151	-1.2034	-78.5915
17	-1.20342	-78.59154	-1.2034	-78.5911
18	-1.20343	-78.59165	-1.2034	-78.5916
19	-1.20331	-78.59166	-1.2031	-78.5916
20	-1.20332	-78.59163	-1.2033	-78.5916

Fuente: Elaborado por el investigador

La información de posicionamiento recolectada por el prototipo presenta un mayor número de decimales en su lectura, lo que aumenta el rango de precisión dentro de la superficie del terreno, mientras que el GPS de la cámara GoPro arroja datos de lectura con tan solo 3 decimales, lo que provoca el desplazamiento de coordenadas con respecto a la ubicación actual. A continuación, se presenta el error relativo porcentual para los datos de longitud y latitud obtenidos por cada dispositivo:

- **Latitud**

$$Error\ relativo\ porcentual = \left(\frac{|1.20349 - 1.2034|}{1.2034} * 100 \right) = \pm 0.074\%$$

- **Longitud**

$$\text{Error relativo porcentual} = \left(\frac{|78.59133 - 78.5913|}{78.5913} * 100 \right) = \pm 0.038\%$$

Mediante los resultados del error relativo porcentual se puede determinar que el posicionamiento del prototipo es más preciso dentro de la ubicación generada en la superficie del terreno a comparación de la posición presentada por la cámara GoPro que se desplaza aproximadamente 3 metros de la posición correcta, figura 192.

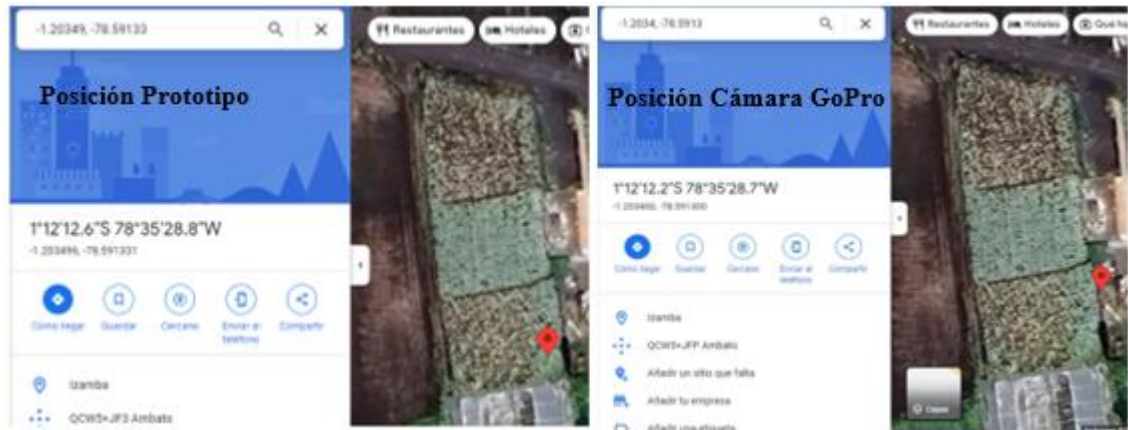


Figura 192. Comparación de coordenadas de Posicionamiento

Elaborado por el investigador

3.2.10 Presupuesto

Los costos que se efectuaron en la realización del proyecto de investigación son financiados por el investigador en su totalidad. El presupuesto invertido en la implementación del sistema de teledetección se presenta en la tabla 30 con su cantidad y valores detallados respectivamente.

Tabla 30. Presupuesto material

Ítem	Elemento	Unidad	Cantidad	Precio Unitario	Precio Total
1	Dron cuadricóptero	Kit	1	250.00	250.00
2	Cámara GoPro Hero 5 Black	c/u	1	220.00	220.00
3	Raspberry Pi 4	c/u	1	120.00	120.00
4	Tarjeta GPRS SIM 808	Kit	1	25.00	25.00
5	Mobile Wifi HUAWEI E586	c/u	1	45.00	45.00
6	Batería 3.7V 6800mAh	c/u	2	6.00	12.00
7	Módulo de carga y descarga de baterías	c/u	1	1.50	1.50
8	Porta Pilas	c/u	1	2.50	2.50

9	Modulo GPS	c/u	1	20.00	20.00
10	Cable USB-C	c/u	1	2.00	2.00
11	Cable micro USB	c/u	2	2.00	4.00
12	Impresión 3D de soporte para cámara	c/u	1	7.00	7.00
13	Baquelita 7x10 cm	c/u	1	2.00	2.00
14	Estaño	rollo	1	3.00	3.00
15	Cautín	c/u	1	2.50	2.50
16	Acido	c/u	1	0.75	0.75
17	Hojas termotransferibles	c/u	1	0.50	0.50
18	Impresión de pistas	c/u	1	0.25	0.25
19	Flash Memory	c/u	1	5.00	5.00
20	Micro SD	c/u	1	6.00	6.00
21	Chip	c/u	1	3.00	3.00
22	Plan celular	c/u	1	8.00	8.00
23	Juego de hélices	kit	5	1.00	5.00
25	Transporte	c/u	48	0.30	14.40
26	Almuerzo	c/u	16	2.50	40.00
SUBTOTAL					793.40
IMPREVISTOS (5%)					40.53
TOTAL					833.93

Fuente: Elaborado por el investigador

En la implementación se tomó en cuenta las horas empleadas en investigación, partiendo de un sueldo de \$858 mensuales que se considera el sueldo promedio de un ingeniero en Telecomunicaciones según el Ministerio del Trabajo de Ecuador. Para calcular el tiempo de investigación empleado se procede de la siguiente manera:

$$\text{Presupuesto investigación}_{\text{implementación}} = \text{Salario mensual} \times \text{tiempo}$$

$$\text{Presupuesto investigación}_{\text{implementación}} = \$858 \times 4 \text{ meses}$$

$$\text{Presupuesto investigación}_{\text{implementación}} = \$3432$$

En conclusión, el costo total invertido en la realización del proyecto parte de los elementos electrónicos y el tiempo invertido en la investigación del mismo, como se detalla en la tabla 31.

Tabla 31 Presupuesto total

Ítem	Descripción	Unidad	Cantidad	Precio Unitario	Precio Total
1	Implementación material	c/u	1	833.93	833.93
2	Tiempo de investigación	c/u	4	858	3432
Total					4265.93

Fuente: Elaborado por el investigador

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- Los drones tienen una amplia variedad de aplicaciones dentro de distintos campos, siendo uno de estos la agricultura que mediante su intervención se desarrolla la incorporación de una agricultura tecnificada. Los drones dedicados netamente al tratamiento e inspección de cultivos mediante tecnología y herramientas avanzadas necesitan adecuarse a ciertas normativas presentadas por la Dirección General de Aviación Civil en el Reglamento de Operación de Aeronaves Pilotadas a Distancia (RPAs) como la altura máxima de vuelo de 400 pies o 122 metros, además de no poder pilotar dentro de zonas pobladas como ciudades, entidades públicas o privadas y zonas utilizadas por miembros de equipos de emergencia o auxilio, estas condiciones aplican para aeronaves cuyo peso este en el rango de 0.25 kilogramos a 150 kilogramos.
- Los componentes presentes en la implementación de un dron varían según el campo al que van dirigidos. En la agricultura los drones se diferencian por componentes como cámaras multiespectrales empleadas en el tratamiento de imágenes para determinar el estado de salud en los cultivos, pulverizadores de fungicidas para el control de agentes contaminantes en el cultivo mediante el uso de inteligencia artificial y radares para la evasión de obstáculos en la ejecución de planes de vuelo programados. Por otro lado, el manejo de estos drones y de su mayoría depende de ciertas variables prevuelo como las normas dictaminadas a nivel nacional para el manejo de estas aeronaves, las condiciones meteorológicas (vientos mayores a 10 m/s, presencia de lluvia), inspección del campo de vuelo, identificación de peligros u obstáculos, espacio de almacenamiento en la memoria digital, verificación de carga completa en las baterías y variables postvuelo tales como descarga de datos recolectados, revisión completa del dron (hélices, nivel de batería, cámara) y un apagado seguro.
- Se construyó el dron Realacc x210Pro FPV de tipo cuadricóptero con las características necesarias para la inspección de cultivos, sus dimensiones son de 300x260x110 mm, un peso de 98 gramos, un peso de carga de 500 gramos

y un tiempo de autonomía en el vuelo de 14.33 minutos para cubrir una extensión de terreno de hasta $40000m^2$. Este dron autónomo consta de accesorios adicionales tales como una cámara GoPro Hero 5 para la obtención de imágenes aéreas de alta definición y un módulo AP.25E GPS/GALILEO que hace posible la autonomía del dron al ejecutar planes de vuelo preprogramados en el software Mission Planer de ArduPilot.

- Se implementó el algoritmo de visión artificial YOLO v5x, debido a su amplia utilidad dentro de proyectos académicos involucrados en el reconocimiento e identificación de objetos. Este algoritmo tiende a ser muy preciso tras un largo entrenamiento y una amplia base de datos existente o recolectada por el investigador. YOLO v5x se entrenó para la detección de agentes patógenos y enfermedades en cultivos de brócoli, detectando de esta manera niveles de color en hojas, tamaños inadecuados de la planta dentro de cada etapa de crecimiento y anomalías en la pella o fruto del cultivo, además, el modelo permite obtener una visión global más precisa de la imagen, minimizando los posibles errores y mejorando la capacidad de reconocimiento de los objetos en cuestión.
- Se implementó el sistema de teledetección para el control de enfermedades y agentes patógenos dentro de los cultivos de brócoli con el uso del algoritmo de visión artificial YOLO v5x obteniendo una precisión de detección del 91.90%, este algoritmo se evaluó mediante la herramienta clearml que permite la mejora dentro de los niveles de entrenamiento de visión artificial. Los resultados obtenidos podrían ser mejores con una base de datos más grande que proporcione información gráfica de los agentes patógenos o enfermedad presente en el cultivo de brócoli con distintos ángulos visión y diferentes alturas, dado que con un entrenamiento medio del algoritmo de visión artificial se pueden producir errores o confusiones dentro de la detección. La implementación del sistema obtuvo gran nivel de precisión en la detección de agentes patógenos para cultivos de brócoli dentro de las investigaciones desarrolladas en el ámbito de la agricultura tecnificada con el uso de inteligencia artificial, como el entrenamiento de la red neuronal Feed-Forward a 200 épocas obteniendo una fase de precisión de 91.67%, además

de tener la aprobación de un ingeniero agrónomo que asesoró y verificó la funcionalidad del prototipo.

4.2. Recomendaciones

- En la construcción del sistema eléctrico del dron es recomendable realizar una conexión piloto de todos los elementos para comprobar su funcionalidad y operatividad antes de adecuarlo al frame o cuerpo de la aeronave.
- Al implementar el módulo AP.25E GPS/GALILEO, se recomienda ubicarlo en la parte superior del dron libre de elementos que se interpongan en la recepción de las señales satelitales, para la correcta ejecución en los vuelos programados.
- Para el entrenamiento de la visión artificial, se recomienda almacenar una gran cantidad de imágenes procesadas en la base de datos para un reconocimiento exitoso al momento de ejecutar el sistema de teledetección.
- Para trabajos de investigación futuros sería recomendable extender el área el trabajo y tiempo de vuelo del dron, mediante el uso de baterías de mayor capacidad, esto quiere decir un mayor tamaño y mayor peso en la fuente de alimentación.
- Antes de ejecutar el sistema de teledetección en el dron es necesario revisar que el nivel de batería de cada uno de los componentes este complementado cargado, para que no existan inconvenientes durante la inspección de las superficies cultivadas.

BIBLIOGRAFÍA

- [1] A. Bula, “Importancia-de-la-agricultura-en-el-desarrollo-socio-económico,” Aug. 2020, Accessed: May 14, 2023. [Online]. Available: <https://observatorio.unr.edu.ar/wp-content/uploads/2020/08/Importancia-de-la-agricultura-en-el-desarrollo-socio-econ%C3%B3mico.pdf>
- [2] M. Garcia, “ESTUDIO REVELA QUE LA PRODUCTIVIDAD AGRÍCOLA DEBE AUMENTAR 1.75 % CADA AÑO PARA ALIMENTAR EL MUNDO EN EL 2050,” *IICA*, 2020. <https://iica.int/es/prensa/noticias/estudio-revela-que-la-productividad-agr%25C3%25ADcola-debe-aumentar-175-cada-a%25C3%25B1o-para#> (accessed May 14, 2023).
- [3] L. Salazar, M. Schling, A. C. Palacios, and N. Pazos, “RETOS PARA LA AGRICULTURA FAMILIAR EN EL CONTEXTO DEL COVID-19,” 2020.
- [4] Somos Agricultura, “Estadísticas mundiales de producción de brócoli,” Nov. 15, 2022. <https://blogagricultura.com/estadisticas-brocoli-produccion/> (accessed May 14, 2023).
- [5] J. García and R. Jinson, “DETERMINACIÓN DE LOS REQUERIMIENTOS HÍDRICOS DEL CULTIVO DE BRÓCOLI (*Brassica oleracea*. L. Var. Avenger) BAJO CONDICIONES EDAFOCLIMÁTICAS DEL CANTÓN RIOBAMBA PROVINCIA DE CHIMBORAZO,” Riobamba, 2016.
- [6] G. Coba, “El brócoli gana protagonismo en la exportación no petrolera del país,” *Revista PRIMICIAS*, Sep. 02, 2020. <https://www.primicias.ec/noticias/economia/brocoli-espacio-canasta-exportaciones/> (accessed May 14, 2023).
- [7] C. Enriquez, “El brócoli mantiene una alta demanda externa ,” *Revista LIDERES*, Mar. 13, 2020. <https://www.revistalideres.ec/lideres/brocoli-mantiene-alta-demanda-externa.html> (accessed May 14, 2023).
- [8] A. Sanchez, T. Vayas, F. Mayorga, and C. Freire, “PRODUCCIÓN DE

- BRÓCOLI EN ECUADOR,” Ambato, 2019.
- [9] B. Manzano, “BOLETIN SITUACIONAL CULTIVO DE BROCOLI,” Quito, May 2022. [Online]. Available: www.agricultura.gob.ec
- [10] R. Bautista, P. Constante, A. Gordon, and D. Mendoza, “Diseño e implementación de un sistema de visión artificial para análisis de datos NDVI en imágenes espectrales de cultivos de brócoli obtenidos mediante una aeronave pilotada remotamente Design and implementation of an artificial vision system for NDVI data analysis in specific images of broccoli crops using a remotely piloted aircraft,” Latacunga, Dec. 2018. Accessed: May 14, 2023. [Online]. Available: <https://journal.espe.edu.ec/ojs/index.php/Infociencia/article/view/1230/pdf>
- [11] R. Por, F. Fouquet, C. Tutorizado, and F. López Valverde, “Inteligencia Artificial aplicada a la agricultura de precisión. Control de hongos en la planta de tomate Artificial Intelligence applied to Precision Agriculture. Control of fungi in the tomato plant,” Málaga, Sep. 2021.
- [12] E. Román and Ruiz Maria de Fatima, “Detección de macronutrientes y enfermedades en campos de cultivo de banano orgánico con Machine Learning,” *Universidad de Piura*, Feb. 2021.
- [13] B. Roldán Ortega, R. Roshan Biswal, and E. Sánchez Delacruz, “Detección de enfermedades en el sector agrícola utilizando Inteligencia Artificial Detection of Diseases in the Agricultural Sector using Artificial Intelligence,” Misantla, Veracruz, México , May 2019.
- [14] A. C. Romero and H. H. Cortes, “Machine learning in plant disease detection Artículo de investigación,” 2019, [Online]. Available: <https://revistas.udistrital.edu.co/index.php/tia/issue/archive>
- [15] F. Reuter, “PRINCIPIOS DE TELEDETECCIÓN Serie Didáctica N° 33,” Argentina, Mar. 2009.
- [16] CNICE, “Plataformas de Teledetección,” 2019. <http://concurso.cnice.mec.es/cnice2006/material121/unidad3/satelite1.htm#:~:text=Plataformas%20de%20Teledetecci%C3%B3n&text=Se%20entie>

nde%20por%20plataforma%20de,y%20transmitir%20im%C3%A1genes
%20a%20distancia. (accessed May 15, 2023).

- [17] “Tema 3: Plataformas, sensores y canales,” España, 2018. Accessed: May 15, 2023. [Online]. Available: <https://www.um.es/geograf/sigmur/teledet/tema03.pdf>
- [18] Agricultura de precision para el desarrollo, “Uso de los Drones Agrícolas: Sobrevolar el Lote para Tomar Decisiones,” 2020. <http://agriculturadeprecisionparaeldesarrollo.com/drones-sobrevolar-el-lote-para-tomar-decisiones/> (accessed May 15, 2023).
- [19] M. Soria and M. Matar, “NOCIONES SOBRE TELEDETECCION,” Argentina, 2019.
- [20] INDRA, “SISTEMAS DE TELEDETECCIÓN Comunicaciones por satélite, observación de la tierra, navegación y posicionamiento y estaciones de control,” Madrid (España), 2018. Accessed: May 16, 2023. [Online]. Available: https://www.indracompany.com/sites/default/files/indra_teledeteccion_es_baja_0.pdf
- [21] R. Edrosa, “Aplicación de la Teledetección para el Monitoreo de Eventos Hídricos Superficiales mediante Imágenes Cosmo Sky Med,” Córdoba - Argentina, Jul. 2018. Accessed: May 17, 2023. [Online]. Available: https://rdu.unc.edu.ar/bitstream/handle/11086/11539/2009_Edrosa-Rodrigo.pdf?sequence=1&isAllowed=y
- [22] N. Bravo, “Teoría - Teledetección Espacial LANDSAT, SENTINEL2, ASTER L1T y MODIS,” Huánuco - Perú, 2017. Accessed: May 17, 2023. [Online]. Available: https://acolita.com/wp-content/uploads/2018/01/Teledeteccion_espacial_ArcGeek.pdf
- [23] J. Fernandez and R. Corbella, “TELEDETECCIÓN: Imágenes Satelitales y Fotografía aérea,” Argentina.
- [24] G. Segarra, “Aplicación de teledetección en los recursos naturales y medio ambiente,” *ESPOCH*, 2019.

<https://www.calameo.com/read/0011312663d99e70ae097> (accessed May 16, 2023).

- [25] R. Bautista, “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE VISIÓN ARTIFICIAL PARA ANÁLISIS DE DATOS NDVI EN IMÁGENES ESPECTRALES DE CULTIVOS DE BRÓCOLI OBTENIDOS MEDIANTE UNA AERONAVE PILOTADA REMOTAMENTE,” Riobamba, Jan. 2018.
- [26] CropLife Latin America, “Uso de drones en la agricultura,” Jan. 2021. <https://www.croplifela.org/es/actualidad/articulos/uso-de-drones-en-la-agricultura> (accessed May 21, 2023).
- [27] P. Ruipérez, “DISEÑO Y FABRICACIÓN DE UN DRON.” Accessed: May 22, 2023. [Online]. Available: <https://riunet.upv.es/bitstream/handle/10251/73170/RUIP%C3%89REZ%20-%20Dise%C3%B1o%20y%20fabricaci%C3%B3n%20de%20un%20dron%20mediante%20impresi%C3%B3n%203D.pdf?sequence=5>
- [28] J. M. Sullivan, “Evolution or revolution? The rise of UAVs,” *IEEE Technology and Society Magazine*, vol. 25, no. 3, pp. 43–49, Sep. 2006, doi: 10.1109/MTAS.2006.1700021.
- [29] A. Gastón and G. Pérez, “Introducción a los UAV’s, Drones o VANTs de uso civil,” Buenos Aires - Argentina, Oct. 2014. [Online]. Available: www.cema.edu.ar/publicaciones/doc_trabajo.html
- [30] A. Chiliquinga, “Montaje del sistema de control de un dron, en base a las especificaciones técnicas del cuadricóptero T4, para la Carrera de Mecánica Aeronáutica de la Unidad de Gestión de Tecnologías–ESPE,” Jul. 2020, Accessed: May 22, 2023. [Online]. Available: <http://repositorio.espe.edu.ec/bitstream/21000/28810/2/M-ESPEL-CMA-0760.pdf>
- [31] J. Yoe, R. Morales, E. S. Bustos, E. C. Calderón, and M. Calixto-Rodriguez, “SEXTO SEMINARIO DE INGENIERIAS ENTRE

- CUERPOS ACADEMICOS DC-DC Topologies for Full Cell Applications View project Small Wind Turbine Optimization and Market Promotion View project,” 2023. [Online]. Available: <https://www.researchgate.net/publication/369472205>
- [32] A. Laura, “SISTEMA AUTOMÁTICO PARA EL CONTROL DE LA CALIDAD DEL CALZADO MEDIANTE VISIÓN ARTIFICIAL,” Ambato, Mar. 2023. Accessed: May 23, 2023. [Online]. Available: <https://repositorio.uta.edu.ec/jspui/bitstream/123456789/38449/1/t2254te.pdf>
- [33] J. Piscocya, “SISTEMA DE VISIÓN ARTIFICIAL PARA APOYAR EN LA IDENTIFICACIÓN DE PLAGAS Y ENFERMEDADES DEL CULTIVO DE SANDÍA EN EL DISTRITO DE FERREÑAFE,” Chiclayo, 2019. Accessed: May 23, 2023. [Online]. Available: https://tesis.usat.edu.pe/bitstream/20.500.12423/2356/1/TL_PiscocyaFerre%C3%B1anJesus.pdf
- [34] R. M. Pérez, J. S. Arias, and A. M. Porras, “Introducción al Aprendizaje Automático con YOLO Introduction to Machine Learning with YOLO.” [Online]. Available: <https://developer.nvidia.com/cuda->
- [35] B. Montenegro and M. Flores, “Detección de peatones en el día y en la noche usando YOLO-v5,” *Ingenius*, vol. 2022, no. 27, pp. 23–31, Jan. 2022, doi: 10.17163/ings.n27.2022.03.
- [36] P. Everardo Zamora, “EL CULTIVO DEL BROCOLI,” México, Jan. 2016. Accessed: May 23, 2023. [Online]. Available: <https://dagus.unison.mx/Zamora/BROCOLI-DAG-HORT-010.pdf>
- [37] C. Sánchez, “Comportamiento de cinco genotipos de brócoli (*Brassica oleracea* var. *itálica*) bajo condiciones de sombreadero y manejo orgánico,” México, Dec. 2009. Accessed: May 23, 2023. [Online]. Available: <http://repositorio.uaaan.mx:8080/xmlui/bitstream/handle/123456789/2412/CAROLINA%20S%C3%81NCHEZ%20P%C3%89REZ.pdf?sequence=1>

&isAllowed=y

- [38] infoAgro, “EL CULTIVO DEL BRÓCULI,” *CORTEVA agriscience*. <https://www.infoagro.com/hortalizas/brocoli.htm> (accessed May 23, 2023).
- [39] A. Trujillo, “Los drones se apoderan del cuidado agrícola en Ecuador,” *Diario Expreso*, Mar. 22, 2023. <https://www.expreso.ec/ciencia-y-tecnologia/drones-apoderan-cuidado-agricola-ecuador-154721.html> (accessed Jun. 09, 2023).
- [40] “Phantom 4 Multispectral,” *djiDRON*. <https://www.djidron.com/dron-dji/phantom-4/phantom-4-multispectral/> (accessed Jun. 09, 2023).
- [41] “Mavic 3 Multispectral,” *djiEcuador*, 2022. <https://djiecuador.ec/producto/dji-mavic-3-multispectral/> (accessed Jun. 09, 2023).
- [42] “DJI Mavic 3M,” 2023. <https://ag.dji.com/mavic-3-m> (accessed Jun. 09, 2023).
- [43] “Matrice 300 RTK,” *djiENTERPRISE*, 2023. <https://enterprise.dji.com/matrice-300/specs> (accessed Jun. 09, 2023).
- [44] “DJI MATRICE 300 RTK,” *djiECUADOR*, 2022. <https://djiecuador.ec/producto/dji-matrice-300-rtk/> (accessed Jun. 09, 2023).
- [45] RCNITRO, “KingKong 260 PNP (Verde),” 2023. <https://www.rcnitro.com.ar/producto/kingkong-260-pnp-verde/> (accessed Jun. 11, 2023).
- [46] BETAFPV, “Beta85X 4K Whoop Cuadróptero (4S),” 2023. <https://betafpv.com/products/beta85x-4k-whoop-quadcopter-4s> (accessed Jun. 11, 2023).
- [47] olbax, “WALKERA FURIOUS 320.” <http://www.olbax.com/walkerafurious320/walkera-furious-320> (accessed Jun. 11, 2023).


- [48] Fundación Carlos Slim, “PROGRAMADOR DE MICROORDENADORES COMPONENTES PRINCIPALES DE LOS MICROORDENADORES,” 2019. Accessed: Jun. 13, 2023. [Online]. Available: <https://capacitateparaeempleo.org/assets/5i1yov2.pdf>
- [49] BricoGeek, “Banana Pi - ARM Cortex-A7 1GHz Dual-Core.” <https://tienda.bricogeek.com/descatalogado/685-banana-pi-arm-cortex-a7-1ghz-dual-core.html> (accessed Jun. 13, 2023).
- [50] xataka, “Raspberry Pi 4 .” <https://www.xataka.com/ordenadores/raspberry-pi-4-caracteristicas-precio-ficha-tecnica> (accessed Jun. 13, 2023).
- [51] SIMCom, “SIM808_Hardware Design_V1.00,” China, 2014. Accessed: Jun. 14, 2023. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/SIM808_Hardware+Design_V1.00.pdf
- [52] Amazon, “Huawei E586 DESBLOQUEADO Inalámbrico HSPA + 21 M Pocket WIFI 3G Módem móvil 21Mbps,” 2023. <https://www.amazon.de/-/en/Huawei-E586-UNLOCKED-Wireless-Pocket-Mobile/dp/B0061JPI4C> (accessed Jun. 14, 2023).
- [53] Filmora, “Análisis completo de la GoPro Hero5 Black,” 2023. <https://filmora.wondershare.es/action-camera/gopro-hero-5-black-review.html> (accessed Jun. 16, 2023).
- [54] DJI, “DJI Pocket 2: Unboxing, características, aspectos destacados,” *godron*, 2022. <https://tienda.godron.mx/dji-pocket-2-unboxing-caracteristicas-aspectos-destacados/> (accessed Jun. 16, 2023).
- [55] D. Rosillo, “APEMAN A80 4K,” *MiCámaraDeportiva.com*, 2020. <https://micamaradeportiva.com/apeman-a80-review/> (accessed Jun. 16, 2023).
- [56] C. Monroy, “EL LENGUAJE PYTHON Y SU POTENCIAL EN EL DESARROLLO DE SOFTWARE DE INTELIGENCIA ARTIFICIAL,” *Instituto de investigaciones científicas y tecnológicas - USAM*, Accessed: Jun. 16, 2023. [Online]. Available:

<https://docs.bvsalud.org/biblioref/2022/05/1368667/masferrer-investigano-12-vol-1-marzo-2022-pags-18-41.pdf>

- [57] M. Ruipérez, “DISEÑO Y FABRICACIÓN DE UN DRON.”
- [58] J. Acosta, B. Martinez, A. Cerda, B. Ferrandez, and E. Nuñez, “ALIMENTOS DE LA REGIÓN DE MURCIA: BRÓCOLI,” Murcia, 2018. Accessed: Jul. 12, 2023. [Online]. Available: https://www.ucam.edu/sites/default/files/catedras/agro-santander/informe_brocoli_web.pdf
- [59] M. M. Hossain and M. J. Mohona, “Shoot-root traits of broccoli (*Brassica oleracea* var. *italica* L.) as influenced by different irrigation schedules,” *Int J Horti Sci Technol*, vol. 5, no. 1, pp. 11–18, 2018, doi: 10.22059/ijhst.2018.256331.235.
- [60] R. Caiza, “Determinación del periodo crítico del cultivo de brócoli (*Brassica oleracea* L.) Var. Avenger en competencia interespecífica,” Riobamba, 2016. Accessed: Jul. 12, 2023. [Online]. Available: <http://dspace.esPOCH.edu.ec/bitstream/123456789/5744/1/13T0839.pdf>



ANEXOS

Anexo A. Ficha técnica de la tarjeta SIM 808



GSM/GPRS+GPS Module

SIM808



SIM808 module is a complete Quad-Band GSM/GPRS module which combines GPS technology for satellite navigation. The compact design which integrated GPRS and GPS in a SMT package will significantly save both time and costs for customers to develop GPS enabled applications. Featuring an industry-standard interface and GPS function, it allows variable assets to be tracked seamlessly at any location and anytime with signal coverage.

General features

- Quad-band 850/900/1800/1900MHz
- GPRS multi-slot class 12/10
- GPRS mobile station class B
- Compliant to GSM phase 2/2+
 - Class 4 (2 W @ 850/900MHz)
 - Class 1 (1 W @ 1800/1900MHz)
- Dimensions: 24*24*2.6mm
- Weight: 3.3g
- Control via AT commands (3GPP TS 27.007, 27.005 and SIMCOM enhanced AT Commands)
- Supply voltage range 3.4 – 4.4V
- Low power consumption
- Operation temperature: -40℃ – 85℃

Specifications for GPRS Data

- GPRS class 12: max. 85.6 kbps (downlink/uplink)
- PBCCH support
- Coding schemes CS 1, 2, 3, 4
- PPP-stack
- USSD

Specifications for SMS via GSM/GPRS

- Point to point MO and MT
- SMS cell broadcast
- Text and PDU mode

Software features

- 0710 MUX protocol
- Embedded TCP/UDP protocol
- FTP/HTTP
- MMS
- POP3/SMTP
- DTMF
- Jamming Detection
- Audio Record
- SSL
- Bluetooth 3.0 (optional)
- TTS CN(optional)
- Embedded AT (optional)

Compatibility

- AT cellular command interface

Specification for GPS

- Receiver type
- 22 tracking /66 acquisition -channel
- GPS L1 C/A code
- Sensitivity
- Tracking: -165 dBm
- Cold starts : -148 dBm
- Time-To-First-Fix
- Cold starts: 32s (typ.)
- Hot starts: <1s
- Warm starts: 3s
- Accuracy
- Horizontal position : <2.5m CEP

Interfaces

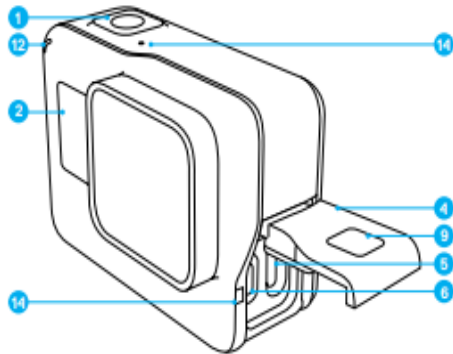
- 68 SMT pads including
- Analog audio interface
- PCM interface(optional)
- SPI interface (optional)
- RTC backup
- Serial interface
- USB interface
- Interface to external SIM 3V/1.8V
- Keypad interface
- GPIO
- ADC
- GSM Antenna pad
- Bluetooth Antenna pad
- GPS Antenna pad

Certifications

- CE
- FCC

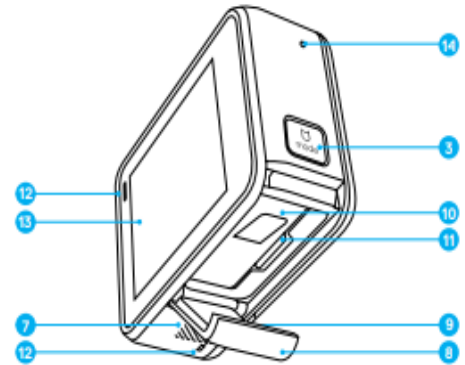
Anexo B. Manual de usuario de cámara GoPro HERO 5

SU HERO5 BLACK



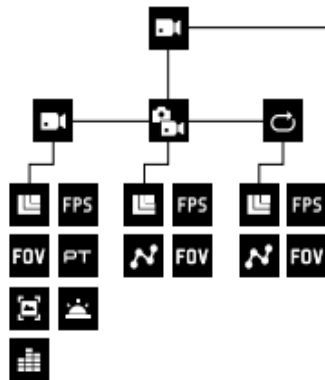
- 1. Botón Obturador []
- 2. Pantalla de estado de la cámara
- 3. Botón Modo []
- 4. Puerta lateral
- 5. Puerto USB-C
- 6. Puerto micro HDMI (cable no incluido)
- 7. Altavoz
- 8. Puerta de la batería
- 9. Botón de liberación del cierre
- 10. Batería

SU HERO5 BLACK



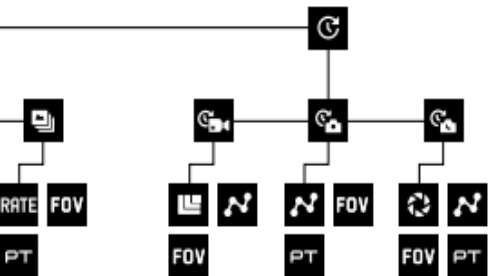
- 11. Ranura para tarjeta microSD
- 12. Luz de estado de la cámara
- 13. Pantalla táctil
- 14. Micrófono

MAPA DE MODOS Y CONFIGURACIÓN



- | | |
|----------------|-------------------------|
| Baja luz auto | Control de audio manual |
| Ráfaga | Foto nocturna |
| FOV FOV | Secuencia nocturna |
| FPS FPS | Modo Foto/Foto |
| Intervalo | PT Protune |
| En bucle | RATE Velocidad |


MAPA DE MODOS Y CONFIGURACIÓN



- | | |
|------------------|-------------------------|
| RAW RAW | Modo Video/Video |
| Resolución (RES) | Video y foto |
| Obturador | Estabilización de video |
| Modo Secuencia | WDR |
| Foto secuencial | |
| Video secuencial | |

Anexo C. Tablas de producción, ventas y pérdidas según INEC 2022

TABLA 6.
SUPERFICIE, SEGÚN PRODUCCIÓN Y VENTAS
POR CULTIVOS TRANSITORIOS
(Hectáreas, Toneladas Métricas)




ÍNDICE

CULTIVOS TRANSITORIOS	SUPERFICIE (Has.)		PRODUCCIÓN (Tm.)	VENTAS (Tm.)
	Sembrada	Cosechada		
TOTAL NACIONAL	Solo	948.999	923.693	
	Asociado	57.434	52.914	
ARROZ (EN CÁSCARA)	Solo	342.421	339.735	1.503.527
	Asociado	546	546	687
ARVEJA SECA (GRANO SECO)	Solo	1.255	1.126	415
	Asociado	93	76	8
ARVEJA TIERNA (EN VAINA)	Solo	5.990	5.425	16.837
	Asociado	554	338	420
BROCOLI (REPOLLO)	Solo	7.133	7.039	129.533
	Asociado	42	24	231
CEBADA (GRANO SECO)	Solo	10.527	9.707	14.681
	Asociado			
CEBOLLA BLANCA (TALLO FRESCO)	Solo	4.845	4.775	12.191
	Asociado			
FRÉJOL SECO (GRANO SECO)	Solo	17.432	16.863	12.944
	Asociado	13.918	12.676	4.773
FRÉJOL TIERNO (VAINA)	Solo	4.509	4.387	11.859
	Asociado	5.005	4.736	5.818
HABA SECA (GRANO SECO)	Solo	2.200	2.091	2.957
	Asociado	1.366	1.252	353
HABA TIERNA (VAINA)	Solo	6.109	5.606	24.081
	Asociado			

ÍNDICE | T1 | T2 | T3 | T4 | T5 | **T6** | T7 | T8 | T9 | T10 | T11 | T12 | T13 | T ... +

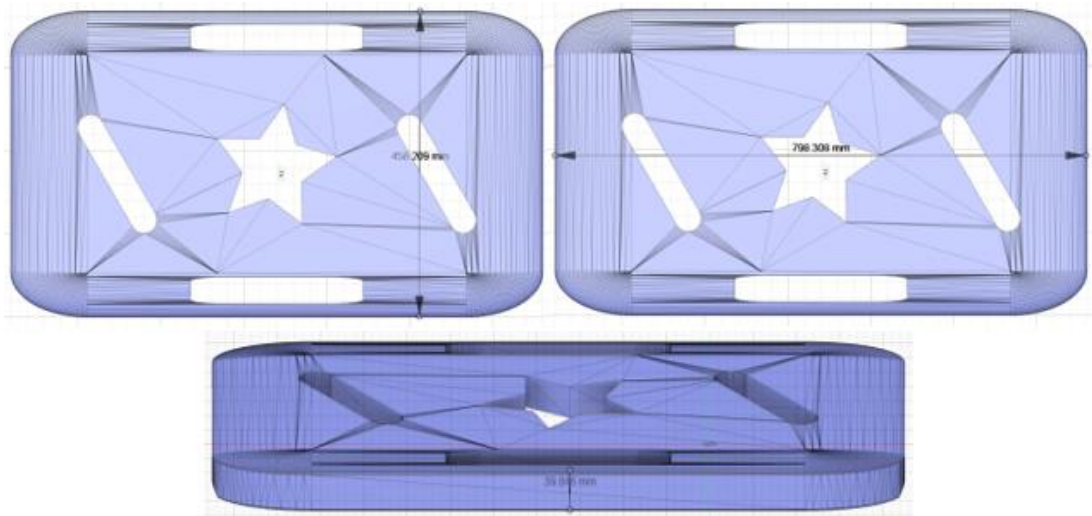
TABLA 7.
SUPERFICIE PERDIDA, SEGÚN DIFERENTES CAUSAS
POR CULTIVOS TRANSITORIOS
(Hectáreas)



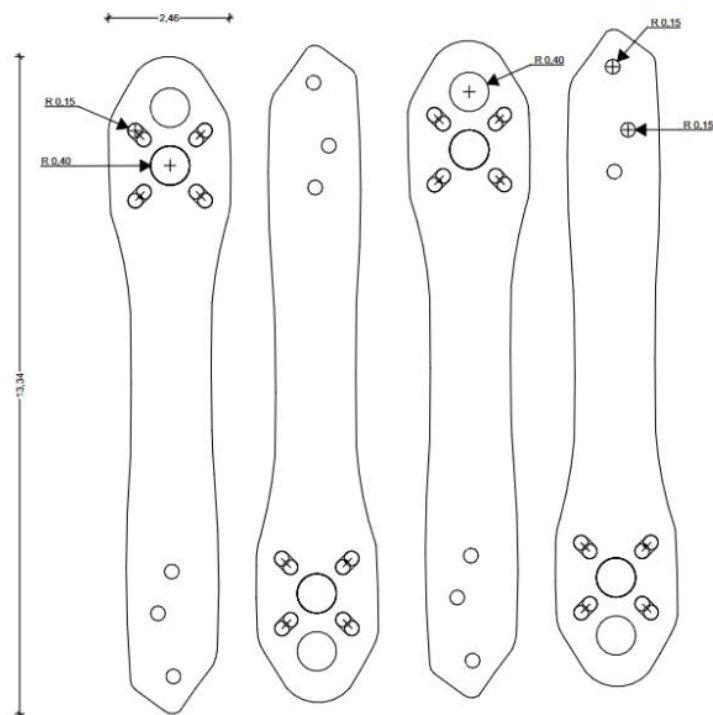
CULTIVOS TRANSITORIOS	SUPERFICIE PERDIDA (Has.)							
	Total	Sequía	Helada	Plagas	Enfermedades	Inundación	Otra razón	
TOTAL NACIONAL	Solo	23.412	2.309	5.083	9.431	1.776	2.755	2.059
	Asociado	4.520	1.073	1.419	1.098	133	400	398
ARROZ (EN CÁSCARA)	Solo	2.686	105	549	1.058		837	137
	Asociado							
ARVEJA SECA (GRANO SECO)	Solo	128		1	7	56	4	60
	Asociado	17	12	2			3	
ARVEJA TIERNA (EN VAINA)	Solo	565	27	41	70	28	282	116
	Asociado	217	134	64	19			
BROCOLI (REPOLLO)	Solo	34		16	33			45
	Asociado	18			4	14		
CEBADA (GRANO SECO)	Solo	820	131	413	91	53	61	71
	Asociado							
CEBOLLA BLANCA (TALLO FRESCO)	Solo	70		11	57	3		
	Asociado							
FRÉJOL SECO (GRANO SECO)	Solo	569	8	11	276	73	138	63
	Asociado	1.242	308	443	233	58	96	103
FRÉJOL TIERNO (EN VAINA)	Solo	122	18	6	56	4	38	
	Asociado	269	40	81	24	3	54	67
HABA SECA (GRANO SECO)	Solo	109	13	17	56	1		23
	Asociado	114	35	47	10		14	8
HABA TIERNA (EN VAINA)	Solo	503	44	121	117	152	67	1
	Asociado	112	22	7	36		12	35
MAÍZ DURO CHOCLO (EN CHOCLO)	Solo	115		3	112			
	Asociado							

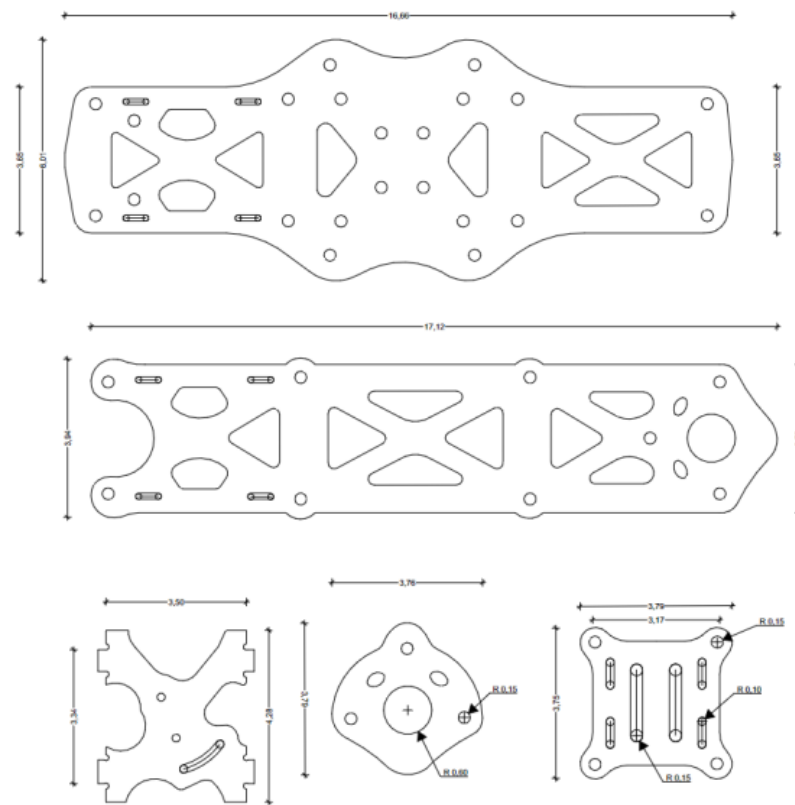
ÍNDICE | T1 | T2 | T3 | T4 | T5 | T6 | **T7** | T8 | T9 | T10 | T11 | T12 | T13 | T ... +

Anexo D. Diseño de protector de batería



Anexo E. Diseño del frame o cuerpo del dron





Anexo F. Código Python para el procesamiento de las imágenes

```

import tkinter
import cv2

import time
import threading
import torch
import numpy as np
import tkinter.font as font
from tkinter import ttk
from tkinter import *
import json as json
from PIL import Image, ImageTk
from urllib.parse import urlencode
from urllib.request import Request, urlopen
import serial
import os, time
import requests

general = 0

```

```

procesoCa = 0
enviando = 0
estado1 = ""
estado2 = ""
estado3 = ""
estado4 = ""
valCamara = 0
valError = ""
valorEstado=0
lat=0
lon=0
alt=0

try:
    conSerial = serial.Serial('COM5', 9600)

    time.sleep(1)
    model = torch.hub.load('ultralytics/yolov5', 'custom',
                           path='C:/Users/Paul
Laura/Documents/python/Deteccion/best.pt')

except:
    print("Error")

# CREAR VENTANA
ventana = tkinter.Tk()
ventana.state("zoomed")
ventana.title("Control de agentes patógenos y enfermedades")
ventana.config(bg="grey")

proceso = tkinter.Toplevel(ventana)
ventana.state("zoomed")
proceso.title("Control de plagas y enfermedades ")

proceso.config(bg="#d9d9d9")

datos = tkinter.Toplevel(proceso)
ventana.state("zoomed")
datos.title("Control de Calidad")
datos.config(bg="#d9d9d9")

# FUNCIONES

# enviar datos

def enviar(dir):

```

```

request = Request(dir)
json = urlopen(request).read().decode()
return

def traerEstado():
    request =
Request("http://localhost/teledeteccion/datos/captura.php")
    json = urlopen(request).read().decode()
    return json

def registrarBDD():

    enviar("http://localhost/teledeteccion/datos/crear.php")

    return

def actualizarBDD():
    global lat
    global lon
    global alt
    global valorEstado
    fecha="2023-06-29"

    enviar("http://localhost/teledeteccion/datos/actualizar.php?id='"+str(v
    alorEstado)

```

Anexo G. Código conexión a Base de datos PhpMyAdmin

```

<?php

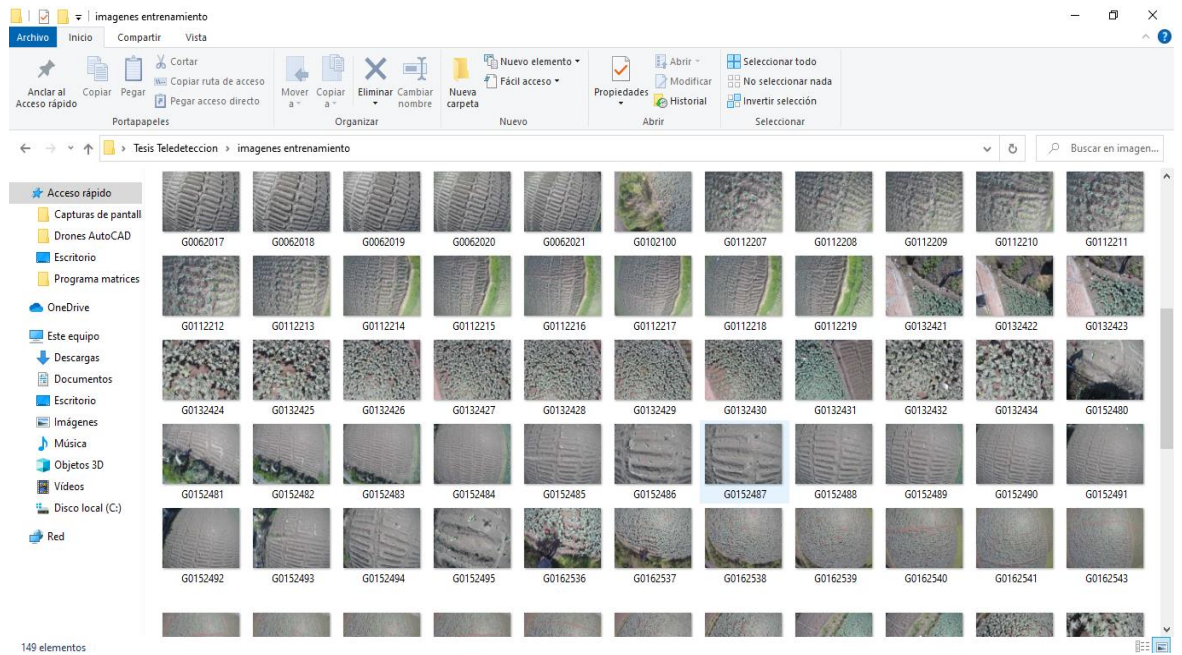
$servername = "localhost";
$dbname = "teledeteccion";
$username = "root";
$password = "";

$conn = mysqli_connect($servername,$username,$password,$dbname);
mysqli = new mysqli($servername,$username,$password,$dbname);
mysqli->set_charset("utf8");
if(!$mysqli){

    die("Error ".mysqli_connect_error());
}
?>

```

Anexo H. Imágenes capturadas para el entrenamiento de la visión artificial



Anexo I. Reglamento de las normativas a seguir dentro de la navegación de drones en el Ecuador.

DIRECCIÓN GENERAL
DE AVIACIÓN CIVIL



Resolución Nro. DGAC-DGAC-2020-0110-R

Quito, D.M., 04 de noviembre de 2020

DIRECCIÓN GENERAL DE AVIACIÓN CIVIL

CONSIDERANDO

Que, la Constitución de la República en el artículo 226 consagra el principio de legalidad o de habilitación legal previa, por el cual: *“Las instituciones del Estado, sus organismos, dependencias, las servidoras o servidores públicos y las personas que actúen en virtud de una potestad estatal ejercerán solamente las competencias y facultades que les sean atribuidas en la Constitución y la ley. [...]”*;

Que, la Constitución de la República en el artículo 394 establece que: *“El Estado garantizará la libertad de transporte [...], aéreo, [...] dentro del territorio nacional, sin privilegios de ninguna naturaleza”*;

Que, la Ley de Aviación Civil en el artículo 5 establece que la *“Dirección General de Aviación Civil es una entidad autónoma de derecho público, con personería jurídica y fondos propios, con sede en el Distrito Metropolitano de Quito”*, y que *“El Director General de Aviación Civil, es la máxima autoridad de la Entidad y será designado por el Presidente de la República, de una terna propuesta por el Consejo Nacional de Aviación Civil”*;

Que, el señor Presidente Constitucional de la República del Ecuador, mediante Decreto Nro. 728 de 29 de abril de 2019, designó al señor Anyelo Patricio Acosta Arroyo como Director General de Aviación Civil;

Que, la Dirección General de Aviación Civil con Resolución Nro. DGAC-DGAC-2020-0074-R de 30 de agosto de 2020, aprobó el nuevo Reglamento que norma la **“Operación de Aeronaves Pilotadas a Distancia (RPAs)”**;

Que, el Comité de Normas, en reunión efectuada el 28 de septiembre de 2020, creó un grupo de trabajo, para la revisión de la Resolución Nro. DGAC-DGAC-2020-0074-R de 30 de agosto de 2020, que trata sobre Reglamento para la **“Operación de Aeronaves Pilotadas a Distancia (RPAs)”**, a fin de adecuarlo a las necesidades que implica esta materia y se revise la carga regulatoria establecida y con Memorando Nro.

ANEXO J: Tabla de recopilación de datos en el sistema de teledetección

N°	Fecha	Hora	Tiempo de procesamiento	Posicionamiento del dron			Velocidad Dron (m/s)	Altura (m)	Velocidad Viento (Km/h)
				Latitud	Longitud	Altitud			
1	2023-07-09	09:10:55	00:00:02	-1.2036	-78.5946	2626.4249	2	3	8
2	2023-07-09	09:11:31	00:00:36	-1.2037	-78.5946	2626.4249	2	3	8
3	2023-07-09	09:12:42	00:01:11	-1.2036	-78.5947	2626.4250	2	3	8
4	2023-07-09	09:14:59	00:02:29	-1.2036	-78.5946	2626.4249	2	3	8
5	2023-07-09	09:15:22	00:00:23	-1.2038	-78.5946	2626.4249	2	3	8
6	2023-07-09	09:16:08	00:00:46	-1.2037	-78.5946	2626.4248	2	3	8
7	2023-07-09	09:17:32	00:01:24	-1.2036	-78.5946	2626.4249	2	3	8
8	2023-07-09	09:18:14	00:00:42	-1.2037	-78.5946	2626.4248	2	3	8
9	2023-07-09	09:20:09	00:02:36	-1.2036	-78.5946	2626.4249	2	3	8
10	2023-07-09	13:08:54	00:00:02	-1.2037	-78.5947	2626.4248	2	3	8
11	2023-07-09	13:09:25	00:00:31	-1.2036	-78.5948	2626.4249	2	3	8
12	2023-07-09	13:10:50	00:00:36	-1.2036	-78.5946	2626.4249	2	3	8
13	2023-07-09	13:11:05	00:01:11	-1.2036	-78.5948	2626.4249	2	3	8
14	2023-07-09	13:12:42	00:02:29	-1.2037	-78.5946	2626.4250	2	3	8
15	2023-07-09	13:13:29	00:00:23	-1.2036	-78.5946	2626.4248	2	3	8
16	2023-07-09	13:14:37	00:00:46	-1.2038	-78.5947	2626.4249	2	3	8
17	2023-07-09	13:15:01	00:01:24	-1.2036	-78.5946	2626.4248	2	3	8
18	2023-07-09	13:20:03	00:00:42	-1.2037	-78.5947	2626.4249	2	3	8
19	2023-07-09	13:21:22	00:00:35	-1.2036	-78.5946	2626.4249	2	3	8
20	2023-07-09	13:23:45	00:02:05	-1.2036	-78.5946	2626.4247	2	3	8
21	2023-07-10	10:11:20	00:00:02	-1.2027	-78.5917	2626.4249	4	3	10
22	2023-07-10	10:12:02	00:00:23	-1.2027	-78.5918	2626.4249	4	3	10
23	2023-07-10	10:13:55	00:00:46	-1.2027	-78.5918	2626.4250	4	3	10

24	2023-07-10	10:15:14	00:01:24	-1.2026	-78.5918	2626.4249	4	3	10
25	2023-07-10	10:16:36	00:00:42	-1.2027	-78.5917	2626.4249	4	3	10
26	2023-07-10	10:17:40	00:02:36	-1.2026	-78.5917	2626.4248	4	3	10
27	2023-07-10	10:20:31	00:00:02	-1.2027	-78.5918	2626.4249	4	3	9
28	2023-07-10	10:21:01	00:02:29	-1.2026	-78.5917	2626.4248	4	3	10
29	2023-07-10	15:05:41	00:00:23	-1.2026	-78.5917	2626.4249	4	3	10
30	2023-07-10	15:06:26	00:00:46	-1.2026	-78.5917	2626.4248	4	3	10
31	2023-07-10	15:08:52	00:01:24	-1.2027	-78.5917	2626.4249	4	3	10
32	2023-07-10	15:10:14	00:00:42	-1.2025	-78.5917	2626.4249	4	3	10
33	2023-07-10	15:11:20	00:02:05	-1.2025	-78.5917	2626.4249	4	3	10
34	2023-07-10	15:13:46	00:00:02	-1.2025	-78.5919	2626.4250	4	3	9
35	2023-07-10	15:14:29	00:00:23	-1.2024	-78.5919	2626.4248	4	3	10
36	2023-07-10	15:16:09	00:00:46	-1.2024	-78.5918	2626.4249	4	3	10
37	2023-07-10	15:20:21	00:01:24	-1.2024	-78.5917	2626.4248	4	3	10
38	2023-07-10	15:22:58	00:00:42	-1.2023	-78.5917	2626.4249	4	3	10
39	2023-07-11	08:05:37	00:00:02	-1.2023	-78.5917	2626.4249	4	3	9
40	2023-07-11	08:06:12	00:00:36	-1.2024	-78.5918	2626.4247	2	3	9
41	2023-07-11	08:08:14	00:01:11	-1.2023	-78.5917	2626.4249	2	3	9
42	2023-07-11	08:09:35	00:02:29	-1.2024	-78.5917	2626.4249	2	3	9
43	2023-07-11	08:10:24	00:00:23	-1.2023	-78.5917	2626.4250	2	3	9
44	2023-07-11	08:11:21	00:00:46	-1.2023	-78.5917	2626.4249	2	3	9
45	2023-07-11	08:12:53	00:01:24	-1.2021	-78.5917	2626.4249	2	3	9
46	2023-07-11	08:14:37	00:00:42	-1.2021	-78.5917	2626.4248	2	3	11
47	2023-07-11	08:15:06	00:02:36	-1.2021	-78.5918	2626.4249	2	3	9
48	2023-07-11	08:20:49	00:00:02	-1.2020	-78.5919	2626.4248	2	3	9
49	2023-07-12	12:15:31	00:00:31	-1.2035	-78.5913	2626.4249	2	3	8
50	2023-07-12	12:16:22	00:02:29	-1.2035	-78.5913	2626.4248	2	3	11
51	2023-07-12	12:17:02	00:00:23	-1.2035	-78.5914	2626.4249	2	3	11

52	2023-07-12	12:20:36	00:00:46	-1.2035	-78.5914	2626.4249	2	3	11
53	2023-07-12	12:22:17	00:01:24	-1.2034	-78.5915	2626.4249	2	3	11
54	2023-07-12	12:23:08	00:00:42	-1.2035	-78.5914	2626.4250	2	3	10
55	2023-07-12	12:24:55	00:02:05	-1.2034	-78.5913	2626.4248	2	3	11
56	2023-07-12	12:25:30	00:00:02	-1.2035	-78.5913	2626.4249	2	3	11
57	2023-07-12	12:26:14	00:00:23	-1.2034	-78.5913	2626.4248	2	3	11
58	2023-07-12	12:27:50	00:00:46	-1.2034	-78.5913	2626.4249	2	3	11
59	2023-07-12	12:28:31	00:01:24	-1.2034	-78.5913	2626.4249	2	3	10
60	2023-07-12	12:29:21	00:02:29	-1.2035	-78.5914	2626.4247	2	3	11
61	2023-07-12	12:30:01	00:00:23	-1.2034	-78.5914	2626.4249	2	3	11
62	2023-07-12	12:34:22	00:00:46	-1.2033	-78.5915	2626.4249	2	3	10
63	2023-07-12	12:35:08	00:01:24	-1.2033	-78.5915	2626.4250	2	3	11
64	2023-07-12	12:36:45	00:00:42	-1.2034	-78.5915	2626.4249	2	3	11
65	2023-07-12	12:37:09	00:00:35	-1.2034	-78.5915	2626.4249	2	3	11
66	2023-07-13	09:10:55	00:02:05	-1.2035	-78.5913	2626.4248	2	3	9
67	2023-07-13	09:11:31	00:00:02	-1.2035	-78.5913	2626.4249	2	3	9
68	2023-07-13	09:12:42	00:00:23	-1.2035	-78.5914	2626.4248	2	3	9
69	2023-07-13	09:14:59	00:00:46	-1.2035	-78.5914	2626.4249	2	3	10
70	2023-07-13	09:15:22	00:01:24	-1.2034	-78.5915	2626.4248	2	3	10
71	2023-07-13	09:16:08	00:00:42	-1.2035	-78.5914	2626.4249	2	3	10
72	2023-07-13	09:17:32	00:00:36	-1.2034	-78.5913	2626.4249	2	3	9
73	2023-07-13	09:18:14	00:01:11	-1.2035	-78.5913	2626.4249	2	3	9
74	2023-07-13	09:20:09	00:02:29	-1.2034	-78.5913	2626.4250	2	3	9
75	2023-07-13	13:08:54	00:00:23	-1.2034	-78.5913	2626.4248	2	3	9
76	2023-07-13	13:09:25	00:00:46	-1.2034	-78.5913	2626.4249	2	3	9
77	2023-07-13	13:10:50	00:01:24	-1.2035	-78.5914	2626.4248	2	3	11
78	2023-07-13	13:11:05	00:00:42	-1.2034	-78.5914	2626.4249	2	3	10
79	2023-07-13	13:12:42	00:02:36	-1.2033	-78.5915	2626.4249	2	3	10

80	2023-07-13	13:13:29	00:00:02	-1.2033	-78.5915	2626.4247	2	3	9
81	2023-07-13	13:14:37	00:00:31	-1.2034	-78.5915	2626.4249	2	3	9
82	2023-07-15	08:05:37	00:02:29	-1.2034	-78.5915	2626.4249	2	3	9
83	2023-07-15	08:06:12	00:00:23	-1.2034	-78.5916	2626.4250	2	3	10
84	2023-07-15	08:08:14	00:00:46	-1.2033	-78.5916	2626.4249	2	3	9
85	2023-07-15	08:09:35	00:01:24	-1.2033	-78.5916	2626.4249	2	3	9
86	2023-07-15	08:10:24	00:00:42	-1.2033	-78.5916	2626.4248	2	3	9
87	2023-07-15	08:11:21	00:02:05	-1.2032	-78.5916	2626.4249	2	3	9
88	2023-07-15	08:12:53	00:00:02	-1.2033	-78.5915	2626.4248	2	3	9
89	2023-07-15	08:14:37	00:00:23	-1.2034	-78.5916	2626.4249	2	3	9
90	2023-07-15	08:15:06	00:00:46	-1.2033	-78.5916	2626.4248	2	3	9
91	2023-07-15	08:20:49	00:01:24	-1.2032	-78.5915	2626.4249	2	3	11
92	2023-07-15	08:21:02	00:00:42	-1.2033	-78.5916	2626.4249	2	3	11
93	2023-07-15	08:22:55	00:00:02	-1.2033	-78.5916	2626.4249	2	3	11
94	2023-07-15	08:23:31	00:00:42	-1.2032	-78.5916	2626.4250	2	3	9
95	2023-07-15	08:25:43	00:02:05	-1.2032	-78.5914	2626.4248	2	3	9
96	2023-07-18	10:11:20	00:00:02	-1.2031	-78.5913	2626.4249	2	3	10
97	2023-07-18	10:12:02	00:00:23	-1.2031	-78.5913	2626.4248	2	3	10
98	2023-07-18	10:13:55	00:00:46	-1.2031	-78.5913	2626.4249	2	3	10
99	2023-07-18	10:15:14	00:01:24	-1.2031	-78.5914	2626.4249	2	3	10
100	2023-07-18	10:16:36	00:02:29	-1.2031	-78.5914	2626.4247	2	3	10
101	2023-07-18	10:17:40	00:00:23	-1.2030	-78.5915	2626.4249	2	3	9
102	2023-07-18	10:20:31	00:00:46	-1.2030	-78.5915	2626.4249	2	3	9
103	2023-07-18	10:21:01	00:01:24	-1.2030	-78.5915	2626.4250	2	3	9
104	2023-07-18	15:05:41	00:00:42	-1.2029	-78.5914	2626.4249	2	3	10
105	2023-07-18	15:06:26	00:00:35	-1.2030	-78.5914	2626.4249	2	3	10
106	2023-07-18	15:08:52	00:02:05	-1.2030	-78.5913	2626.4248	2	3	10
107	2023-07-18	15:10:14	00:00:02	-1.2030	-78.5915	2626.4249	2	3	10

108	2023-07-18	15:11:20	00:00:36	-1.2029	-78.5939	2626.4248	2	3	9
109	2023-07-18	15:13:46	00:01:11	-1.2028	-78.5939	2626.4249	2	3	9
110	2023-07-20	09:10:55	00:02:29	-1.2027	-78.5939	2626.4248	2	3	11
111	2023-07-20	09:11:31	00:00:23	-1.2026	-78.5940	2626.4249	2	3	11
112	2023-07-20	09:12:42	00:00:46	-1.2026	-78.5939	2626.4249	2	3	11
113	2023-07-20	09:14:59	00:01:24	-1.2026	-78.5939	2626.4249	2	3	11
114	2023-07-20	09:15:22	00:00:42	-1.2028	-78.5938	2626.4250	2	3	10
1115	2023-07-20	09:16:08	00:02:36	-1.2029	-78.5938	2626.4248	2	3	11
116	2023-07-20	09:17:32	00:00:02	-1.2029	-78.5938	2626.4249	2	3	10
117	2023-07-20	09:18:14	00:00:31	-1.2026	-78.5937	2626.4248	2	3	11
118	2023-07-20	09:20:09	00:02:29	-1.2028	-78.5937	2626.4249	2	3	10
119	2023-07-20	13:08:54	00:00:46	-1.2029	-78.5938	2626.4249	2	3	11
120	2023-07-20	13:09:25	00:01:24	-1.2028	-78.5937	2626.4247	2	3	10
121	2023-07-20	13:10:50	00:00:42	-1.2028	-78.5937	2626.4249	2	3	10
122	2023-07-20	13:11:05	00:00:36	-1.2027	-78.5937	2626.4249	2	3	10
123	2023-07-22	09:10:55	00:01:11	-1.2027	-78.5938	2626.4250	2	3	9
124	2023-07-22	09:11:31	00:02:29	-1.2027	-78.5938	2626.4249	2	3	9
125	2023-07-22	09:12:42	00:00:23	-1.2026	-78.5937	2626.4249	2	3	10
126	2023-07-22	09:14:59	00:00:46	-1.2026	-78.5938	2626.4248	2	3	9
127	2023-07-22	09:15:22	00:01:24	-1.2026	-78.5938	2626.4249	2	3	9
128	2023-07-22	09:16:08	00:00:42	-1.2025	-78.5937	2626.4248	2	3	9
129	2023-07-22	09:17:32	00:02:36	-1.2025	-78.5936	2626.4249	2	3	9
130	2023-07-22	09:18:14	00:00:02	-1.2027	-78.5935	2626.4248	2	3	11
131	2023-07-22	09:20:09	00:00:31	-1.2028	-78.5935	2626.4249	2	3	11
132	2023-07-22	13:08:54	00:00:23	-1.2029	-78.5934	2626.4249	2	3	11
133	2023-07-22	13:09:25	00:00:46	-1.2027	-78.5934	2626.4249	2	3	11
134	2023-07-23	09:10:55	00:01:24	-1.2026	-78.5935	2626.4250	3	3	15
135	2023-07-23	09:11:31	00:00:42	-1.2028	-78.5935	2626.4248	3	3	15

136	2023-07-23	09:12:42	00:02:05	-1.2026	-78.5935	2626.4249	3	3	15
137	2023-07-23	09:14:59	00:00:02	-1.2025	-78.5934	2626.4248	3	3	15
138	2023-07-23	09:15:22	00:00:23	-1.2025	-78.5935	2626.4249	3	3	14
139	2023-07-23	09:16:08	00:00:46	-1.2024	-78.5933	2626.4249	3	3	14
140	2023-07-23	09:17:32	00:01:24	-1.2026	-78.5932	2626.4247	3	3	15
141	2023-07-23	09:18:14	00:00:42	-1.2025	-78.5932	2626.4249	3	3	15
142	2023-07-23	09:20:09	00:00:02	-1.2024	-78.5932	2626.4249	3	3	15
143	2023-07-23	09:21:11	00:00:42	-1.2023	-78.5931	2626.4250	3	3	14
144	2023-07-23	09:23:59	00:02:05	-1.2023	-78.5931	2626.4249	3	3	14
145	2023-07-26	13:10:50	00:00:46	-1.2025	-78.5930	2626.4249	3	3	10
146	2023-07-26	13:11:05	00:01:24	-1.2023	-78.5930	2626.4248	3	3	10
147	2023-07-26	13:12:42	00:00:42	-1.2026	-78.5930	2626.4249	3	3	11
148	2023-07-26	13:13:29	00:02:05	-1.2027	-78.5930	2626.4248	3	3	11
149	2023-07-26	13:14:37	00:00:02	-1.2029	-78.5930	2626.4249	3	3	11
150	2023-07-26	13:15:09	00:00:23	-1.2029	-78.5930	2626.4248	3	3	10

Nota: Cada uno de los datos recolectados fueron obtenidos tras pruebas de vuelo a distintas horas del día, en donde se varió la velocidad del dron para comprobar la estabilidad de imagen capturada por la cámara GoPro Hero 5 resultando adecuada una velocidad de 2m/s durante su plan de vuelo. La altura del dron está definida mediante la prueba 2 realizada para cada una de las etapas del cultivo en donde se concluyó que la altura ideal para la detección satisfactoria de agentes patógenos y enfermedades es de 3 metros. Por otro lado, el tiempo de captura de la imagen se lo hace en un intervalo de 2 a 5 segundos, pero al estar trabajando con un modem portable fue necesario tener en cuenta la señal de cobertura existente en la superficie terrena evaluada. Por ende, se observa retardos hasta de 2 minutos para observar la imagen procesada en el interfaz.

Anexo K: Datasheet SIM 808 GSM/GPRS

