



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**  
**CARRERA DE INGENIERÍA DE TELECOMUNICACIONES**

**TRABAJO DE INTEGRACIÓN CURRICULAR**

**TEMA:**

**” DESARROLLO DE UN PROTOTIPO PARA LA DETECCIÓN DE ROYA (PHRAGMIDIUM MUCRONATUM) EN CULTIVOS DE ROSALES UTILIZANDO UN DRON. ”**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA DE TELECOMUNICACIONES**

**Línea de investigación:** Desarrollo y aplicación de software con enfoque en inteligencia artificial y visión por computador.

**AUTOR:** Avila Felix Farid Josue

**DIRECTOR:** Ing. Vásquez Ayala Carlos Alberto, Msc.

Ibarra – Ecuador

**2026**

**UNIVERSIDAD TÉCNICA DEL NORTE  
BIBLIOTECA UNIVERSITARIA**

**IDENTIFICACIÓN DE LA OBRA**

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

<b>DATOS DE CONTACTO</b>			
<b>CÉDULA DE IDENTIDAD:</b>	100401742-0		
<b>APELLIDOS Y NOMBRES:</b>	AVILA FELIX FARID JOSUE		
<b>DIRECCIÓN:</b>	IMBABURA, ATUNTAQUI, GENERAL ENRIQUEZ Y 10 DE AGOSTO		
<b>EMAIL:</b>	FARID2001AVIL@GMAIL.COM		
<b>TELF. FIJO:</b>	—	<b>TELF. MÓVIL:</b>	0998001187

<b>DATOS DE LA OBRA</b>	
<b>TÍTULO:</b>	DESARROLLO DE UN PROTOTIPO PARA LA DETECCIÓN DE ROYA (PHRAGMIDIUM MUCRONATUM) EN CULTIVOS DE ROSALES UTILIZANDO UN DRON.
<b>AUTOR(ES):</b>	AVILA FELIX FARID JOSUE
<b>FECHA:</b>	02/13/2026
<b>CARRERA/PROGRAMA:</b>	PREGRADO
<b>TÍTULO POR EL QUE OPTA:</b>	INGENIERÍA DE TELECOMUNICACIONES
<b>DIRECTOR:</b>	ING. VÁSQUEZ AYALA CARLOS ALBERTO, MSC.

## CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, 13 de febrero del 2026

### **EL AUTOR:**

Avila Felix Farid Josue

## **CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR**

Ibarra, 13 de febrero del 2026

Ing. Vásquez Ayala Carlos Alberto, Msc.  
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

**CERTIFICA:**

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

---

*Ing. Vásquez Ayala Carlos Alberto, Msc.*

*C.C: 100242498-2*

## DEDICATORIA

*Este trabajo de titulación está dedicado, en primer lugar, a mi padre, Darío Ávila, quien ha sido mi mayor ejemplo desde muy pequeño y me ha guiado con su esfuerzo y constancia a formarme como persona, enseñándome a seguir adelante incluso frente a las adversidades que se presente en el camino.*

*A mi madre Tatiana Felix por inculcarme valores que me recuerdan la importancia de la humildad, el respeto y el amor.*

*A mis hermanos Mildred, Jeremy y Gael, quienes son mi fortaleza y mi mayor motivación para demostrar que siempre es posible soñar y luchar por cumplir nuestros sueños.*

*A mi tía Daniela, quien ha sido un pilar fundamental en mi vida y parte importante de este proceso y de la realización de este trabajo.*

*A mis amigos Francisco, Zhara, Israel, Bryan, Kevin y Eduardo, así como a todo mi grupo de amigos con quienes compartí momentos dentro y fuera de las aulas, y a quienes llevo siempre en mi corazón.*

*Finalmente, y de manera muy especial, a mi abuelita Marta Solano, quien siempre creyó en mí incluso antes de iniciar este camino como estudiante universitario. Su apoyo y confianza fueron una fuente constante de motivación. Siempre la recuerdo cada día y atesoro con nostalgia la felicidad que sintió cuando comencé este proceso. Este trabajo es por y para ella.*

*“El esfuerzo silencioso construye los logros más grandes.”*

Avila Felix Farid Josue

## AGRADECIMIENTO

*Quiero agradecer a mis padres por el apoyo incondicional brindado en esta etapa de mi vida.*

*Agradezco a cada uno de los docentes de la carrera de ingeniería en Telecomunicaciones por sus conocimientos y enseñanzas que han puesto en mi para formarme como futuro profesional.*

*Quiero brindar mis mas sincero agradecimiento a mi tutor el ingeniero Carlos Vásquez y a mi asesor el ingeniero Henry Farinango que me brindaron de su tiempo y orientación para poder culminar este trabajo de titulación.*

*Finalmente, expreso mi agradecimiento a la empresa DAFRE y al ingeniero Fredy Castillo por brindarme la oportunidad de acceder a los invernaderos de la empresa y facilitar el desarrollo de este trabajo de titulación, permitiéndome realizar las pruebas necesarias.*

Avila Felix Farid Josue

## ÍNDICE DE CONTENIDOS

DEDICATORIA . . . . .	6
AGRADECIMIENTO . . . . .	7
ÍNDICE DE FIGURAS . . . . .	13
ÍNDICE DE TABLAS . . . . .	16
RESUMEN . . . . .	18
ABSTRACT . . . . .	19
CAPÍTULO I	
1.1 Planteamiento del Problema . . . . .	20
1.2 Objetivos . . . . .	21
1.2.1 Objetivo General . . . . .	21
1.2.2 Objetivos Específicos . . . . .	21
1.3 Alcance y delimitación . . . . .	22
1.4 Justificación . . . . .	24
CAPÍTULO II	
2.1 Cultivo de Rosas . . . . .	26
2.1.1 Principales características de los cultivos de rosas . . . . .	26
2.1.1.1 Requerimientos ambientales para el cultivo de rosas . . . . .	27
2.1.1.2 Manejo del cultivo . . . . .	27
2.1.2 Parámetros aplicados en agricultura de precisión . . . . .	28
2.1.3 Análisis general de la producción de rosas a nivel mundial y en Ecuador . . . . .	29
2.1.3.1 Producción mundial de rosas . . . . .	29
2.1.3.2 Producción de rosas en Ecuador . . . . .	30

2.1.3.3	Evolución de las exportaciones en el Ecuador . . . . .	32
2.2	Roya del Rosal ( <i>phragmidium mucronatum</i> ) . . . . .	33
2.2.1	Síntomas . . . . .	33
2.2.1.1	Impacto de la Roya en los Cultivos de Rosa . . . . .	34
2.2.1.2	Estrategias de Manejo para el Control de la Roya . . . . .	35
2.3	Visión Artificial . . . . .	36
2.3.1	Técnicas de visión artificial . . . . .	37
2.3.1.1	Técnica de segmentación . . . . .	37
2.3.1.2	Filtros . . . . .	39
2.3.2	Iluminación. . . . .	42
2.3.3	Etapas en el proceso de visión artificial . . . . .	47
2.3.4	Etapas del procesamiento de imágenes . . . . .	49
2.3.5	Componentes de un Sistema de Visión Artificial. . . . .	51
2.3.6	Algoritmos básicos para el reconocimiento de patrones . . . . .	52
2.3.6.1	Etapas de reconocimiento en los algoritmos de reconoci- miento de patrones . . . . .	53
2.3.6.2	Técnicas más comunes de reconocimiento de patrones . . . . .	53
2.4	Redes Neuronales . . . . .	54
2.4.0.1	Ventajas que ofrecen las red neuronal. . . . .	55
2.4.0.2	Elementos básicos de una red neuronal . . . . .	55
2.4.0.3	Tipos de neuronas artificiales. . . . .	57
2.4.1	Redes neuronales convolucionales. . . . .	58
2.4.1.1	Estructura de redes neuronales convolucionales (CNN) . . . . .	60

	10
2.4.2 Formación y aprendizaje de una red neuronal convencional . . . . .	61
2.4.3 Tipos de redes neuronales convolucionales . . . . .	62
2.5 Los drones, una herramienta clave en la agricultura de precisión . . . . .	63
2.5.1 Tipos y Clases de Vehículos No Tripulados . . . . .	64
2.5.2 Mapeo y topografía de cultivos . . . . .	66
2.6 Metodología del diseño . . . . .	67
2.6.1 Modelo de desarrollo en cascada . . . . .	67
 CAPÍTULO III	
3.1 Metodología del Modelo en Cascada . . . . .	69
3.2 Situación Actual . . . . .	71
3.2.1 Análisis de los resultados . . . . .	73
3.2.2 Propósito del prototipo . . . . .	73
3.2.3 Dimensionamiento de Stakeholdres . . . . .	74
3.3 Requerimientos . . . . .	75
3.3.1 Nomenclatura de los requerimientos . . . . .	75
3.3.2 Requerimientos de Stakeholders . . . . .	75
3.3.3 Requerimientos del sistema . . . . .	76
3.3.4 Requerimientos de la arquitectura . . . . .	78
3.4 Selección de Hardware y Software . . . . .	79
3.4.1 Hardware de desarrollo . . . . .	79
3.4.1.1 Selección de la placa de desarrollo . . . . .	79
3.4.1.2 Selección del dron . . . . .	80

	11
3.4.2 Software .....	81
3.4.3 Sistema Operativo .....	81
3.4.3.1 Entorno de programación .....	82
3.4.4 Lenguaje de programación .....	82
3.4.5 Algoritmos de visión artificial .....	83
3.4.6 Aplicación web .....	84
3.4.7 Base de datos .....	85
3.4.8 Aplicación para la planificación de vuelo .....	86
3.5 Diseño del Sistema .....	87
3.5.1 Bloque 1: recolección de datos .....	90
3.5.2 Bloque 2: procesamiento de la información .....	103
3.5.3 Bloque 3: tratamiento de los datos .....	113
3.5.4 Bloque 4: visualización de resultados .....	119
 CAPÍTULO IV	
4.1 Resultados del modelo de detección .....	126
4.2 Casos de pruebas .....	127
CONCLUSIONES .....	152
RECOMENDACIONES .....	153
BIBLIOGRAFÍA .....	154
ANEXOS .....	158
Anexo 1. Entrevista dirigida al propietario empresa florícola DAFRE .....	158
Anexo 2. Encuesta dirigida a los trabajadores de la empresa de DAFRE .....	158
Anexo 3. Resultados de la encuesta realizada a los trabajadores de DAFRE .....	159
Anexo 4. Localización del campo de cultivo de rosas de la empresa DAFRE .....	168

Anexo 5. Tabla de verificación de los requerimientos. . . . .	169
Anexo 6. Placa de desarrollo./ Dell G15.- Ryzen 7 . . . . .	171
Anexo 7. Dron DJI Air.3. . . . .	172
Anexo 8. Sistema operativo /Ubuntu . . . . .	173
Anexo 9. Entorno de programación / Visual.Code. . . . .	174
Anexo 10. Lenguaje de programación / Python . . . . .	175
Anexo 11. Algoritmo de visión artificial./ Yolov5. . . . .	175
Anexo 12. Aplicacion Web./ Flask. . . . .	176
Anexo 13. Base de datos / Mysql . . . . .	177
Anexo 14. Aplicación para la planificación de vuelo / Litchi . . . . .	178
Anexo 15. Procedimiento para la planificación de vuelo . . . . .	178
Anexo 16. Procedimiento de etiquetado con Roboflow . . . . .	184
Anexo 17. Procedimiento de entrenamiento con Google Colab. . . . .	190
Anexo 18. Código . . . . .	194

## ÍNDICE DE FIGURAS

Figura 1. <i>Producción de rosas en invernadero</i> . . . . .	28
Figura 2. <i>Evolución del Área de Flores y Rosas en el Ecuador</i> . . . . .	31
Figura 3. <i>Progreso de las exportaciones de flores en el Ecuador</i> . . . . .	32
Figura 4. <i>Características de la Roya</i> . . . . .	34
Figura 5. <i>Estrategias de Manejo para el Control de la Roya</i> . . . . .	36
Figura 6. <i>Segmentación de una hoja de café</i> . . . . .	38
Figura 7. <i>Segmentación de regiones en imágenes de guacamayos.</i> . . . . .	38
Figura 8. <i>Imagen de hoja de café con filtro gaussiano</i> . . . . .	40
Figura 9. <i>Imagen de hoja de café filtrada</i> . . . . .	41
Figura 10. <i>Imagen de hoja de café filtrada (LoG)</i> . . . . .	42
Figura 11. <i>Iluminación frontal y sus aplicaciones</i> . . . . .	43
Figura 12. <i>Inspección de latas con luz lateral</i> . . . . .	44
Figura 13. <i>Inspección con campo oscuro (Dark Field)</i> . . . . .	45
Figura 14. <i>Inspección de filamentos de bombillas con contraluz</i> . . . . .	45
Figura 15. <i>Inspección de cojinetes con iluminación axial difusa</i> . . . . .	46
Figura 16. <i>Inspección con iluminación difusa tipo domo</i> . . . . .	46
Figura 17. <i>Etapas en el procesamiento de imágenes</i> . . . . .	51
Figura 18. <i>Estructura de una red neuronal</i> . . . . .	57
Figura 19. <i>Arquitectura en capas de la corteza visual humana y las capas de una red neuronal convolucional</i> . . . . .	58
Figura 20. <i>Tipos y clases de vehículos no tripulados</i> . . . . .	66
Figura 21. <i>Modelo en cascada</i> . . . . .	67
Figura 22. <i>Metodología en cascada aplicada al sistema</i> . . . . .	70
Figura 23. <i>Diagrama de bloques del sistema</i> . . . . .	87
Figura 24. <i>Arquitectura del sistema</i> . . . . .	89
Figura 25. <i>Diagrama de flujo del sistema</i> . . . . .	90
Figura 26. <i>Diagrama de bloques de la recolección de datos</i> . . . . .	91
Figura 27. <i>Diagrama de flujo del plan de vuelo</i> . . . . .	91
Figura 28. <i>Ortomapa del plan de vuelo</i> . . . . .	92
Figura 29. <i>Diseño de la ruta de vuelo en Google Earth</i> . . . . .	96
Figura 30. <i>Interfaz de planificación de ruta de vuelo en Litchi</i> . . . . .	97
Figura 31. <i>Visualización del recorrido de la ruta mediante waypoints en Python</i> . . . . .	98
Figura 32. <i>Diagrama de flujo de la ejecución de vuelo</i> . . . . .	99
Figura 33. <i>Punto de inicio para la ejecución del vuelo</i> . . . . .	99
Figura 34. <i>Inicio de pruebas de vuelo en el invernadero</i> . . . . .	100
Figura 35. <i>Diagrama de flujo de la captura de imágenes</i> . . . . .	101
Figura 36. <i>Captura de imagen del dron.</i> . . . . .	101
Figura 37. <i>Diagrama de flujo del almacenamiento local de datos</i> . . . . .	102
Figura 38. <i>Fotografía del cultivo de rosas almacenada en la memoria local del dron.</i> . . . . .	103
Figura 39. <i>Diagrama de bloques del procesamiento de la información</i> . . . . .	103
Figura 40. <i>Diagrama de flujo del proceso de captura de datos</i> . . . . .	104
Figura 41. <i>Imágenes capturadas para el dataset de roya</i> . . . . .	104
Figura 42. <i>Organización del dataset de roya por niveles de severidad</i> . . . . .	105
Figura 43. <i>Diagrama de bloques del proceso de preprocesamiento</i> . . . . .	106

Figura 44. Diagrama del proceso de preprocesamiento . . . . .	107
Figura 45. Diagrama de flujo del proceso de segmentación . . . . .	107
Figura 46. Diagrama de flujo del proceso de segmentación . . . . .	108
Figura 47. Diagrama de flujo del proceso de identificación de objetos . . . . .	109
Figura 48. Flujo del proceso de entrenamiento del modelo YOLOv5s en Google Colab . . . . .	110
Figura 49. Arquitectura general del modelo YOLOv5 . . . . .	112
Figura 50. Diagrama de flujo del tratamiento de los datos . . . . .	113
Figura 51. Modelo entidad-relación para el tratamiento de datos . . . . .	114
Figura 52. Consulta de registros en la tabla archivos de la base de datos MySQL . . . . .	114
Figura 53. Fragmento de código para registrar archivos en la base de datos . . . . .	115
Figura 54. Fragmento de código para el análisis y clasificación de severidad . . . . .	118
Figura 55. Fragmento de código para la obtención de datos climáticos en tiempo real . . . . .	119
Figura 56. Diagrama de bloques de la visualización de resultados . . . . .	119
Figura 57. Diagrama de flujo de la interfaz de usuario . . . . .	120
Figura 58. Interfaz de usuario . . . . .	120
Figura 59. Diagrama de flujo de la presentación de resultados . . . . .	122
Figura 60. Resultados de detección en nivel Crítico . . . . .	123
Figura 61. Diagrama de bloques de la generación de reportes . . . . .	124
Figura 62. Fragmento de código para la generación de reportes en PDF . . . . .	125
Figura 63. Resultados de los archivos generados del modelo de detección . . . . .	126
Figura 64. Validación de las características de la batería del dron DJI Air 3 . . . . .	128
Figura 65. Verificación del voltaje de la batería . . . . .	128
Figura 66. Validación de los parámetros de la cámara del dron DJI Air 3 . . . . .	130
Figura 67. Validación del funcionamiento de la cámara del dron . . . . .	130
Figura 68. Validación del funcionamiento de geolocalización del GPS . . . . .	131
Figura 69. Validación del recorrido del dron y captura de imágenes en cada way-point . . . . .	132
Figura 70. Verificación de la configuración de seguridad del dron antes del vuelo . . . . .	133
Figura 71. Validación en la ejecución del vuelo y captura de imágenes en los waypoints establecidos . . . . .	133
Figura 72. Validación en la visualización de las imágenes capturadas en la galería del dron . . . . .	134
Figura 73. Verificación de la ejecución del modelo YOLOv5 . . . . .	135
Figura 74. Verificación del procesamiento de imágenes en el sistema . . . . .	135
Figura 75. Clasificación de detecciones por nivel de severidad por waypoints . . . . .	136
Figura 76. Verificación del almacenamiento de datos en la base de datos MySQL . . . . .	137
Figura 77. Verificación de los resultados de detección en la interfaz web. . . . .	137
Figura 78. Verificación de los resultados de detección en la interfaz web. . . . .	138
Figura 79. Verificación de los resultados de detección en la interfaz web. . . . .	139
Figura 80. Mapa de severidad del cultivo – Prueba 1 aérea . . . . .	140
Figura 81. Niveles severidad del cultivo – Prueba 1 aérea . . . . .	141
Figura 82. Análisis de severidad del cultivo – Prueba 1 aérea . . . . .	141
Figura 83. Mapa de severidad del cultivo – Prueba 2 aérea . . . . .	142
Figura 84. Niveles severidad del cultivo – Prueba 2 aérea . . . . .	143
Figura 85. Análisis de severidad del cultivo – Prueba 2 aérea . . . . .	143
Figura 86. Mapa de severidad del cultivo – Prueba 3 aérea . . . . .	144
Figura 87. Niveles severidad del cultivo – Prueba 3 aérea . . . . .	144
Figura 88. Análisis de severidad del cultivo – Prueba 3 aérea . . . . .	145
Figura 89. Mapa de severidad del cultivo – Prueba 1 terrestre . . . . .	145
Figura 90. Niveles severidad del cultivo – Prueba 1 terrestre . . . . .	146

Figura 91. <i>Análisis de severidad del cultivo – Prueba 1 terrestre</i>	146
Figura 92. <i>Mapa de severidad del cultivo – Prueba 2 terrestre</i>	147
Figura 93. <i>Niveles severidad del cultivo – Prueba 2 terrestre</i>	147
Figura 94. <i>Análisis de severidad del cultivo – Prueba 2 terrestre</i>	148
Figura 95. <i>Mapa de severidad del cultivo – Prueba 3 terrestre</i>	148
Figura 96. <i>Niveles severidad del cultivo – Prueba 3 terrestre</i>	149
Figura 97. <i>Análisis de severidad del cultivo – Prueba 3 terrestre</i>	149
Figura 98. <i>Validación de precisión de las detecciones en el cultivo</i>	150
Figura 99. <i>Validación de precisión de las detecciones en el cultivo</i>	151
Figura 100. <i>Tabulación de la pregunta 1</i>	159
Figura 101. <i>Tabulación de la pregunta 2</i>	160
Figura 102. <i>Tabulación de la pregunta 3</i>	161
Figura 103. <i>Tabulación de la pregunta 4</i>	161
Figura 104. <i>Tabulación de la pregunta 5</i>	162
Figura 105. <i>Tabulación de la pregunta 6</i>	163
Figura 106. <i>Tabulación de la pregunta 7</i>	163
Figura 107. <i>Tabulación de la pregunta 8</i>	164
Figura 108. <i>Tabulación de la pregunta 9</i>	164
Figura 109. <i>Tabulación de la pregunta 10</i>	165
Figura 110. <i>Tabulación de la pregunta 11</i>	166
Figura 111. <i>Tabulación de la pregunta 12</i>	166
Figura 112. <i>Tabulación de la pregunta 13</i>	167
Figura 113. <i>Tabulación de la pregunta 14</i>	167
Figura 114. <i>Tabulación de la pregunta 15</i>	168
Figura 115. <i>Localización del campo del cultivo de rosas</i>	168
Figura 116. <i>Especificaciones técnicas del hardware del equipo de procesamiento (CPU y GPU)</i>	172
Figura 117. <i>Diagrama esquemático del dron DJI Air 3 y sus componentes principales</i>	172
Figura 118. <i>Arquitectura del núcleo del intérprete de Python y sus fases de procesamiento</i>	175
Figura 119. <i>Arquitectura general del modelo YOLOv5 para detección de objetos en una sola etapa</i>	176
Figura 120. <i>Arquitectura general de la aplicación web desarrollada en Flask</i>	177
Figura 121. <i>Arquitectura general del motor de base de datos MySQL y sus componentes internos</i>	177
Figura 122. <i>Interfaz principal de la aplicación Litchi y sus elementos de control de vuelo</i>	178
Figura 123. <i>Medición de la distancia entre waypoints en el cultivo</i>	179
Figura 124. <i>Captura de coordenadas con la aplicación GPS Logger</i>	179
Figura 125. <i>Archivo generado por la aplicación GPS Logger</i>	180
Figura 126. <i>Visualización de coordenadas exportadas desde GPS Logger</i>	180
Figura 127. <i>Organización de coordenadas y parámetros de vuelo en Excel</i>	181
Figura 128. <i>ArcMap 10.8</i>	182
Figura 129. <i>Carga del archivo de coordenadas en ArcMap</i>	182
Figura 130. <i>Visualización de coordenadas en ArcMap mediante Display XY Data</i>	183
Figura 131. <i>Configuración de campos de coordenadas en ArcMap</i>	183
Figura 132. <i>Ruta de vuelo generada en ArcMap a partir de los waypoints</i>	184
Figura 133. <i>Creación de un nuevo proyecto en Roboflow</i>	185
Figura 134. <i>Configuración del proyecto de detección de roya en Roboflow</i>	185
Figura 135. <i>Organización y carga del dataset de roya en Roboflow</i>	185
Figura 136. <i>Panel de anotaciones y progreso de etiquetado en Roboflow</i>	186

Figura 137. <i>Definición de clases de severidad de roya en Roboflow</i> . . . . .	186
Figura 138. <i>Segmentación del área afectada por roya en Roboflow</i> . . . . .	187
Figura 139. <i>Parámetros del preprocesamiento en Roboflow</i> . . . . .	188
Figura 140. <i>Distribución del dataset etiquetado para el entrenamiento del modelo</i> . .	189
Figura 141. <i>Descarga del dataset etiquetado en formato compatible con YOLOv5</i> . .	189
Figura 142. <i>Repositorio oficial de YOLOv5 en GitHub</i> . . . . .	190
Figura 143. <i>Interfaz de trabajo de Google Colab con el notebook de YOLOv5</i> . . . .	191
Figura 144. <i>Selección de GPU en Google Colab</i> . . . . .	191
Figura 145. <i>Clonación del repositorio de YOLOv5 en Google Colab</i> . . . . .	192
Figura 146. <i>Importación del dataset desde Google Drive hacia Google Colab</i> . . . .	192
Figura 147. <i>Verificación de la carga del dataset en Google Colab</i> . . . . .	193
Figura 148. <i>Ejecución del entrenamiento y parámetros configurados en YOLOv5</i> . .	193
Figura 149. <i>Descarga del modelo entrenado en Google Colab</i> . . . . .	194

## ÍNDICE DE TABLAS

Tabla 1	<i>Parámetros de agricultura de precisión para el cultivo de rosas</i> . . . . .	29
Tabla 2	<i>Los diez principales exportadores de flores a nivel mundial</i> . . . . .	30
Tabla 3	<i>Ciclo de vida de la roya del rosal y los tipos que existen</i> . . . . .	35
Tabla 4	<i>Pasos en el procesamiento de imágenes</i> . . . . .	48
Tabla 5	<i>Componentes en el procesamiento de imágenes</i> . . . . .	52
Tabla 6	<i>Etapas en el reconocimiento de patrones</i> . . . . .	53
Tabla 7	<i>Técnicas más comunes de reconocimiento de patrones</i> . . . . .	54
Tabla 8	<i>Ventajas de las redes neuronales artificiales</i> . . . . .	55
Tabla 9	<i>Tipos de redes neuronales y métodos de aprendizaje</i> . . . . .	58
Tabla 10	Avances en arquitecturas de redes neuronales y sus aplicaciones . . . . .	60
Tabla 11	Ventajas de los drones agrícolas . . . . .	64
Tabla 12	Clasificación de Drones . . . . .	65
Tabla 13	Método de Investigación y Encuesta . . . . .	72
Tabla 14	<i>Lista de Stakeholders</i> . . . . .	74
Tabla 15	Abreviatura de los requerimientos . . . . .	75
Tabla 16	Requerimientos Operacionales y de Usuarios . . . . .	76
Tabla 17	Requerimientos del Sistema (SySR) . . . . .	77
Tabla 18	Requerimientos de Arquitectura (SRSH) . . . . .	78
Tabla 19	Selección de la placa de desarrollo . . . . .	79
Tabla 20	Selección del Dron Profesional . . . . .	80
Tabla 21	Selección del Sistema Operativo . . . . .	81
Tabla 22	Selección de entorno de programación . . . . .	82
Tabla 23	Selección del lenguaje de programación . . . . .	83
Tabla 24	Selección de algoritmos de visión artificial . . . . .	84
Tabla 25	Selección de la aplicación web . . . . .	84
Tabla 26	Selección de la Base de Datos . . . . .	85
Tabla 27	Selección de la aplicación de planificación de vuelo . . . . .	86
Tabla 28	Coordenadas y parámetros de vuelo del dron . . . . .	93
Tabla 29	<i>Escala para evaluar el grado de severidad de <i>Phragmidium mucronatum</i> en cultivos de rosales.</i> . . . . .	116
Tabla 30	<i>Matriz de confusión para la clasificación automática del grado de severidad de <i>Phragmidium mucronatum</i>.</i> . . . . .	117
Tabla 31	Validación de hardware del dron . . . . .	127
Tabla 32	Validación en la ejecución del vuelo y captura de imágenes . . . . .	131
Tabla 33	Validación del modelo Yolov5 . . . . .	134
Tabla 34	Validación de reportes . . . . .	138
Tabla 35	Test de funcionamiento en escenario real de cultivo . . . . .	139
Tabla 36	Test de validación de precisión de las detecciones por escenario . . . . .	150
Tabla 37	Validación del desempeño del sistema de detección . . . . .	151
Tabla 38	Tabla de verificación de los requerimientos de Stakeholders . . . . .	169
Tabla 39	Tabla de verificación de los requerimientos del sistema (SySR) . . . . .	170
Tabla 40	Tabla de verificación de los requerimientos de arquitectura (SRSH) . . . . .	171

## RESUMEN

La producción de rosas es una actividad muy importante dentro del sector agrícola y de exportación en el Ecuador, sin embargo, se ve afectada por enfermedades como la roya del rosal (*Phragmidium mucronatum*), que puede causar grandes pérdidas económicas si no se identifica a tiempo. Tradicionalmente, la identificación de esta enfermedad se realiza mediante inspecciones visuales manuales, un proceso que resulta subjetivo, depende en gran medida de la experiencia del personal técnico y requiere un considerable consumo de tiempo.

Como respuesta a este problema se realiza el diseño de un prototipo que permita la detección de roya en cultivos de rosas utilizando técnicas de visión artificial basadas en redes neuronales convolucionales y el uso de un dron. Además, el sistema permite capturar imágenes aéreas del cultivo para procesarlas y detectar la enfermedad mediante el modelo YOLOv5 y mostrar los resultados a través de una aplicación web.

Los resultados obtenidos permiten evidenciar que el prototipo ayuda a clasificar y detectar con los diferentes niveles de gravedad de la enfermedad, esto permite mejorar el control de la salud del cultivo y reducir de forma significativa el tiempo necesario para las inspecciones. De esta manera, la propuesta se presenta como una herramienta útil para la agricultura de precisión, ya que ayuda a tomar mejores decisiones y a usar de forma más eficiente los insumos agrícolas.

**Palabras clave:** agricultura de precisión, roya del rosal, visión artificial, drones, redes neuronales convolucionales.

## ABSTRACT

Rose production is a very important activity within Ecuador's agricultural and export sector; however, it is affected by diseases such as rose rust (*Phragmidium mucronatum*), which can cause significant economic losses if not identified in time. Traditionally, this disease is identified through manual visual inspections, a subjective process that depends heavily on the experience of the technical staff and is considerably time-consuming.

In response to this problem, a prototype was designed to detect rust in rose crops using computer vision techniques based on convolutional neural networks and the use of a drone. The system also captures aerial images of the crop for processing and disease detection using the YOLOv5 model, displaying the results through a web application.

The results obtained demonstrate that the prototype helps classify and detect the disease at different severity levels, improving crop health management and significantly reducing the time required for inspections. In this way, the proposal is presented as a useful tool for precision agriculture, since it helps to make better decisions and to use agricultural inputs more efficiently.

**Keywords:** precision agriculture, rose rust, computer vision, drones, convolutional neural networks.

# CAPÍTULO I

## INTRODUCCIÓN

### 1.1 Planteamiento del Problema

La roya (*Phragmidium mucronatum*) es una de las principales enfermedades dentro de los cultivos de rosas en el Ecuador, debido a que al ser el principal exportador de rosas de alta calidad a nivel mundial. De acuerdo con Expoflores el sector florícola genera alrededor del 9 % del PIB agrícola nacional y da empleo a más de 110.000 personas, beneficiando indirectamente a más de 400.000 familias. Enfermedades como la roya generan un gran impacto en la producción de rosas lo que causa pérdidas económicas que van entre el 15 % y el 40 % en campos severamente infestados (Exflores, 2025).

Esta enfermedad es causada por hongos, se inicia con manchas anaranjadas y amarillas en las hojas, defoliación, debilitamiento de la planta y menor floración, lo que causa en situaciones extremas pérdidas totales en zonas no tratadas. Este problema surge por condiciones ambientales como temperaturas moderadas (18-25°C) y alta humedad (más del 80 %), que favorecen la aparición de la roya Ministerio de Agricultura, Pesca y Alimentación de España, 2025.

El control de la roya generalmente se realiza con fungicidas, lo que aumenta los costos de producción debido al uso constante de estos químicos se estima que más del 20 % del presupuesto en floricultura se destina al control de plagas y enfermedades. Además, diversos estudios muestran que hasta un 30 % de estas aplicaciones químicas podrían evitarse si la enfermedad se detectara de manera temprana y precisa (Da Jiang Innovations, 2024).

No obstante, los métodos contemporáneos para la detección de la roya presentan conside-

rables restricciones. Los métodos convencionales, tales como la inspección visual realizada por personal de campo, son lentos, onerosos e inexactos, especialmente en el caso de cultivos de carácter extensivo. En este punto, las técnicas de visión artificial implementadas en Python y el procesamiento de imágenes pueden ofrecer soluciones más eficaces, identificando la enfermedad en sus fases iniciales con exactitud y posibilitando la implementación de medidas preventivas tempranas que reduzcan el impacto en la producción (Pardo, 2021).

## **1.2 Objetivos**

### **1.2.1 Objetivo General**

Implementar un prototipo mediante un dron para la identificación de la plaga de Roya (*Phragmidium mucronatum*) en cultivos de rosales, optimizando la precisión en su detección y en el uso de fungicidas en el tratamiento de dichos cultivos.

### **1.2.2 Objetivos Específicos**

- Investigar las principales fitopatologías que afectan a los cultivos de rosas para identificar sus características, causas y efectos en la producción.
- Identificar los requerimientos para el diseño del sistema mediante la metodología en cascada para un desarrollo secuencial y ordenado.
- Diseñar un prototipo para la detección de la plaga de Roya en cultivos de rosas mediante la construcción de un dron con visión artificial.
- Verificar el funcionamiento del sistema en condiciones reales que demuestre su efectividad en la detección de Roya y su impacto en el uso de fungicidas.

### 1.3 Alcance y delimitación

Los recursos bibliográficos es parte esencial para establecer las bases teóricas del proyecto. Por esta razón, se realizará una revisión detallada para el marco teórico relacionada con enfermedades fúngicas en cultivos agrícolas y sistemas tecnológicos de visión artificial aplicados a la detección de plagas. Esta revisión incluirá artículos científicos, tesis y estudios similares que permitan identificar técnicas, herramientas y metodologías aplicables al desarrollo del sistema propuesto.

En la primera fase de requisitos, se establecerá el marco teórico necesario para el desarrollo del sistema propuesto. Se llevará a cabo una revisión detallada de literatura científica que servirá como base para identificar las características biológicas de la roya (*Phragmidium mucronatum*), los factores ambientales que favorecen su aparición y los principios fundamentales de visión artificial aplicados a la agricultura de precisión.

En la segunda fase de diseño, se llevará a cabo un análisis detallado de los requerimientos de stakeholders, con el objetivo de garantizar que el sistema cumpla con las expectativas del proyecto. Este análisis permitirá identificar los requerimientos de rendimiento que el sistema debe cumplir para el uso en el entorno agrícola. El diseño de la arquitectura general del sistema es un aspecto crucial, ya que establece la base estructural que garantizará el correcto funcionamiento y desempeño del dron en el campo agrícola. Una arquitectura bien planificada permite integrar de manera eficiente los componentes de hardware y software para cumplir con los objetivos del proyecto. Además de incorporar algoritmos avanzados de visión artificial y módulos de comunicación, asegurando un procesamiento preciso y una transmisión de datos confiable. Este enfoque no solo maximiza la eficiencia operativa del sistema. La arquitectura del sistema se organizará en cuatro bloques funcionales: Recolección de datos, Procesamiento de la información, Tratamiento de los datos y Visualización de los resultados. En la tercera fase de implementación, el sistema se centrará en la transmisión de los datos recolectados por el dron hacia el sistema de procesamiento. Esto incluirá la selección de tecnologías de comunicación óptimas para garantizar una amplia cobertura en la zona de monitoreo en los cultivos

de rosas. Los datos procesados serán almacenados y analizados en una base de datos, mientras que los resultados serán presentados a los usuarios a través de una aplicación con interfaz visual, permitiendo un monitoreo claro y efectivo de las zonas afectadas.

**Recolección de datos:** En esta etapa inicial, un dron equipado con una cámara se encargará de capturar imágenes de las zonas de cultivo. Estas imágenes permiten identificar tanto las áreas afectadas por la roya, caracterizadas por manchas anaranjadas en las hojas, como las zonas en buen estado. El dron envía los datos recolectados al sistema para su posterior análisis.

**Procesamiento de la información:** Las imágenes enviadas son procesadas en un sistema AI especializado. Durante esta etapa, se emplean algoritmos avanzados de visión artificial para analizar las imágenes capturadas e identificar la presencia de la roya en las áreas afectadas del cultivo. Este procesamiento garantiza que se identifique de forma precisa.

**Tratamiento de los datos:** Una vez procesadas las imágenes, estas se almacenan en una base de datos central donde se organizan y analizan. Este análisis identifica patrones, localiza zonas críticas afectadas y genera información útil para apoyar la toma de decisiones y la visualización de resultados.

**Visualización de resultados:** Los datos tratados son integrados en una aplicación que proporciona una interfaz visual intuitiva. A través de gráficos, métricas y alertas, los usuarios pueden monitorear el estado de los cultivos.

En la fase final de verificación, se valida el sistema en un entorno real, realizando pruebas exhaustivas en los cultivos de rosas para evaluar su desempeño. Se mide la precisión del dron y del software para detectar la roya, comparándolo con métodos tradicionales de inspección. Además, se analiza el impacto del sistema en la optimización del uso de fungicidas, buscando reducir su aplicación innecesaria y fomentar prácticas agrícolas sostenibles. Las pruebas también incluyen la evaluación de la respuesta del sistema ante diferentes escenarios, asegurando que cumpla con los objetivos de mejorar la productividad y reducir costos operativos, mientras se contribuye a la sostenibilidad agrícola.

## 1.4 Justificación

El presente proyecto, que propone el diseño de un sistema para la detección de la roya (*Phragmidium mucronatum*) en cultivos de rosas mediante el uso de un dron implementado con visión artificial, se encuentra alineado con los Objetivos de Desarrollo Sostenible (ODS), particularmente con los ODS 9 y 12. La integración de tecnologías avanzadas en la agricultura no solo responde a la necesidad de modernizar el sector, sino que también fomenta la innovación y la adopción de soluciones sostenibles que pueden transformar la manera en que se gestionan los cultivos, contribuyendo al desarrollo económico y ambiental de las comunidades agrícolas. Por lo tanto, este proyecto busca promover un enfoque más eficiente y responsable en la producción agrícola, alineándose con la visión de un futuro más sostenible y tecnológicamente avanzado. (Martínez & Castro, 2020)

El ODS 9: Industria, Innovación e Infraestructura promueve la incorporación de tecnologías avanzadas para modernizar sectores clave de la economía, y en este caso, el uso de un dron y algoritmos de visión artificial transforma los métodos tradicionales de inspección agrícola, facilitando la detección temprana y precisa de enfermedades y optimizando los procesos de manejo de cultivos. Esto fortalece la infraestructura tecnológica en el sector agrícola y posiciona a la industria como un motor de innovación y competitividad. (ENAE Business School, 2024).

Por otro lado, el ODS 12: Producción y Consumo Responsables busca garantizar modalidades de consumo y producción sostenibles. Este proyecto contribuye a este objetivo al permitir una aplicación más eficiente de fungicidas, evitando el uso excesivo y protegiendo los recursos naturales como el suelo y el agua, al reducir la dependencia de productos químicos. Además, al optimizar la gestión de los cultivos, se minimiza el riesgo de pérdidas en la producción de rosas y se fomenta un modelo de producción que prioriza la sostenibilidad ambiental promoviendo un uso más consciente en el sector agrícola. (ENAE Business School, 2024).

La detección temprana de enfermedades es fundamental para prevenir pérdidas significati-

vas en la producción agrícola. Según la FAO, las plagas y enfermedades pueden causar pérdidas de entre el 20 % y el 40 % de la producción agrícola mundial, lo que subraya la importancia de implementar herramientas tecnológicas que permitan una intervención rápida y precisa. El uso de drones con capacidad de análisis de imágenes permite identificar áreas afectadas de manera eficiente y con menor esfuerzo humano, lo que se traduce en una gestión más precisa y menos dependiente de métodos invasivos o el uso excesivo de productos químicos. Esto no solo optimiza el manejo de los cultivos, sino que también contribuye a la reducción de costos operativos y a la protección de los recursos naturales (FAO, 2022). En el caso específico de la Roya (*Phragmidium mucronatum*) en cultivos de rosas, un sistema basado en un dron con visión artificial puede detectar signos de la enfermedad, minimizando el uso de fungicidas, reduciendo costos y promoviendo prácticas sostenibles. (Fernández & Sánchez, 2018)

## **CAPÍTULO II**

### **MARCO TEÓRICO**

#### **2.1 Cultivo de Rosas**

La rosa es una planta muy cautivadora y de gran atractivo con un inmenso valor ornamental y económico por su apariencia que ha cautivado a la sociedad desde tiempos inmemorables. Como expresa Yong (2004) la rosa es perteneciente a la familia de las Rosáceas, se destaca como un símbolo de belleza, adorado por egipcios, romanos, griegos y sirios. En la actualidad, esta maravilla ornamental ocupa un espacio destacado como una de las especies más demandadas, apreciada por sus llamativas flores, ideales para ramos y adornos. Su encanto inigualable la consolidan como una favorita del público. la rosa tiene usos en la medicina, la gastronomía e incluso en la extracción de aceites esenciales para la cosmética y la perfumería. Ninguna otra flor ornamental ha gozado de una admiración como lo es la rosa . El cultivo de rosas requiere de un ambiente adecuado y cuidados específicos, pues así se consiguen flores saludables y de buena calidad como lo indica el Ministerio de Agricultura Ganadería y Alimentación de Guatemala (2024).

##### **2.1.1 Principales características de los cultivos de rosas**

Cultivar rosas es una actividad agrícola relevante, ya que tiene valor tanto comercial como ornamental. Es fundamental comprender y poner en práctica los principios que rigen el manejo eficaz del cultivo, las exigencias medioambientales y las pautas relacionadas con la agricultura de precisión para asegurar un desarrollo adecuado de las plantas y una floración de alta calidad.

### **2.1.1.1 Requerimientos ambientales para el cultivo de rosas**

La mayoría de los cultivos de rosas necesitan temperaturas ideales entre 17°C y 25°C para garantizar un crecimiento apropiado y una producción eficaz, según Vásquez Tubón (2013). Sin embargo, si la temperatura baja a 15°C por la noche o sube hasta 28°C durante el día, el crecimiento de las rosas podría verse perjudicado por condiciones ambientales desfavorables. Asimismo, es esencial asegurar que se exponga a la luz durante al menos seis horas cada día. Con respecto a la altitud, para favorecer el óptimo desarrollo de las rosas se aconseja que su plantación tenga lugar entre 500 y 2800 metros sobre el nivel del mar.

### **2.1.1.2 Manejo del cultivo**

Steve Ramsay (2021) destaca que la implementación de buenas prácticas en el manejo del cultivo de rosas requiere un sistema eficiente de drenaje y aireación, lo que permite mantener un pH óptimo entre 5 y 6.5. Antes de la siembra, es recomendable realizar un análisis del suelo para garantizar una fertilización de fondo adecuada.

La producción de rosas generalmente se lleva a cabo en platabandas o camas, cuyas dimensiones varían en función del tamaño del invernadero. La densidad de las plantas depende de estos factores, aunque se sugiere distribuirlas en dos filas con distancias de 40 x 20 cm o 60 x 12.5 cm, dejando pasillos de al menos un metro para facilitar su mantenimiento y optimizar los costos de producción. En términos de densidad de plantación, se estima un rango entre 25,000 y 30,000 plantas por hectárea, dependiendo del método de siembra empleado (Steve Ramsay, 2021).

La Figura 1 muestra con claridad y de forma precisa los elementos esenciales del cultivo de rosas, que incluyen las condiciones ambientales ideales y las prácticas de manejo sugeridas para su correcto crecimiento.

**Figura 1.**

*Producción de rosas en invernadero*



*Nota.* La fotografía muestra las características de la plantación de rosas en un invernadero, como se indicó previamente. Se tomó de Fernandez Acevedo (2022).

### **2.1.2 Parámetros aplicados en agricultura de precisión**

La agricultura de precisión tiene un rol esencial en la transformación del sector agrícola, brindando enormes beneficios para optimizar la eficiencia y sostenibilidad, según lo que dice Rodolfo Ríos Hernández (2021). Esta práctica permite un mayor conocimiento de las propiedades del suelo, lo que ayuda a lograr niveles de productividad más elevados. Asimismo, ayuda a optimizar el empleo de los recursos, disminuir los costos de operación y reducir el impacto sobre el medio ambiente. Utilizar distintas tecnologías que mejoran los procedimientos agrícolas es un elemento crucial de la agricultura de precisión.

La Tabla 1 presenta los principales parámetros considerados para la implementación de la agricultura de precisión en el cultivo de rosas. Estos elementos permiten mejorar la gestión del cultivo, favoreciendo un desarrollo adecuado y una floración de calidad, mediante el uso de tecnologías que facilitan el monitoreo y la optimización precisa de las condiciones de producción.

**Tabla 1.***Parámetros de agricultura de precisión para el cultivo de rosas*

<b>Parámetro</b>	<b>Descripción</b>
<b>Temperatura</b>	Las rosas prefieren temperaturas entre 17°C y 25°C. Temperaturas por debajo de 15°C por la noche o superiores a 28°C durante el día pueden afectar su crecimiento y la calidad de las flores.
<b>Humedad</b>	Para un correcto crecimiento, la humedad relativa debe estar entre el 60 % y el 70 %. Para fomentar el crecimiento, tras la plantación, es necesario elevar temporalmente el nivel de humedad entre un 85 % y un 90 %.
<b>Luminosidad</b>	La cantidad y la calidad de la luz son esenciales. La escasa luz produce tallos que son delgados y flexibles. La producción de rosas aumenta en invierno por la escasa intensidad de la luz del sol.
<b>Sustrato</b>	El suelo tiene que estar bien ventilado y tener la capacidad de evacuar el agua de manera eficaz. Para prevenir la acumulación de sales, el pH debe oscilar entre 6 y 7, mientras que la conductividad eléctrica no tiene que exceder los 0,1 a 0,2 dS/m.

**Nota.** La tabla resume las variables más relevantes asociadas a la agricultura de precisión aplicadas al cultivo de rosas, entre las que se incluyen la temperatura, la humedad, la luminosidad y las características del sustrato, factores que influyen directamente en el crecimiento y desarrollo adecuado de la planta. Adaptado de Rodolfo Ríos Hernández (2021).

### **2.1.3 Análisis general de la producción de rosas a nivel mundial y en Ecuador**

Tanto la economía como el medio ambiente y la cultura dependen de la producción de rosas. En naciones como Ecuador, que es un mercado relevante para el Producto Interno Bruto (PIB) y las exportaciones (en particular hacia Estados Unidos), constituye una fuente significativa de ingresos y empleo. Además, las rosas poseen un significado simbólico profundo en muchas culturas por ser parte de celebraciones y momentos especiales, lo que estimula la demanda mundial. La industria de las flores, en lo concerniente a la ecología, está aplicando progresivamente métodos sustentables con el fin de reducir su impacto, como por ejemplo el empleo eficaz de los recursos naturales.

#### **2.1.3.1 Producción mundial de rosas**

La producción de rosas es una de las actividades agrícolas más relevantes en el comercio internacional de flores, gracias a su alta demanda y valor exportador. Además de su atracti-

vo estético, las rosas tienen aplicaciones en la medicina y la industria cosmética. Países como Ecuador, Colombia y los Países Bajos han incorporado tecnología y optimización de recursos naturales para mantener una producción eficiente, generando empleo y fortaleciendo sus economías.

De acuerdo con Ekos Negocios (2024), las exportaciones de flores ecuatorianas entre enero y noviembre de 2018 ascendieron a 792,7 millones de dólares, lo que representó el 4 % del total de sus exportaciones. A escala global, los Países Bajos van a la cabeza con el 49 % del mercado, luego están Colombia y Ecuador con el 16 % y el 10 %, respectivamente. Estos tres países juntos representan el 75 % del comercio mundial de flores.

La Tabla 2 muestra de manera destacada a los principales países exportadores de rosas a nivel mundial, resaltando su participación en el mercado global.

**Tabla 2.**

*Los diez principales exportadores de flores a nivel mundial*

Ranking	País	Valor exportaciones (millones de USD)	Cantidad exportada (miles de toneladas)	Tasa de crecimiento Anual	Participación global (%)
1	Holanda	4.207,0	591,6	1 %	48,7 %
2	Colombia	1.399,6	246,1	7 %	16,2 %
3	Ecuador	881,5	159,0	10 %	10,2 %
4	Kenia	540,9	143,9	6 %	6,3 %
5	Etiopía	196,6	48,3	3 %	2,3 %
6	Malasia	107,5	41,1	3 %	1,3 %
7	China	103,9	32,1	2 %	1,2 %
8	Italia	98,2	11,2	5 %	1,1 %
9	Bélgica	98,8	11,8	-1 %	1,1 %
10	Alemania	72,2	12,2	-9 %	0,8 %
<b>Total mundial</b>		8.639,8	–	3 %	100 %

**Nota.** La tabla presenta los diez principales países exportadores de flores a nivel mundial, destacando su valor de exportación, volumen comercializado, tasa de crecimiento anual y participación global en el mercado. Adaptado de Ekos Negocios (2024).

### 2.1.3.2 Producción de rosas en Ecuador

Ecuador se posiciona como uno de los principales exportadores de rosas a nivel mundial, gracias a su extraordinaria diversidad natural y condiciones climáticas privilegiadas. Su ubicación geográfica, cerca de la línea ecuatorial, proporciona una combinación ideal de luz solar,

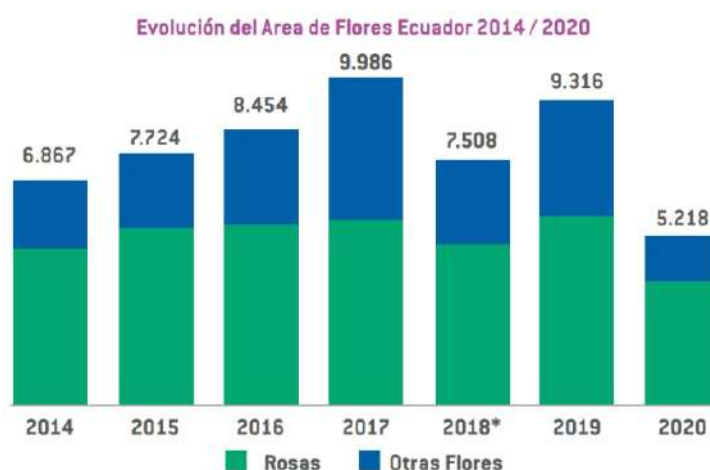
altitud y temperatura que favorece el cultivo de rosas de alta calidad. Gracias a esta riqueza natural y a métodos de producción sofisticados, el país puede ofrecer una oferta significativa en el mercado mundial, convirtiéndose en un referente de la floricultura a nivel internacional.

En Ecuador, de acuerdo con Metroflor (2024), existen más de 1.700 fincas que producen flores y más de 600 compañías dedicadas a su exportación. En 2020, en términos de producción de rosas, ocupó el cuarto lugar más grande en América Latina, con aproximadamente el 10 % del área ornamental regional. El 4 % de la producción se mantiene en el mercado local, y el 96 % restante, que está orientado a flores de corte, es exportado. El sector emplea a 105,000 personas, con una ocupación de 12 empleados por hectárea y una participación del 60 % de mujeres. El porcentaje de superficie cultivada con rosas para exportación oscila entre el 56 % y el 73 %.

La evolución de la superficie dedicada a la producción de rosas y otras flores en Ecuador, desde el 2014 hasta el 2020, se muestra en la Figura 2. Se ha registrado una expansión positiva en los años recientes a lo largo de este lapso, lo cual ha favorecido a la producción y también a las ganancias del país.

**Figura 2.**

*Evolución del Área de Flores y Rosas en el Ecuador*



**Nota.** La evolución del campo de flores en Ecuador entre 2014 y 2020 se presenta mediante un gráfico de barras. Gráfico extraído de Metroflor (2024).

### 2.1.3.3 Evolución de las exportaciones en el Ecuador

De acuerdo con la Corporación Financiera Nacional (2020), Ecuador se posicionó como líder mundial en la exportación de flores, enviando productos a más de 100 países, entre ellos Estados Unidos, la Unión Europea y Rusia. Las rosas representaron el 70 % de las exportaciones florales, con una producción de 3.800 millones de tallos, de los cuales el 68 % correspondió a esta especie. El valor total de exportaciones alcanzó los 827 millones de dólares, consolidando al país como el tercer mayor exportador mundial de flores cortadas (Corporación Financiera Nacional, 2020).

La calidad de las flores ecuatorianas y su afianzamiento en el mercado mundial se observa a través de la evolución de las exportaciones florales, que se presenta en la Figura 3.

**Figura 3.**

*Progreso de las exportaciones de flores en el Ecuador*



**Nota.** El gráfico muestra cómo han cambiado el volumen y el valor de las exportaciones de flores en Ecuador entre 2009 y 2020, con una tendencia de crecimiento sostenido e incrementos anuales moderados. Tomado de Corporación Financiera Nacional (2020).

## **2.2 Roya del Rosal (*phragmidium mucronatum*)**

Como se hace referencia en Pataky (2021) la roya común en rosas es una enfermedad fúngica que fue documentada por primera vez en 1665 y, con el tiempo, se ha expandido ampliamente en diversas regiones del mundo, especialmente en Norteamérica y Europa. También se ha identificado en otros países donde se cultivan rosas ornamentales, lo que indica su capacidad de adaptación a distintos entornos. A pesar de su distribución global, la roya no suele representar una amenaza grave para todos los cultivos, ya que su desarrollo está condicionado por factores ambientales específicos.

Para que el hongo causante de esta patología complete su ciclo vital, necesita niveles apropiados de humedad y temperatura. Para que la infección tenga lugar, es crucial que las esporas hallen condiciones adecuadas para germinar, esto generalmente ocurre en climas templados con elevada humedad relativa. En regiones donde estas condiciones no son constantes, la enfermedad tiende a manifestarse de forma esporádica o con menor intensidad. Sin embargo, en zonas donde la humedad y la temperatura favorecen su proliferación, la roya puede convertirse en un problema recurrente, afectando la salud y el desarrollo de las plantas (Pataky, 2021).

### **2.2.1 Síntomas**

Como se menciona en Bonillo Carrasco (2021) al inicio de la primavera, comienzan a observarse pequeñas manchas brillantes de color rojo anaranjado, conocidas como espermagonios o picnios, sobre la cara superior de las hojas. Poco después, se presentan manchas amarillas denominadas aecios en el reverso. Estas estructuras, de forma casi circular, aumentan de tamaño y liberan esporas polvorientas (aeciosporas) que rompen la epidermis inferior de la hoja. A medida que esto ocurre, su color se vuelve de un naranja. Las manchas, desde la parte de arriba, toman un color amarillento o anaranjado claro.

Además, Bonillo Carrasco (2021) indica que las hojas, otras partes de la planta también pueden verse comprometidas, como los pecíolos, tallos verdes, cálices, sépalos e incluso los

frutos. En los tallos jóvenes, las lesiones tienden a ser alargadas y estrechas. Estas pueden provocar deformaciones, curvaturas y la destrucción de las yemas. De acuerdo con lo que se muestra en la Figura 4, los síntomas que se observan en este periodo del año son fundamentales para detectar precozmente la enfermedad, lo cual posibilita una identificación más precisa y rápida.

**Figura 4.**

*Características de la Roya*



**Nota.** La fotografía muestra hojas de rosal afectadas por roya, donde se observan pústulas anaranjadas características de la enfermedad, causadas por el hongo *Phragmidium mucronatum*. Fotografía tomada de Masso Garden (2024).

### 2.2.1.1 Impacto de la Roya en los Cultivos de Rosa

Según indica Windham (2021), la etapa más crítica y visible de esta enfermedad en los cultivos de rosales es el período de verano de la roya provocada por *Phragmidium mucronatum*. Durante este periodo, se forman en el reverso de las hojas y en los pecíolos unas pequeñas pústulas de tono naranja rojizo, conocidas como uredias. Estas pústulas contienen masas de esporas pulverulentas, llamadas urediosporas, que presentan una coloración amarillo anaranjada brillante. Desde la superficie superior de la hoja, las uredias se perciben como manchas necróticas de forma angular, que pueden alcanzar hasta 1,5 milímetros de longitud.

Como explican Horst y Cloyd (2020), esta fase tiene un comportamiento recurrente, repitiéndose aproximadamente cada 10 a 14 días siempre que las condiciones climáticas sean

favorables. Las urediosporas liberadas durante estos ciclos generan constantes reinfecciones en el follaje, incrementando de forma significativa el daño ocasionado. Las hojas gravemente afectadas tienden a torcerse, marchitarse y caer entre cinco y diez días después de manifestarse los primeros signos. A su vez, los brotes jóvenes pueden presentar deformaciones, adoptar una coloración rojiza y mostrar un crecimiento irregular. En estadios más avanzados, la infección se disemina ampliamente, afectando tanto a los folíolos como a los tallos, lo que debilita notablemente a la planta. Este tipo de manifestación severa es común en regiones de clima templado y húmedo, como las zonas costeras del Pacífico, donde las condiciones favorecen la proliferación del hongo (Horst y Cloyd, 2020).

En la Tabla 3 se presenta de manera detallada el ciclo de vida de la roya del rosal, mostrando las distintas etapas que atraviesa el hongo desde su formación inicial hasta la propagación y reinicio del ciclo en la siguiente temporada.

**Tabla 3.**

*Ciclo de vida de la roya del rosal y los tipos que existen*

<b>Características</b>	<b>Descripción</b>
<b>Ciclo de vida</b>	La roya de los rosales es provocada por un hongo parásito obligado que sigue un ciclo de vida complejo, pasando por hasta cinco fases: espermogonio, ecio, uredio, telio y basidio. Cada fase genera diferentes tipos de esporas.
<b>Etapas del ciclo</b>	Espermagonia: Fase inicial, en la que las picniosporas se desarrollan. Ecio: Genera eciosporas. Uredio: Produce urediosporas, que son las responsables de la difusión activa. Telio: Genera teliosporas. Basidio: Genera basidiosporas para dar inicio de nuevo al ciclo.
<b>Clasificación de roya</b>	Roya heteroica: Para terminar su ciclo de vida, debe infectar dos plantas distintas. Roya autóica: Al infectar un solo tipo de planta, tiene la posibilidad de concluir su ciclo vital, lo cual favorece su propagación.

*Nota.* Adaptado de Santana Mayorga (2020)

### 2.2.1.2 Estrategias de Manejo para el Control de la Roya

Como indica Andrés Ares (2015) para evitar la propagación temprana de la roya y reducir su impacto en los cultivos de rosales, es recomendable retirar las hojas infectadas durante la estación vegetativa. También se sugiere eliminar las hojas más viejas antes de la poda de in-

vierno o a principios de primavera, justo antes de que aparezcan los nuevos brotes. Durante las épocas en que las condiciones ambientales son propicias para el desarrollo de la roya, se debe aplicar fungicidas preventivos cada siete días para mantener el control de la enfermedad (Andrés Ares, 2015).

Utilizar variedades de rosas que sean resistentes a la roya es una táctica beneficiosa, sobre todo en jardinería. Los cultivares y los híbridos de rosa presentan distintos grados de resistencia. Las rosas de té, por ejemplo, son muy resistentes; en cambio, las poliantas, las enredaderas y los híbridos de té tienen también un grado significativo de resistencia. Según lo que se muestra en la Figura 5, los cultivares miniatura generalmente no tienen problemas con la roya.

### **Figura 5.**

*Estrategias de Manejo para el Control de la Roya*



*Nota.* La fotografía muestra labores de manejo fitosanitario en un cultivo de rosas, orientadas al control y prevención de la roya. Fotografía tomada de Andrés Ares (2015).

## **2.3 Visión Artificial**

Tello Marulanda (2023) menciona que el propósito de la visión artificial es permitir que las computadoras y otros dispositivos tecnológicos procesen y analicen la información visual, permitiéndoles reconocer objetos, personas, movimientos, formas y otras características visuales en tiempo real. Para lograrlo, se utilizan técnicas y algoritmos avanzados de procesa-

miento de imágenes, aprendizaje automático y redes neuronales artificiales. Así, numerosas instituciones, incluyendo universidades como empresas, están actualmente investigando formas de mejorar y aplicar los sistemas de procesamiento de imágenes y visión artificial existentes, motivadas por el prometedor potencial de crecimiento en estos ámbitos. El interés y la investigación activa en mejorar y aplicar estos sistemas reflejan el emocionante potencial de crecimiento en la visión artificial y el procesamiento de imágenes (Tello Marulanda, 2023).

### **2.3.1 Técnicas de visión artificial**

Las técnicas de visión artificial comprenden diversos procedimientos que se emplean para labores como la detección de patrones, el reconocimiento de objetos, la segmentación de imágenes, el seguimiento del movimiento y la clasificación de imágenes. Estas técnicas tienen usos en distintos ámbitos, como la seguridad, la agricultura, la robótica y la medicina, entre otros. En estos campos es necesario procesar e extraer información útil de imágenes o secuencias de video. Cada uno de estos métodos ayuda a que los sistemas tengan una mayor habilidad para reaccionar e interpretar datos visuales en tiempo real.

#### **2.3.1.1 Técnica de segmentación**

Borrella (2022) define la segmentación como una técnica clave en la visión artificial, utilizada para analizar y comprender imágenes y videos. Consiste en dividir una imagen digital en varias regiones o grupos de píxeles, conocidos como segmentos.

Los algoritmos diseñados para esta tarea se basan en las propiedades locales del color. Como se indica en la Figura 6, un factor esencial del algoritmo empleado para el procesamiento es su habilidad para distinguir entre los datos de color y los de luminosidad, lo cual es determinante para lograr una segmentación más precisa.

**Figura 6.***Segmentación de una hoja de café*

**Nota.** Ilustración tomada de Borrella, 2022, donde se muestra el proceso de segmentación de una hoja de café en tres etapas: (a) imagen original, (b) imagen binarizada y (c) imagen segmentada que destaca el objeto de interés.

La segmentación de imágenes, como se observa en la Figura 7, permite dividir una imagen en regiones homogéneas basadas en características como el color, la textura o la forma. Este proceso simplifica la información visual, facilitando que los algoritmos identifiquen las áreas más relevantes y reduzcan la complejidad del análisis.

**Figura 7.***Segmentación de regiones en imágenes de guacamayos.*

**Nota.** Ilustración extraída de Aparicio del Río (2005). (a) Fotografía original con guacamayos de varias especies; (b) representación dividida por regiones cromáticas, resaltando la silueta de uno de los especímenes.

### 2.3.1.2 Filtros

En la visión artificial, los filtros juegan un papel crucial en el procesamiento de imágenes, ya que permiten resaltar aspectos importantes y eliminar el ruido o elementos que no son relevantes para el análisis. Como se señala en el estudio de Borrella (2022), los filtros se aplican para modificar los píxeles de una imagen, cambiando aquellos cuya intensidad no coincide con la de sus píxeles vecinos. Los filtros que se usan con más frecuencia en visión artificial son los de suavizado, detección de bordes y umbral. En particular, los filtros más utilizados para el suavizado son el filtro de medias, el filtro Box y el filtro Gaussiano. Estos filtros son particularmente útiles para las tareas de reconocimiento de objetos, dado que ofrecen imágenes más precisas y nítidas, lo cual disminuye los errores y favorece que se identifiquen correctamente los objetos. Los filtros se mencionan siempre en proyectos vinculados a la visión artificial porque son relevantes para asegurar que las imágenes procesadas tengan calidad (Borrella, 2022).

De acuerdo con la propuesta de Vives Garnique et al. (2014), el filtrado de imágenes es un procedimiento esencial en la visión artificial que posibilita optimizar la calidad visual para hacer más sencillo su análisis. A continuación, se explican las clases de filtrado más importantes que se utilizan en este ámbito, analizando sus bases, usos y el impacto que tienen sobre las imágenes.

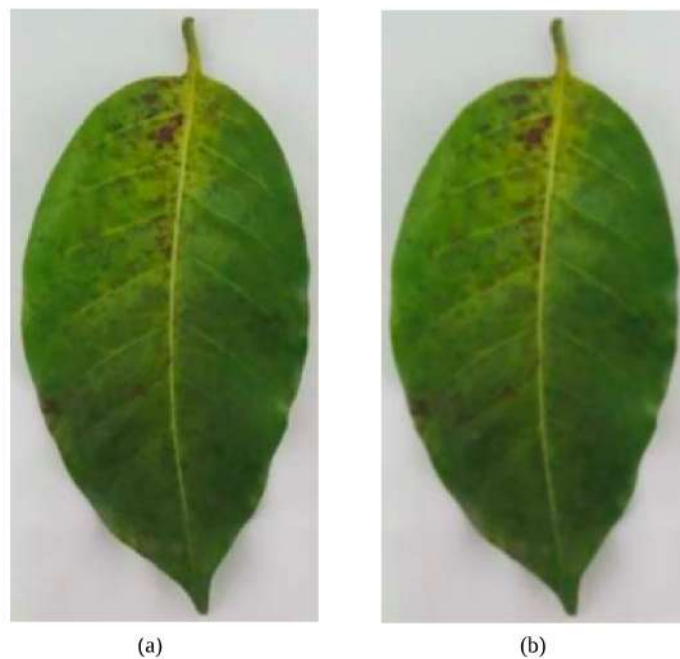
1. **Filtros suavizantes:** Se utilizan en procesamiento de imágenes con el objetivo de atenuar las frecuencias altas, es decir, aquellas asociadas a bordes y detalles finos. Esta reducción permite eliminar el ruido presente en la imagen y obtener una representación más uniforme, útil como etapa previa a la segmentación u otras tareas de análisis. Tal como se señala en la literatura, su uso es esencial para mejorar la calidad visual sin comprometer significativamente la estructura global del objeto de interés.
2. **Filtro de mediana:** Es una técnica no lineal muy empleada para la eliminación de ruido impulsivo. Funciona al ordenar los valores de un vecindario, normalmente de 3x3 píxe-

les, y asignar al píxel central la mediana de dicho conjunto. Este procedimiento ayuda a preservar los bordes y las estructuras más definidas en comparación con los filtros promedios, lo que lo hace especialmente adecuado para imágenes con ruido tipo sal y pimienta. Este filtro es comúnmente utilizado como paso previo en procesos de detección de bordes o segmentación de imágenes.

3. **Filtro gaussiano:** Empleando la distribución normal para aplicar un desenfoque controlado en la imagen, brinda una opción más sofisticada que el suavizado. La desviación estándar  $\sigma$  se incluye en la fórmula bidimensional y controla el grado del suavizado. A mayor valor de  $\sigma$ , mayor es la pérdida de nitidez. Tal y como se representa en la Figura 8, este filtro es eficaz para reducir progresivamente las variaciones de intensidad, manteniendo una apariencia natural y evitando la aparición de artefactos visibles. Se utiliza comúnmente como una fase inicial en los algoritmos de detección de características.

**Figura 8.**

*Imagen de hoja de café con filtro gaussiano*



**Nota.** Ilustración tomada de Vives Garnique et al. (2014), donde se presenta una imagen de hoja de café después de aplicar un filtro gaussiano con:  $\sigma = 0,6$  en (a) y  $\sigma = 1,0$  en (b).

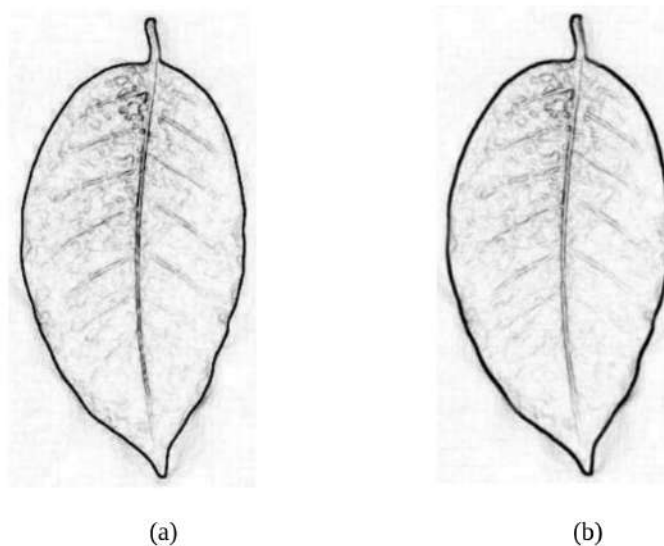
4. **Filtros realzantes:** A diferencia de los filtros suavizantes, los filtros realzantes se cen-

tran en destacar detalles finos de la imagen, en especial los bordes. Su función principal es resaltar transiciones abruptas en la intensidad de píxeles, lo cual es fundamental para tareas de reconocimiento de patrones o segmentación precisa. Dentro de esta categoría, se destacan tres enfoques principales: el filtro de Sobel, el filtro Laplaciano y el Laplaciano de Gaussiano (LoG).

5. **Filtro de Sobel:** Se fundamenta en el análisis del gradiente de la imagen, que examina las variaciones de la intensidad en términos verticales y horizontales a través de operadores diferenciales. La ubicación de los bordes y su dirección se pueden determinar gracias a la magnitud y dirección del gradiente. Su puesta en marcha es sólida y eficaz, lo que resulta particularmente valioso para identificar bordes en imágenes que han sido suavizadas antes. Según se observa en la Figura 9, el resultado mejora considerablemente si se combina con filtros anteriores, como el gaussiano o de mediana; esto mejora el rendimiento general del proceso.

**Figura 9.**

*Imagen de hoja de café filtrada*



*Nota.* Ilustración tomada de Vives Garnique et al. (2014), donde se muestra una imagen de hoja de café filtrada previamente con: (a) filtro de mediana y (b) filtro gaussiano con  $\sigma = 1,0$  en (b).

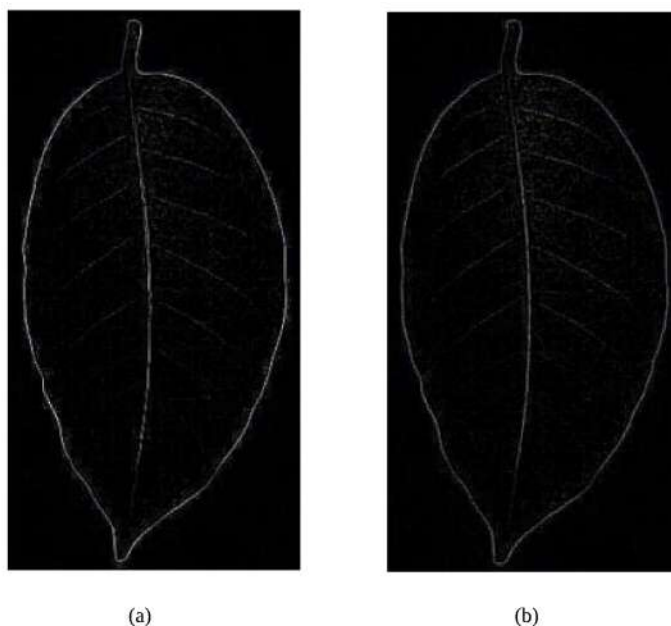
6. **Filtro Laplaciano:** El operador Laplaciano representa una derivada de segundo orden que responde a cambios rápidos en la intensidad de la imagen. Aunque es potente pa-

ra detectar bordes, presenta ciertas limitaciones como la alta sensibilidad al ruido y la generación de bordes dobles. Por esta razón, rara vez se utiliza de forma aislada. Sin embargo, cuando se combina con filtros de suavizado, puede ofrecer una detección de bordes más precisa y estable.

7. **Filtro Laplaciano de Gaussiano (LoG):** El filtro LoG surge de la combinación del suavizado gaussiano con el operador Laplaciano, lo cual permite reducir el ruido antes de aplicar la detección de bordes. Como se muestra en la Figura 10, su formulación matemática permite calcular la máscara de convolución utilizada en esta operación, lo que resulta en un detector de bordes más robusto y preciso.

**Figura 10.**

*Imagen de hoja de café filtrada (LoG)*



*Nota.* Ilustración tomada de Vives Garnique et al. (2014), donde se muestra una imagen de hoja de café procesada mediante dos tipos de filtrado: (a) filtro de mediana y (b) filtro gaussiano.

### 2.3.2 Iluminación

La iluminación es uno de los elementos más determinantes en los sistemas de visión artificial, ya que afecta directamente la calidad de adquisición de imágenes y la precisión en los

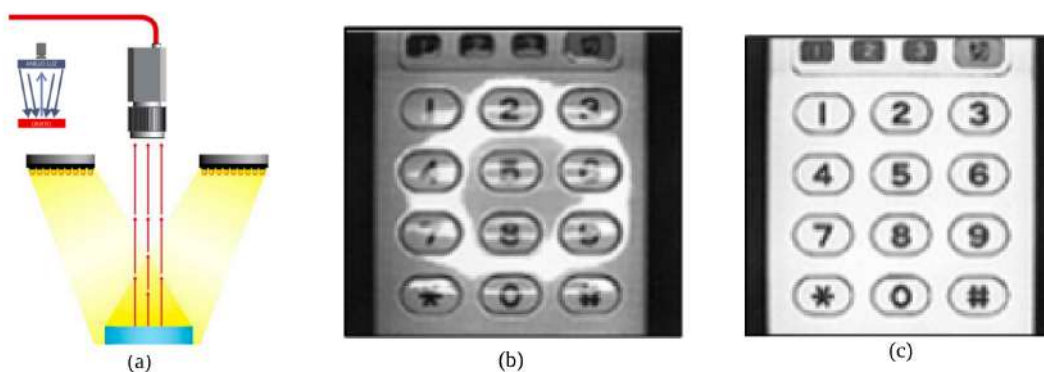
procesos de análisis posteriores. Como enfatizan Noguer y Antiga (2025), la elección adecuada del tipo de iluminación facilita que se destaquen detalles importantes, se reduzcan los errores de interpretación y se ajuste la inspección a diversas clases de superficies y materiales. En el campo de la visión artificial, seleccionar la iluminación apropiada es esencial para asegurar la correcta adquisición y análisis de imágenes.

Tal como lo explican Noguer y Antiga (2025), existen diversas configuraciones de iluminación que se adaptan a necesidades específicas, como la identificación de contornos, la evaluación de texturas, o la detección de imperfecciones y marcas sobre distintas superficies. A continuación, se presentan las principales técnicas de iluminación utilizadas en sistemas de visión artificial.

1. **Iluminación Frontal:** Este tipo de iluminación consiste en posicionar la cámara y la fuente de luz en la misma dirección hacia el objeto. Esta configuración permite suavizar las texturas y minimizar los efectos de rayas, polvo e imperfecciones, ya que la luz reflejada regresa directamente a la cámara. Como se observa en la Figura 11, una de las principales ventajas de esta técnica es la eliminación de sombras, así como la capacidad de aplicarla a grandes distancias. No obstante, su uso también conlleva el inconveniente de generar reflejos intensos en superficies altamente reflectantes.

**Figura 11.**

*Iluminación frontal y sus aplicaciones*

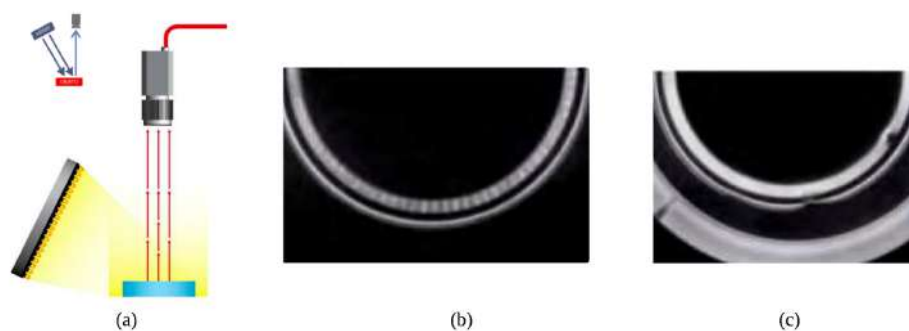


**Nota.** Imagen adaptada de Noguer y Antiga (2025), que resalta el impacto de la iluminación frontal en superficies con escasa reflexión, al comparar la imagen sin dicha iluminación (b) con la imagen resultante al aplicarla (c).

2. **Iluminación lateral:** Para que la cámara vea el objeto de frente, la fuente de luz debe estar colocada perpendicularmente a este. Al crear sombras bien definidas, incluso en los detalles más pequeños, esta orientación ayuda a resaltar bordes, fisuras y relieves. La inclinación del rayo de luz se adapta dependiendo del nivel de contraste que se quiera. Se emplea cuando se requiere la identificación de anomalías específicas en la superficie. Como se ilustra en la Figura 12, si el ángulo de incidencia de la luz respecto a la horizontal es muy bajo, puede ocurrir que la luz produzca sombras extensas, lo cual complica un examen exacto de toda la pieza.

**Figura 12.**

*Inspección de latas con luz lateral*

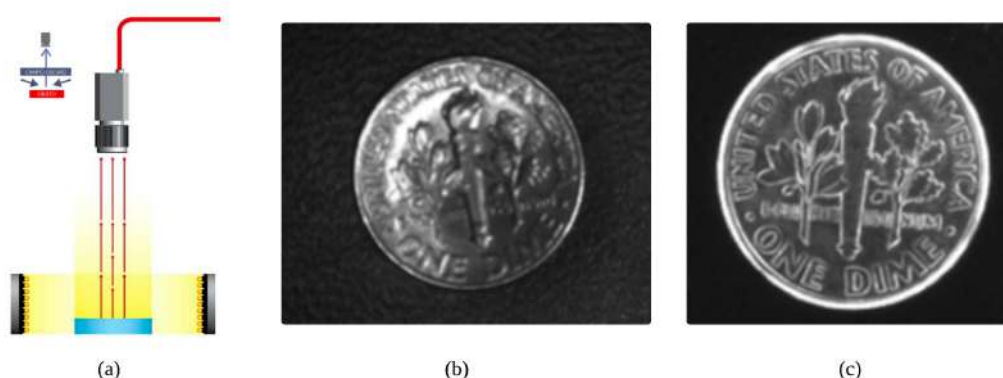


**Nota.** Representación gráfica adaptada de Noguer y Antiga (2025) que muestra el procedimiento de inspeccionar latas a través de la iluminación lateral: (a) lata sin defectos y (b) lata con defectos, en la cual la luz lateral facilita la identificación de imperfecciones potenciales y la acentuación de los relieves.

3. **Iluminación por Campo Oscuro (Dark Field):** En este método, la luz se proyecta lateralmente desde un ángulo extremadamente bajo mediante un anillo que distribuye la iluminación en todas las direcciones. La luz rebota en defectos superficiales e incide en la cámara, destacando detalles que normalmente pasarían desapercibidos. Es muy utilizada para detectar grabados, códigos o marcas poco visibles sobre superficies metálicas o de bajo contraste, como metal sobre metal o gris sobre gris. Como se observa en la Figura 13, aunque esta técnica resalta detalles finos en condiciones de bajo contraste, su efectividad disminuye considerablemente en superficies que absorben la luz.

**Figura 13.**

*Inspección con campo oscuro (Dark Field)*

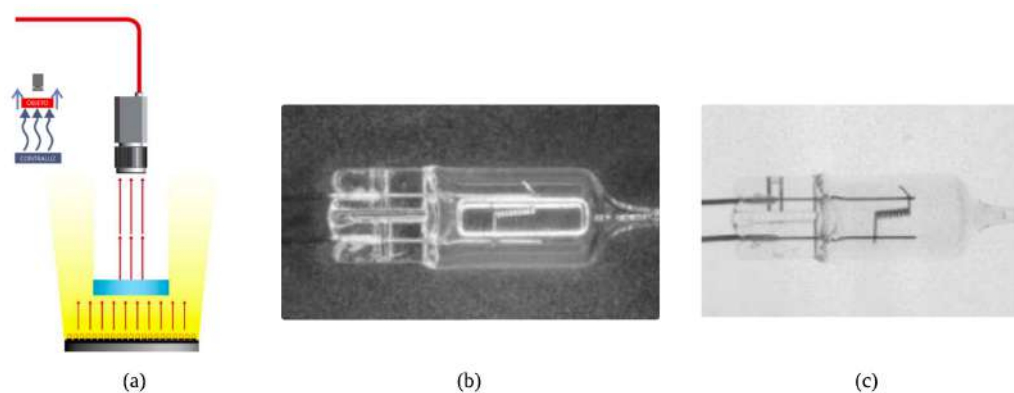


**Nota.** Ilustración adaptada de Noguer y Antiga (2025) que muestra la inspección con iluminación de campo oscuro: (a) sin darkfield y (b) con darkfield, donde este tipo de luz permite resaltar detalles en superficies con bajo contraste.

4. **Iluminación por Contraste (Backlight):** Este tipo de iluminación se basa en colocar la fuente luminosa detrás del objeto, de forma que este quede entre la luz y la cámara. La cámara capta la silueta del objeto por contraste, permitiendo realizar mediciones altamente precisas, eliminando completamente las sombras. Como se muestra en la Figura 14, aunque esta técnica es eficaz para identificar contornos e impurezas, no permite observar detalles superficiales, como inscripciones o códigos.

**Figura 14.**

*Inspección de filamentos de bombillas con contraluz*



**Nota.** La figura 14 es una adaptación de Noguer y Antiga (2025) y muestra el análisis de filamentos de bombillas con contraluz: (a) sin iluminación posterior y (b) con contraluz, que hace que los detalles de la silueta se destaquen nítidamente.

5. **Iluminación Axial Difusa:** La iluminación axial difusa se obtiene mediante un haz de luz lateral que es desviado  $90^\circ$  por un espejo semitransparente. De esta forma, la luz incide en la superficie del objeto en la misma dirección del eje óptico de la cámara, generando una iluminación homogénea y difusa. Esta técnica es adecuada para superficies planas reflectantes, como placas electrónicas (PCBs), etiquetas brillantes o zonas de impresión en materiales metálicos. Como se observa en la Figura 15, su principal ventaja radica en la reducción de reflejos especulares no deseados. No obstante, no permite identificar los relieves presentes en el objeto.

**Figura 15.**

*Inspección de cojinetes con iluminación axial difusa*

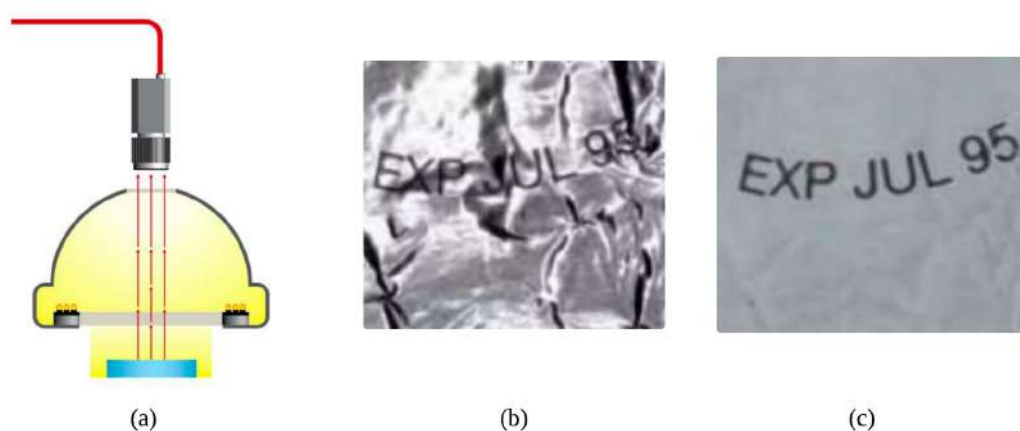


**Nota.** Se muestra la revisión de cojinetes con luz axial difusa en esta ilustración adaptada de Noguer y Antiga (2025): (a) sin iluminación y (b) con iluminación. La luz lateral destaca los detalles sutiles de la superficie.

6. **Iluminación Difusa Tipo Domo:** Esta modalidad de iluminación utiliza una cúpula esférica que dispersa la luz desde todas direcciones sobre el objeto, lo que permite eliminar sombras, suavizar texturas y minimizar efectos provocados por rayas, polvo o curvaturas. También se conoce como iluminación de día nublado, ya que proporciona una luz suave y uniforme que no genera sombras marcadas. Como se presenta en la Figura 16, esta técnica resulta especialmente útil para la inspección de superficies altamente reflectantes o curvas, como el instrumental médico, espejos o latas. Sin embargo, su principal desventaja es el elevado coste asociado a su implementación.

**Figura 16.**

*Inspección con iluminación difusa tipo domo*



**Nota.** Ilustración adaptada de Noguera y Antiga (2025) que ilustra el procedimiento de inspección por medio de iluminación difusa tipo domo: (a) sin iluminación y (b) con iluminación, demostrando cómo la luz difusa incrementa notablemente la visibilidad del texto grabado.

### 2.3.3 Etapas en el proceso de visión artificial

La visión artificial, como apunta González Marcos (2006), está conformada por una serie de procesos interconectados que incluyen elementos tanto de software como de hardware, apoyados en una base teórica firme proveniente de la inteligencia artificial, el procesamiento digital de imágenes y la óptica. Su objetivo principal es posibilitar que las máquinas tengan la capacidad de observar e interpretar su entorno a través de la adquisición, análisis y entendimiento de datos visuales.

Estos procesos abarcan desde la adquisición de imágenes, donde se obtienen los datos visuales mediante sensores ópticos, cámaras RGB o cámaras multiespectrales, hasta el preprocesamiento, que mejora la calidad de la imagen y reduce el ruido. Posteriormente, el procesamiento y análisis permiten la extracción de características relevantes como bordes, texturas o colores que servirán para la interpretación final, donde el sistema toma decisiones o genera respuestas automatizadas.

En la Tabla 4 se presentan las etapas del flujo de trabajo de la visión artificial, cada una con una función esencial para asegurar el cumplimiento de los objetivos del sistema.

**Tabla 4.**

*Pasos en el procesamiento de imágenes*

<b>Paso</b>	<b>Descripción</b>
<b>1. Adquisición de la imagen</b>	El primer paso es adquirir la imagen digital utilizando sensores que digitalizan la señal producida por el sensor.
<b>2. Preprocesamiento</b>	Una vez obtenida la imagen digital, se realiza el preprocesamiento con el objetivo de mejorarla, aumentando las probabilidades de éxito en el proceso.
<b>3. Segmentación</b>	La segmentación consiste en dividir la imagen en las partes que la constituyen. Es uno de los problemas más difíciles, ya que una segmentación incorrecta puede llevar al fracaso del proceso.
<b>4. Representación de datos</b>	Después de la segmentación, se determina si se empleará una representación por región completa o por frontera. La representación por frontera se utiliza para rasgos externos, en tanto que la representación por región se centra en las propiedades internas.
<b>5. Selección o parametrización de rasgos</b>	Esta etapa extrae características que producen información cuantitativa relevante o que son esenciales para distinguir entre clases de objetos.
<b>6. Reconocimiento e interpretación</b>	El reconocimiento consiste en otorgar una etiqueta a los objetos basándose en los descriptores que se extraen, mientras que la interpretación confiere un significado a los objetos que se reconocen.

*Nota.* La tabla presenta las principales etapas del procesamiento de imágenes en sistemas de visión artificial, desde la adquisición y preprocesamiento hasta el reconocimiento e interpretación de los objetos. Adaptado de González Marcos (2006).

### 2.3.4 Etapas del procesamiento de imágenes

El procesamiento de imágenes permite transformar la información visual en datos relevantes para su análisis en diferentes campos de aplicación. Seguir de forma organizada cada etapa del proceso resulta esencial para asegurar una mejor calidad de las imágenes y facilitar su interpretación posterior. Cada fase cumple un rol específico, por lo que respetar la secuencia lógica favorece la extracción de características importantes y permite lograr resultados más precisos y confiables. Según Rubén Wainschenker (2011), una estructura adecuada en el procesamiento de imágenes contribuye de manera significativa a mejorar la eficiencia de los sistemas de reconocimiento y clasificación, lo que impacta directamente en el éxito de los proyectos que dependen del análisis visual.

Wainschenker (2011) describe los pasos fundamentales del procesamiento de imágenes a continuación, explicando cómo cada etapa ayuda a asegurar un análisis preciso y eficaz de la información visual.

1. **Captura:** En esta fase inicial se determina de manera meticulosa la forma en que se capturará la imagen. Esto supone elegir el tipo de cámara o sensor apropiado en función de lo que requiera el proyecto (por ejemplo, cámaras térmicas, RGB, hiperespectrales); establecer parámetros como la apertura, la resolución (megapíxeles), la velocidad de obturación y el ISO; así como determinar la distancia focal. Para garantizar que la imagen contenga la mayor cantidad de datos útiles para el análisis posterior, también se deben tener en cuenta elementos como el ángulo de captura, las condiciones de iluminación y la lejanía del objeto de interés. Una adecuada captura asegura que los datos sean de mejor calidad y hace más sencillos los pasos posteriores.
2. **Pre-procesamiento:** En esta fase se llevan a cabo acciones con el propósito de optimizar la calidad de las imágenes para un análisis más eficaz. El preprocesamiento puede abarcar la normalización del color, el ajuste de brillo y contraste, la corrección de distorsiones ópticas y la eliminación de ruido (por medio de filtros gaussianos o medianos).

Para hacer que las características importantes sean más visibles, también es posible utilizar métodos de ecualización de histograma. El propósito es disminuir o eliminar aquellos componentes que no son significativos (por ejemplo, artefactos de captura o fondos complejos) para enfocarse en las características o áreas que sí lo son.

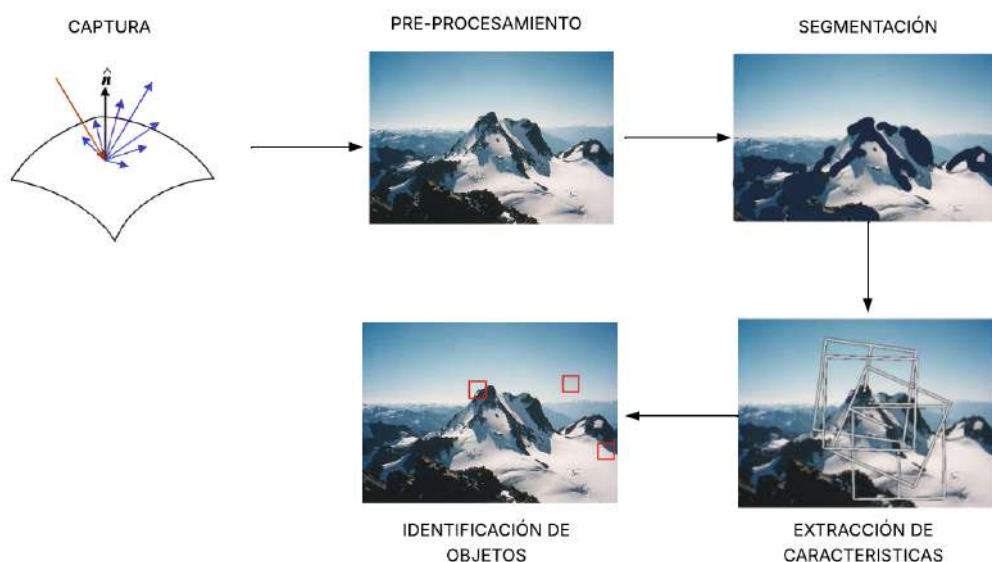
3. **Segmentación:** La segmentación consiste en dividir la imagen en regiones significativas, separando los objetos de interés del resto de la escena. Esta etapa es crucial porque permite trabajar directamente con las partes relevantes de la imagen. Se pueden utilizar algoritmos de clustering (como el k-means), segmentación por bordes, técnicas más sofisticadas que se basan en redes neuronales, o métodos de umbralización (adaptativa o simple). El resultado de la segmentación es una o varias máscaras que destacan las áreas correspondientes a cada objeto o característica específica.
4. **Extracción de características:** Aquí se identifican y cuantifican atributos específicos de las regiones segmentadas para representar de manera más compacta y significativa a los objetos. Estas características pueden ser simples, como el área, perímetro, color promedio, o más complejas, como descriptores de textura (GLCM, LBP), contornos, momentos invariantes o características extraídas por redes profundas (deep features). La calidad y pertinencia de las características extraídas son fundamentales para el éxito de la clasificación posterior, ya que representan la "firma" del objeto dentro del sistema.
5. **Identificación de objetos:** En esta etapa se utiliza un modelo de clasificación para asignar cada conjunto de características extraídas a una categoría específica. Para esto, se emplean clasificadores que pueden ser algoritmos tradicionales (como SVM, k-NN, árboles de decisión) o modelos más complejos basados en aprendizaje profundo (como CNNs). El modelo debe ser previamente entrenado con ejemplos representativos para que pueda generalizar correctamente a nuevos datos. La toma de decisión puede basarse en medidas de similitud, reglas estadísticas o redes de neuronas, y permite determinar finalmente qué objeto está presente en la imagen y a qué clase pertenece.

Una vez descritas las etapas del procesamiento de imágenes, se observa cómo la informa-

ción obtenida por los sensores atraviesa diversas fases, que van desde la mejora y segmentación de la imagen hasta la extracción de características y la interpretación final. Cada una de estas etapas cumple un papel esencial dentro del sistema de visión artificial, ya que la precisión de los resultados depende directamente de la calidad del procesamiento previo. Este flujo permite transformar los datos visuales en información significativa, facilitando tareas como la detección, el reconocimiento y el análisis automatizado. La Figura 17 presenta la aplicación de dicho proceso en el análisis de imágenes.

**Figura 17.**

*Etapas en el procesamiento de imágenes*



*Nota.* Ilustración adaptada de Rubén Wainschenker (2011), que detalla de manera secuencial las fases esenciales del procesamiento de imágenes, desde la obtención hasta la interpretación final.

### 2.3.5 Componentes de un Sistema de Visión Artificial

Cada uno de estos elementos cumple una función específica que contribuye a transformar la información visual capturada por los sensores en datos interpretables por un computador. Este flujo permite ejecutar tareas complejas como la detección de objetos, el reconocimiento de patrones, la medición de características o la automatización de procesos en entornos in-

dustriales y agrícolas. En conjunto, estos componentes constituyen la base tecnológica que posibilita el funcionamiento eficiente de los sistemas de visión artificial, garantizando precisión, rapidez y confiabilidad en la toma de decisiones automatizadas. La Tabla 5 presenta los componentes principales de un sistema de visión artificial, los cuales intervienen de manera coordinada en el proceso de adquisición, procesamiento y análisis de imágenes.

**Tabla 5.**

*Componentes en el procesamiento de imágenes*

<b>Componente</b>	<b>Descripción</b>
<b>Sensor óptico</b>	El sensor óptico, que puede ser una cámara de color o monocromática, captura imágenes completas del entorno. La cámara puede actualizar la imagen cada 1/30 segundos. También se utilizan escáneres que generan una línea de imagen a la vez, creando una imagen bidimensional. La elección del tipo de sensor depende de la aplicación específica.
<b>Tarjeta de adquisición de imagen</b>	La tarjeta de adquisición de imagen se encarga de digitalizar la señal de video que recibe del sensor óptico, convirtiéndola en una imagen que puede ser procesada por un computador.
<b>Computador</b>	Una vez digitalizada la imagen, esta debe almacenarse en la memoria del computador para ser procesada y manipulada mediante programas especializados en visión artificial.
<b>Monitor de vídeo</b>	El monitor de vídeo permite visualizar tanto las imágenes capturadas como los resultados del procesamiento.

*Nota.* Los elementos más relevantes del sistema para el procesamiento de imágenes se muestran en la tabla.

Adaptado según González Marcos (2006).

### **2.3.6 Algoritmos básicos para el reconocimiento de patrones**

En visión artificial, el reconocimiento y clasificación de imágenes permite identificar patrones y asignar etiquetas a objetos en una imagen. En el caso de las hojas de rosas, su aplicación

es clave para detectar la presencia y el tipo de plaga que afecta al cultivo.

Para ello, se emplean diferentes algoritmos de reconocimiento, entre los que destacan los métodos basados en características, que detectan elementos clave en la imagen, y los enfoques de aprendizaje automático, que aprenden patrones a partir de datos de entrenamiento, logrando mayor precisión en la identificación (James Le, 2018).

### 2.3.6.1 Etapas de reconocimiento en los algoritmos de reconocimiento de patrones

Las etapas de reconocimiento en los algoritmos de reconocimiento de patrones son fundamentales para que un sistema sea capaz de identificar patrones o características relevantes en datos complejos. Estas etapas permiten que el algoritmo procese información de manera estructurada, de modo que pueda reconocer patrones significativos, clasificarlos y actuar en consecuencia. La Tabla 6 presenta las diferentes etapas del proceso de reconocimiento de patrones, que incluyen las siguientes fases clave:

**Tabla 6.**

*Etapas en el reconocimiento de patrones*

<b>Etapas</b>	<b>Descripción</b>
<b>Reconocimiento exploratorio de patrones</b>	Es la primera etapa en los algoritmos de detección de patrones. Utiliza el aprendizaje automático para examinar los datos sin categorías predefinidas, agrupando los elementos similares en una clasificación no supervisada. Es como clasificar calcetines por colores sin lista previa.
<b>Reconocimiento descriptivo de patrones</b>	Después de descubrir patrones ocultos, esta etapa los clasifica en categorías predefinidas. Utiliza clasificación supervisada, donde el algoritmo se entrena con datos etiquetados. Se aplica en tareas como la detección de spam, el reconocimiento de imágenes o la transcripción de audio.

*Nota.* Se describen las etapas que conforman el reconocimiento de patrones, desde el análisis exploratorio de datos hasta la clasificación supervisada de la información. Adaptado de Bansal (2025).

### 2.3.6.2 Técnicas más comunes de reconocimiento de patrones

El reconocimiento de patrones es una parte fundamental de la visión artificial, ya que permite identificar y clasificar datos según diferentes criterios. Existen diversas técnicas para lo-

garlo, las cuales se resumen en la Tabla 7. Entre las más destacadas se encuentran:

**Tabla 7.**

*Técnicas más comunes de reconocimiento de patrones*

<b>Método</b>	<b>Descripción</b>	<b>Aplicaciones</b>
<b>Reconocimiento estadístico</b>	Analiza regularidades en los datos mediante algoritmos estadísticos.	Detección de spam, análisis de frecuencia de palabras.
<b>Reconocimiento sintáctico</b>	Identifica patrones según su estructura geométrica.	Escritura a mano, secuencias biológicas.
<b>Redes neuronales</b>	Procesan datos no estructurados mediante entrenamiento.	Visión por computadora, reconocimiento del habla.
<b>Aprendizaje automático</b>	Algoritmos que mejoran la precisión del reconocimiento.	Recomendaciones comerciales, vehículos autónomos.

*Nota.* Se sintetizan las principales técnicas empleadas en el reconocimiento de patrones, junto con sus características y aplicaciones en distintos ámbitos tecnológicos. Adaptado de Bansal (2025).

## 2.4 Redes Neuronales

Las redes neuronales son modelos informáticos que se fundamentan en el funcionamiento del cerebro humano y sirven para identificar patrones, aprender de los datos, hacer predicciones y clasificaciones. Matich (2024) indica que, a pesar de que existen varias maneras de definirlas, se pueden sintetizar en las siguientes:

- **Nueva forma de computación:** Las redes neuronales son un paradigma informático que toma como inspiración los procesos biológicos cerebrales. En ellas, las unidades individuales colaboran para ejecutar tareas complejas.
- **Modelo matemático organizado en niveles:** Se componen de numerosos elementos procesales (neuronas) dispuestos en capas, donde cada capa realiza una operación matemática para transformar los datos de entrada en salidas útiles, lo que permite la realización de tareas como clasificación, regresión o reconocimiento de patrones.
- **Sistema de computación interconectado:** Están formadas por elementos simples (neuronas) que están altamente interconectados, procesando información en función de su estado dinámico, lo que les permite aprender de las entradas y adaptarse a diferentes situaciones.

- **Redes neuronales artificiales:** Son redes de unidades simples y adaptativas organizadas jerárquicamente, diseñadas para imitar el sistema nervioso biológico, con el objetivo de interactuar y aprender del mundo real, permitiendo tareas como el reconocimiento de objetos, el procesamiento de lenguaje natural o la predicción de datos.

#### 2.4.0.1 Ventajas que ofrecen las red neuronal.

Las redes neuronales ofrecen diversas ventajas en el campo de la inteligencia artificial, especialmente cuando se trata de procesamiento de datos complejos, patrones no lineales y tareas de aprendizaje automático. A continuación en la Tabla 8 se detallan algunas de las principales ventajas que ofrecen:

**Tabla 8.**

*Ventajas de las redes neuronales artificiales*

Ventajas	Descripción
<b>Aprendizaje adaptativo</b>	Las redes neuronales pueden aprender a partir de la experiencia o el entrenamiento, mejorando su rendimiento en tareas específicas sin ser programadas explícitamente para cada caso.
<b>Auto-organización</b>	Durante el proceso de aprendizaje, la red neuronal organiza y representa la información recibida por sí misma, optimizando su capacidad para tareas como clasificación y predicción.
<b>Tolerancia a fallos</b>	A pesar de los daños en partes de su estructura, las redes neuronales pueden seguir funcionando, lo que les otorga una notable resistencia a fallos parciales.
<b>Operación en tiempo real</b>	Gracias a la capacidad de realizar cálculos en paralelo, las redes neuronales operan rápidamente, ideales para aplicaciones como el reconocimiento de imágenes y procesamiento de señales.
<b>Fácil inserción en tecnología existente</b>	Los chips especializados permiten integrar las redes neuronales en sistemas y dispositivos ya existentes, mejorando la eficiencia en tareas específicas.

*Nota.* Se destacan las principales ventajas de las redes neuronales artificiales, entre ellas su capacidad de aprendizaje, adaptación y funcionamiento eficiente en entornos complejos. Adaptado de Amazon Web Services (2024).

#### 2.4.0.2 Elementos básicos de una red neuronal

Las redes neuronales artificiales están formadas por neuronas interconectadas y distribuidas en capas organizadas en niveles. Los datos ingresan a través de la capa de entrada, pasan por las capas ocultas y finalmente se generan las salidas en la capa de salida. Es importante

destacar que las capas ocultas pueden estar compuestas por varias subcapas, lo que aumenta la complejidad y capacidad de la red.

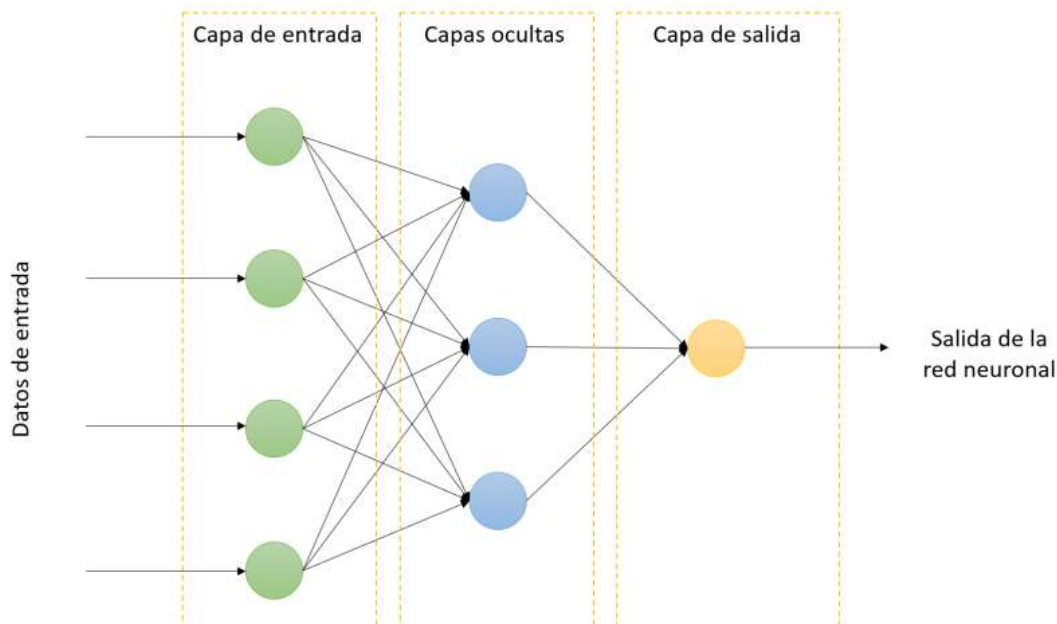
Para comprender cómo funcionan las redes neuronales, es fundamental entender cómo operan las neuronas artificiales. Estas se inspiran en las neuronas biológicas y tienen ciertas similitudes: ambas reciben entradas, utilizan pesos para procesar la información y generan una salida.

Las capas de una red neuronal se pueden clasificar de la siguiente manera:

- **Capa de entrada:** Es la que obtiene la información externa que la red procesará. En esta capa no se lleva a cabo ningún procesamiento; simplemente se reciben los datos.
- **Capas ocultas:** Son las capas internas, que son el lugar donde se lleva a cabo el procesamiento de los datos. Estas capas pueden ser muchas, según lo compleja que sea la red; además, no tienen contacto directo con el medio exterior. La topología de la red se define por el número de capas ocultas y cómo están conectadas entre sí.
- **Capa de salida:** Tiene la responsabilidad de transmitir hacia el exterior los datos procesados, produciendo así el resultado final. En problemas de clasificación, es posible tener una neurona por clase; en los problemas de regresión, solo se necesita una sola neurona que ofrezca el valor de salida.

Una red neuronal totalmente conectada es aquella en la que cada neurona de una capa está conectada con todas las neuronas de la capa siguiente, lo que permite una comunicación eficiente entre los diferentes niveles de la red.

La Figura 18 representa la estructura básica de una red neuronal, conformada por tres capas: una capa de entrada que recibe los datos, una o más capas ocultas que procesan la información y una capa de salida que entrega el resultado final.

**Figura 18.***Estructura de una red neuronal*

*Nota.* El gráfico muestra la arquitectura de una red neuronal (Qamar y Zardari, 2023).

### 2.4.0.3 Tipos de neuronas artificiales.

Según UNIR Mexico (2024), las neuronas artificiales se dividen en dos tipos según los valores que pueden tomar: neurona binaria, que solo puede tener valores en 0, 1 o -1, 1, y neurona real, que puede tomar valores en  $[0, 1]$  o  $[-1, 1]$ . Los pesos de las conexiones generalmente no tienen restricciones, pero en algunos casos específicos pueden estar limitados a ciertos rangos.

En la Tabla 9 se muestra una clasificación pormenorizada de las redes neuronales, que se basa en su estructura. Esta clasificación facilita la identificación y diferenciación de diversos tipos de redes según cómo están organizadas las neuronas en sus capas y cómo se conectan entre sí, lo cual afecta directamente su desempeño y el potencial que tienen para procesar información.

**Tabla 9.***Tipos de redes neuronales y métodos de aprendizaje*

<b>Tipo</b>	<b>Estructura</b>	<b>Método de aprendizaje</b>
<b>Monocapa</b>	Una sola capa de neuronas que recibe valores externos, los procesa y genera valores de salida. Las neuronas no se interconectan entre sí.	<b>Supervisado:</b> El aprendizaje se realiza con secuencias de entrenamiento donde se conoce el resultado deseado, y se ajustan los pesos para minimizar el error.
<b>Multicapa</b>	Contiene una capa de entrada, varias capas ocultas y una capa de salida. Las neuronas de cada capa se conectan con todas las de la capa siguiente, pero no entre ellas.	<b>Corrección de error:</b> Se utilizan algoritmos como el mínimo error cuadrado (LMS) o back-propagation.
<b>Convolutacional</b>	Similar a la multicapa, pero las neuronas de cada capa solo se conectan con un subconjunto de las neuronas de la siguiente capa, reduciendo la complejidad del sistema.	<b>Estocástico:</b> Los cambios en los pesos se hacen de forma aleatoria hasta obtener los mejores resultados.
<b>Concurrentes</b>	No están organizadas en capas, sino que las neuronas están interconectadas de manera no estructurada, permitiendo que la información anterior influya en el procesamiento futuro.	<b>No supervisado:</b> La red ajusta los pesos buscando patrones o relaciones en los datos sin conocer el resultado de antemano (por ejemplo, técnicas de clustering).

*Nota.* Se describen los principales tipos de redes neuronales junto con sus estructuras y métodos de aprendizaje, destacando sus diferencias en la organización de capas y en la forma en que ajustan los pesos para optimizar el rendimiento. Adaptado de (Espinoza, 2017).

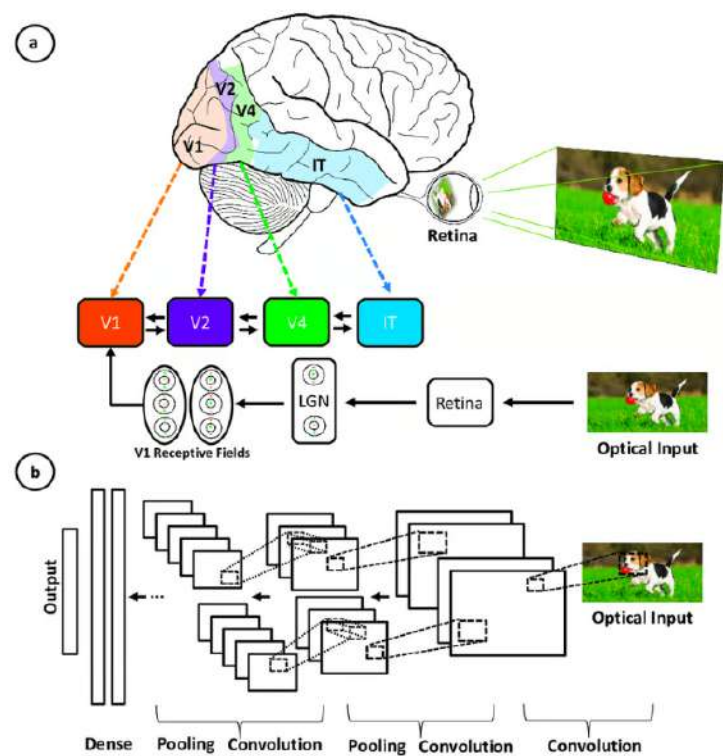
### 2.4.1 Redes neuronales convolucionales

Las redes neuronales convolucionales (CNN) han transformado la visión artificial al posibilitar que las máquinas analicen imágenes con mayor precisión y eficiencia. Como señala Bajaña Calle y Pin Guerrero (2024), estas redes sobresalen en labores como la detección de rostros, el reconocimiento de objetos y la categorización de imágenes. Su estructura está compuesta por numerosas capas interconectadas que analizan la información visual a diferentes niveles, lo cual posibilita una mejor comprensión del contenido.

La Figura 19 muestra las diferencias y similitudes entre la arquitectura en capas de la corteza visual humana y las capas de una red neuronal convolucional (CNN). La imagen muestra que la corteza visual del ser humano procesa la información visual en varias fases, cada una de las cuales está especializada en identificar elementos concretos de la imagen, como formas, colores y bordes. Asimismo, en una CNN, las capas de agrupamiento, convolución y completamente conectadas se ocupan de extraer, reducir y clasificar las propiedades de las imágenes jerárquicamente, simulando el procesamiento visual del cerebro humano.

**Figura 19.**

*Arquitectura en capas de la corteza visual humana y las capas de una red neuronal convolucional*



**Nota.** Ilustración de la correspondencia entre las áreas asociadas a la corteza visual primaria y las capas de una red neuronal convolucional (Bajaña Calle y Pin Guerrero, 2024).

En diversos sectores se puede observar el impacto de estas tecnologías. En la industria alimentaria, por ejemplo, han posibilitado la optimización de la clasificación de productos, lo que ha llevado a una reducción en los tiempos y a un mejoramiento en la automatización de los procedimientos (Aguilar Campoverde, 2019). Asimismo, su implementación se expande a campos como la conducción autónoma, la seguridad, el reconocimiento de objetos en tiempo real y la medicina. Esto permite que se desarrollen sistemas inteligentes que cambian la manera en que interactúan la tecnología y el mundo real.

La Tabla 10 muestra un resumen de los avances recientes en las arquitecturas de redes neuronales, subrayando innovaciones importantes en el aprendizaje profundo y su efecto en diferentes industrias. Estos avances han mejorado la precisión en el procesamiento de imágenes, el reconocimiento de patrones y el análisis predictivo. Esto ha facilitado su uso en áreas como la

conducción autónoma, la salud y la seguridad informática.

**Tabla 10.**

Avances en arquitecturas de redes neuronales y sus aplicaciones

Sector	Aplicación	Beneficios
<b>Conducción autónoma</b>	Implementación de redes neuronales convolucionales (CNN) en los sistemas de Tesla.	Mejora en un 20 % la toma de decisiones, optimiza el reconocimiento de objetos y la capacidad de respuesta en tiempo real.
<b>Salud</b>	Uso de redes neuronales recurrentes (RNN) en IBM Watson Health para analizar datos médicos.	Reducción del tiempo de diagnóstico en un 30 %, detección temprana de enfermedades crónicas y mejora en la precisión del tratamiento.
<b>Análisis predictivo</b>	Optimización de arquitecturas de redes neuronales en diversos sectores.	Mejora en la habilidad de prever tendencias en infraestructura, salud y finanzas, lo que hace más fácil tomar decisiones estratégicas.

*Nota.* Se presentan ejemplos de aplicación de diferentes arquitecturas de redes neuronales en sectores como la conducción autónoma, la salud y el análisis predictivo, resaltando sus beneficios en precisión, eficiencia y toma de decisiones. Adaptado de Aguilar y Campoverde (2019).

#### 2.4.1.1 Estructura de redes neuronales convolucionales (CNN)

Según Purwono et al. (2023), las CNN son notables por su estructura única, que incluye capas de convolución, capas de agrupamiento y capas completamente conectadas. En este tipo de redes, las neuronas se disponen en diferentes capas. Las capas totalmente conectadas hacen un procesamiento avanzado al unir cada nodo de la capa anterior con cada nodo de la siguiente. Normalmente, estas capas utilizan una función de activación softmax para dar a cada entrada una probabilidad, lo que permite clasificar la información de forma precisa entre 0 y 1.

Bansal (2025) describe los componentes principales de las CNN de la siguiente manera:

- **Capas de convolución:** Estas son el corazón de las CNN y se encargan de extraer características de la imagen de entrada. Utilizan pequeños filtros (por ejemplo, matrices 3x3 o 5x5) que recorren la imagen para detectar detalles como bordes, texturas o patrones específicos. Cada filtro genera un mapa de características que resalta una particularidad de la imagen. Además, después de aplicar el filtro se suele emplear una función de activación no lineal, para capturar relaciones complejas en los datos.

- **Capas de agrupación (pooling):** Consiste en disminuir la dimensión de los mapas de características que se consiguen en las capas de convolución, lo cual reduce tanto el número de parámetros como el peligro de sobreajuste. Esto se consigue al conservar únicamente la información más relevante. El average-pooling y el max-pooling, que escoge el máximo de cada área, son dos de los métodos más frecuentes; sin embargo, este último método es normalmente menos agresivo en cuanto a la disminución de detalles.
- **Capas completamente conectadas:** Situadas normalmente al final de la red, estas capas funcionan como clasificadores. Establecen una conexión entre cada neurona de la capa anterior y todas las neuronas de la siguiente, fusionando los rasgos extraídos para generar una representación total de la imagen. En los casos de clasificación, es frecuente que estas capas hagan uso de la función softmax, que transforma los valores en probabilidades (que oscilan entre 0 y 1) para establecer con qué clases se relaciona cada uno. En el transcurso del entrenamiento, los pesos de estas conexiones se modifican con el objetivo de reducir al mínimo el error de predicción del modelo.

#### 2.4.2 Formación y aprendizaje de una red neuronal convencional

El aprendizaje automático se utiliza para mejorar el rendimiento de la generalización y combatir el sobreaprendizaje en las redes neuronales convolucionales. Es un proceso iterativo que ajusta los pesos de la red para minimizar una función de pérdida que describe la desviación entre las predicciones del modelo y los valores reales de los datos de entrenamiento.

A continuación Purwono et al. (2023) menciona el proceso de la formación y aprendizaje de una red convencional:

- **Propagación Directa (Forward Pass):** En esta fase, la imagen de entrada se procesa a través de todas las capas de la red (convolución, agrupamiento y totalmente conectadas), generando una salida o predicción final. Cada capa transforma los datos recibidos, pasando la información de una a la siguiente hasta producir una estimación basada en los

parámetros actuales del modelo.

- **Cálculo de la pérdida:** Cuando se recibe la predicción, se calcula el error comparándola con el valor real a través de una función de pérdida apropiada, como la entropía cruzada para trabajos de clasificación o el error cuadrático medio para regresión. Este cálculo establece la diferencia entre la salida del modelo y la realidad del terreno, brindando así una medición exacta del desempeño de dicho modelo en ese momento.
- **Retropropagación:** En esta fase, el error calculado se propaga desde la capa de salida hacia la de entrada, lo que posibilita calcular los gradientes de la función de pérdida en relación con cada uno de los pesos. Para establecer cómo cada peso afecta al error, se emplean derivadas parciales, lo que permite modificar el modelo de forma informada.
- **Actualización de pesos:** Se utilizan métodos de optimización, como el gradiente descendente, con los gradientes ya calculados, para regular los pesos de la red en sentido contrario al gradiente. El tamaño de cada paso de ajuste está determinado por la tasa de aprendizaje, lo cual garantiza que la red avance progresivamente hacia un mínimo de la función de pérdida sin sobrepasar el óptimo.
- **Iteración por épocas:** Para cada ejemplar del conjunto de entrenamiento y durante varias épocas, se repite el proceso total, que incluye desde la propagación directa hasta la actualización de pesos. Con cada repetición, la red aumenta su habilidad para identificar patrones en los datos y disminuye el error global, lo que optimiza su capacidad de generalizarse a datos nuevos.

### 2.4.3 Tipos de redes neuronales convolucionales

A lo largo del tiempo, las redes neuronales han experimentado una notable evolución, lo que ha llevado al desarrollo de diversas arquitecturas diseñadas para abordar problemas específicos. Según Alessandro Giusti (2024), estas arquitecturas incluyen:

- **LeNet:** Fue una de las primeras CNN desarrolladas y se diseñó específicamente para

reconocer dígitos escritos a mano. Su estructura sentó las bases para muchas redes posteriores.

- **VGGNet:** Se distingue por su estructura sencilla pero profunda, lo que la vuelve efectiva para trabajos de clasificación de imágenes.
- **GoogLeNet (Inception):** Presenta una arquitectura que es más eficaz que las anteriores, ya que posibilita el reconocimiento de diversas características de forma simultánea y mejora la utilización de parámetros.
- **DenseNet:** Su método se fundamenta en la reutilización de conexiones entre capas, lo que permite crear modelos más compactos y con menos parámetros sin sacrificar el rendimiento.

## 2.5 Los drones, una herramienta clave en la agricultura de precisión

Los drones han llegado a ser una herramienta esencial en la agricultura de precisión, especialmente en el monitoreo de las parcelas agrícolas. Tal como se menciona en Pino-Vargas (2019), a través de la captura de imágenes aéreas, los drones generan datos exactos que, al ser procesados de manera adecuada, ofrecen información crucial sobre diferentes aspectos del cultivo, tales como:

- Estrés hídrico de las plantas.
- Deficiencias nutricionales en los cultivos.
- Incidencia de plagas, enfermedades y malas hierbas.
- Estado de desarrollo y fenología de las plantas.

Una de las principales ventajas de los drones es la facilidad con la que los datos obtenidos pueden transferirse a un pendrive, permitiendo su posterior carga en el ordenador del tractor.

Como se detalla en la Tabla 11, se destacan los beneficios de la implementación de drones en este contexto.

**Tabla 11.**

Ventajas de los drones agrícolas

<b>Ventaja</b>	<b>Descripción</b>	<b>Impacto</b>
Optimización de insumos	Los drones permiten aplicar agroquímicos de manera precisa, ajustándose a las necesidades específicas de cada área del campo.	Contribuye a reducir el uso innecesario de insumos, mejorando la eficiencia en la aplicación.
Ahorro de tiempo y dinero	La precisión en la aplicación de productos agroquímicos reduce el desperdicio, lo que se traduce en un ahorro significativo de tiempo y costos.	Optimiza los costos operativos y acelera las tareas de monitoreo y tratamiento.
Protección del medio ambiente	Al optimizar el uso de agroquímicos, los drones contribuyen a una menor contaminación del suelo y el agua.	Favorece la sostenibilidad y reduce el impacto ambiental de la agricultura.
Integración con otras fuentes de datos	Los drones pueden conectarse con otras fuentes de datos para proporcionar información adicional que apoye la toma de decisiones agrícolas.	Mejora la rentabilidad y la toma de decisiones estratégicas para la explotación agrícola.

**Nota.** Se resumen las principales ventajas que ofrecen los drones en el ámbito agrícola, destacando su papel en la optimización de recursos, la reducción de costos y el impulso hacia una producción más sostenible. Adaptado de Pino-Vargas (2019).

### 2.5.1 Tipos y Clases de Vehículos No Tripulados

Los vehículos no tripulados, conocidos comúnmente como drones, pueden ser aéreos, terrestres o acuáticos, y representan una de las innovaciones tecnológicas más relevantes de las últimas décadas. Estos dispositivos han experimentado una notable evolución tanto en diseño como en capacidad operativa, lo que ha permitido su incorporación en múltiples sectores, como la seguridad, la cartografía, la investigación científica y, de manera destacada, la agricultura de precisión.

Su clasificación puede abordarse desde distintas perspectivas, incluyendo el uso (recreativo, comercial o científico), el tipo de control (autónomo o remoto) y la forma física o estructura (multirrotor, de ala fija o híbrido). La Tabla 12 presenta los principales tipos y clases de vehículos no tripulados, destacando sus características y aplicaciones más comunes.

**Tabla 12.**  
Clasificación de Drones

Clasificación	Descripción	Ejemplos
Según su uso	<b>Drones militares:</b> Diseñados para operaciones de seguridad y defensa, pueden estar armados o ser utilizados para espionaje.	Misiones de ataque o reconocimiento.
	<b>Drones civiles:</b> Se utilizan en diversas áreas.	<b>Drones comerciales:</b> Fotogrametría, producción multimedia, mensajería. <b>Drones para aficionados:</b> Uso recreativo y de entretenimiento.
	<b>Drones gubernamentales:</b> Empleados por organismos estatales.	Actividades como seguridad, rescate y extinción de incendios.
Según el tipo de control	<b>Autónomos:</b> No requieren intervención humana, guían por sensores y sistemas programados.	Operaciones totalmente automáticas.
	<b>Monitorizados:</b> Supervisados por un operador sin intervención directa en el control.	Fotogrametría y agricultura de precisión.
	<b>Supervisados:</b> Requieren intervención ocasional del operador.	Operación en entornos complejos.
	<b>Preprogramados:</b> Siguen un plan de vuelo previamente establecido.	Inspecciones y monitoreo programado.
	<b>Controlados remotamente:</b> Son pilotados en tiempo real por un operador.	Juegos, vigilancia y control a distancia.
Según su forma	<b>Multirrotores:</b> Alta estabilidad, pero limitado por consumo energético.	Tricópteros, cuadricópteros, hexacópteros, octocópteros.
	<b>Helicópteros:</b> Gran capacidad de carga y autonomía, pero complejos de manejar.	Fotogrametría, vigilancia, agricultura de precisión.
	<b>Ala fija:</b> Mayor autonomía y eficiencia aerodinámica.	Fotogrametría, agricultura de precisión (grandes terrenos).

*Nota.* Se muestran las principales clasificaciones de drones . Adaptado de Stehr (2015.)

En la Figura 20 se muestran diseños de drones utilizados en la agricultura: el multirrotor (cuadricóptero), ideal para áreas pequeñas y monitoreo detallado, y el dron de ala fija, adecuado para cubrir grandes extensiones de terreno de manera eficiente.

**Figura 20.**

*Tipos y clases de vehículos no tripulados*



**Nota.** Ilustraciones de drones utilizados en la agricultura: (a) multirrotor (cuadricóptero), (b) de ala fija Pino-Vargas (Pino-Vargas, 2019).

### 2.5.2 Mapeo y topografía de cultivos

Como menciona Stehr (2015), una de las principales ventajas de utilizar la tecnología de drones agrícolas es la facilidad con la que se puede realizar un monitoreo a gran escala de los cultivos. Gracias a esta tecnología, es posible obtener imágenes en tiempo real y animaciones que reflejan la evolución de los cultivos. Estos datos son altamente fiables y proporcionan una excelente visión general del estado de los cultivos en la parcela. Con esta tecnología, se pueden determinar aspectos clave como:

- El estado fitosanitario de las plantas y del cultivo en general.
- La distribución de la tierra según el cultivo.
- La fenología (etapa del ciclo de desarrollo) del cultivo en tiempo real.
- Mapas georreferenciados detallados del área de cultivo.

## 2.6 Metodología del diseño

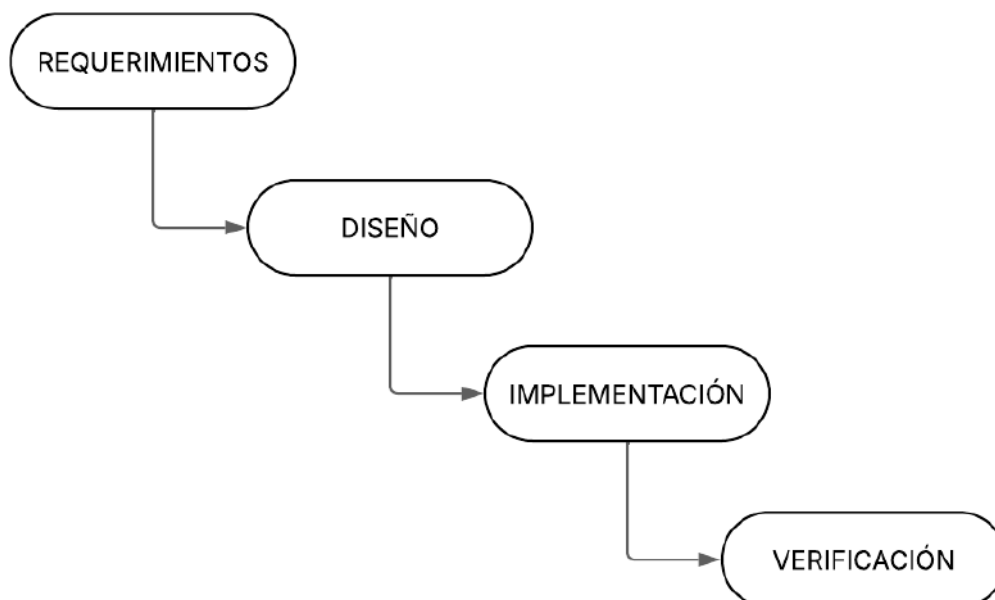
En el desarrollo de proyectos tecnológicos es fundamental contar con una metodología que permita un control adecuado del proceso. El modelo de desarrollo en cascada destaca por su enfoque secuencial y lineal, el cual facilita la organización del proyecto.

### 2.6.1 Modelo de desarrollo en cascada

El modelo de desarrollo en cascada, también conocido como metodología lineal, se fundamenta en un enfoque estructurado en el que el proceso de desarrollo se organiza en fases consecutivas, las cuales deben completarse y validarse antes de avanzar a la siguiente, lo que permite un control progresivo del proyecto Digital Talent Agency (2018). En la Figura 21 se ilustran las fases de la metodología en cascada.

**Figura 21.**

*Modelo en cascada*



**Nota.** Ilustración propia que representa el Modelo en Cascada, mostrando de manera secuencial las fases del desarrollo del prototipo.

De acuerdo con Digital Talent Agency (2018), el modelo en cascada se compone de una serie de fases que se desarrollan en el siguiente orden:

1. **Análisis de requerimientos:** Fase en la que se identifican y documentan las necesidades del proyecto, considerando los objetivos, restricciones y expectativas de las partes involucradas.
2. **Diseño:** Fase en la que se define la estructura general de la solución, estableciendo componentes, herramientas y estrategias necesarias para su desarrollo.
3. **Implementación:** Corresponde a la construcción del sistema conforme a lo establecido en la fase de diseño, incluyendo la identificación y corrección de errores.
4. **Pruebas y verificación:** Fase destinada a comprobar que el sistema cumple con los requerimientos definidos, asegurando su correcto funcionamiento.

## **CAPÍTULO III**

### **MATERIALES Y MÉTODOS**

En el presente capítulo se presenta el diseño del sistema orientado a la detección de roya (*Phragmidium mucronatum*) en cultivos de rosas, para lo cual se emplea la metodología del Modelo en Cascada. Esta metodología permite estructurar el desarrollo del proyecto en fases secuenciales claramente definidas, facilitando un control adecuado de cada una de las etapas del proceso. La aplicación del Modelo en Cascada contribuye a una planificación ordenada como lo es la identificación de los requerimientos del sistema. De igual manera, favorece al procesamiento de las imágenes capturadas y el modelo de visión artificial utilizado para la detección de la enfermedad.

#### **3.1 Metodología del Modelo en Cascada**

Para el desarrollo del sistema se adoptó el Modelo en Cascada, seleccionado por su enfoque secuencial y estructurado, el cual se ajusta a los requerimientos del diseño e implementación del prototipo de detección de roya mediante el uso de un dron.

Este modelo permitió una planificación progresiva, abarcando desde la definición de los requerimientos del sistema y la selección de los componentes tecnológicos, hasta el procesamiento de imágenes, el entrenamiento del modelo de visión artificial y la visualización de los resultados obtenidos. La Figura 22 ilustra las fases que conforman este proceso.

**Figura 22.**

*Metodología en cascada aplicada al sistema*



**Nota.** Metodología del Modelo en Cascada aplicada al desarrollo del prototipo. Elaborado por el autor.

En la fase de requerimientos, se definieron las necesidades del sistema, considerando aspectos técnicos y operativos. Se determinó que se utilizaría un dron para el monitoreo aéreo, incorporando un modelo de visión por computadora para la detección de *Phragmidium mucronatum*. Se establecieron los parámetros necesarios para el monitoreo eficiente de los cultivos de rosales, priorizando una operación autónoma y segura del dron durante los vuelos.

En la fase de diseño, se estructuró la arquitectura del sistema, definiendo los módulos principales como el vuelo autónomo del dron, la captura de imágenes, el procesamiento con técnicas de visión por computadora y el análisis visual. Se establecieron los algoritmos de segmentación de imágenes para detectar síntomas de roya.

Durante la fase de implementación, se llevara a cabo la integración de los diferentes componentes del sistema, así como la programación de algoritmos para el procesamiento y análisis

de las imágenes capturadas.

Finalmente, en la fase de verificación, se realizarán pruebas en campo en cultivos reales de rosales. Se evaluará la precisión del sistema en comparación con observaciones manuales, determinando el grado de acierto en la detección de la roya. También se verificará la estabilidad del dron durante el vuelo, la nitidez de las imágenes capturadas y la eficacia del procesamiento automático.

### **3.2 Situación Actual**

Para entender la situación en la que se implementará el sistema de detección de roya en plantaciones de rosas, se debe realizar un análisis muy detallado del estado actual. Es por ello que se definen los criterios importantes para el desarrollo del prototipo, se recogió la información necesaria durante este proceso.

Se adquirió la información a través de encuestas dirigidas específicamente a trabajadores y productores responsables de gestionar los rosales, con el objetivo de abordar directamente la actividad agrícola. Mediante sus respuestas, logramos determinar las condiciones en las que se lleva a cabo el monitoreo fitosanitario, los retos habituales de la detección de enfermedades y las necesidades verdaderas del campo.

La roya (*Phragmidium mucronatum*) pertenece a una infección fúngica que se propaga con facilidad cuando el clima se encuentra en las condiciones apropiadas. Esta enfermedad amenaza la salud de las plantas de rosa y reduce la producción. Gill et al. (2021) menciona que esta enfermedad es provocada por los hongos *Phragmidium mucronatum* y tiene mayor presencia en zonas con mucha humedad. En estos casos es fundamental monitorear constantemente el cultivo para detectar con tiempo los indicios de infección y actuar.

Asimismo, Gill et al. (2021) señala que en el crecimiento, examinar con regularidad los tallos y las hojas contribuye a identificar más pronto plagas y enfermedades.

En el marco de la transformación digital en la agricultura, por otro lado, Herrera et al. (2024) sugieren un sistema novedoso que combina visión por computadora y vehículos aéreos no tripulados (UAVs) para evaluar la condición de los rosales.

La Tabla 13 muestra los pormenores del método de investigación que se utilizó para conseguir la información esencial acerca de las percepciones de los productores y trabajadores que se ocupan de gestionar los rosales, así como sobre las prácticas agrícolas actuales.

**Tabla 13.**

Método de Investigación y Encuesta

Aspecto	Descripción	Impacto
Método de Investigación	Investigación descriptiva orientada a observar, analizar y describir las prácticas actuales del cultivo de rosas frente a la plaga de roya.	Permite conocer el contexto real para el desarrollo de un sistema basado en visión artificial y drones.
Instrumento de recolección	Encuesta estructurada con preguntas de tipo cerrado.	Facilita la cuantificación, tabulación y posterior análisis estadístico de los datos.
Temas tratados en la encuesta	1. Prácticas de cultivo actuales. 2. Métodos de control y prevención de la roya. 3. Viabilidad y aceptación del uso de drones para detección temprana.	Apoya el diseño de un prototipo eficaz y adaptado a las necesidades del entorno agrícola.
Justificación del tipo de preguntas	Las preguntas cerradas permiten obtener indicadores precisos y comparables.	Agiliza la interpretación de resultados y facilita su representación visual.
Formato de aplicación	Se aplicó en formato físico y digital a productores y trabajadores del sector agrícola.	Amplía el alcance y mejora la representatividad de la muestra.
Análisis de resultados	Se emplearán herramientas estadísticas para identificar patrones y conclusiones relevantes.	Garantiza un respaldo cuantitativo sólido para validar el desarrollo del sistema.

**Nota.** Método de investigación y estructura de la encuesta aplicada para recopilar y analizar datos sobre prácticas agrícolas y uso de drones en la detección de roya. Elaboración propia.

### 3.2.1 Análisis de los resultados

La encuesta realizada para los 52 trabajadores agrícolas dentro del sector de Tabacundo, en la provincia de Pichincha permitió evidenciar que el 52 % de los participantes manifestó conocer la roya como enfermedad que afecta a los cultivos de rosas, mientras que el 48 % no la identifica con claridad. No obstante, el 77 % considera “muy importante” su detección oportuna para evitar daños en el cultivo, y un 94 % reconoce que el uso de drones puede mejorar la rapidez y precisión del monitoreo frente a los métodos tradicionales.

En cuanto al diseño del prototipo, se notó que el 63 % de los encuestados indicaron que es muy importante que el dron sea pequeño y fuerte; por otro lado, el 83 % pensaron que la facilidad de manejo es fundamental, incluso para aquellos sin formación técnica. Además, el 67 % considera que almacenar los resultados en una base de datos es ventajoso, mientras que el 96 % destaca la importancia de contar con un registro de imágenes y sugerencias automáticas para abordar la roya. Estos porcentajes respaldan la idea de un sistema de detección que emplea drones y visión por computadora, el cual tiene como finalidad ayudar en el proceso de tomar decisiones para cultivar rosas.

### 3.2.2 Propósito del prototipo

El propósito de este proyecto es desarrollar un prototipo que, mediante un dron dotado con una cámara de alta resolución, posibilite la detección eficiente de la roya (*Phragmidium mucronatum*) en cultivos de rosas. El propósito de este sistema es potenciar la eficiencia del monitoreo de los cultivos. Esto se consigue a través de la identificación rápida de los síntomas de roya, lo cual es esencial para evitar que se propague por mayor tiempo. La meta de este prototipo es proporcionar a los agricultores una herramienta innovadora que optimice la supervisión de la roya, disminuya el uso de productos químicos al mínimo y, en consecuencia, aumente tanto la calidad como el volumen de rosas producidas.

De igual manera, el sistema propuesto permitirá reducir el tiempo dedicado a las inspec-

ciones tradicionales y brindar información que ayude a tomar decisiones eficientes y en el momento adecuado para salvaguardar las cosechas. La ejecución de este modelo facilitará que los agricultores empleen tecnologías más sofisticadas, lo cual les posibilitará administrar sus recursos de manera más eficiente y asegurar la rentabilidad y sostenibilidad de las cosechas.

### 3.2.3 Dimensionamiento de Stakeholders

En la fase inicial de creación del prototipo, se lleva a cabo la identificación de los stakeholders, que son los participantes principales que influyen o tienen alguna participación en el diseño, implementación y funcionamiento del sistema para detectar la roya (*Phragmidium mucronatum*) en plantaciones de rosales por medio de un dron.

Este análisis incluye a los stakeholders tanto internos como externos. El equipo de desarrollo del prototipo es el primero. En cuanto a los stakeholders externos, se encuentran principalmente los cultivadores de rosas, que son los más beneficiados con el sistema, y también técnicos agrícolas cuya experiencia es esencial para validar y aplicar el prototipo en la práctica.

La Tabla 14 proporciona información detallada acerca de los actores directos e indirectos que están participando en el proyecto, incluyendo la clase de relación que tienen con el sistema desarrollado, así como su nivel de influencia y grado de participación.

**Tabla 14.**

*Lista de Stakeholders*

<b>Categoría</b>	<b>Nombres</b>
<b>Usuarios directos</b>	UTN
<b>Usuarios indirectos</b>	Sr. Fredy Castillo
<b>Administradores</b>	Sra. Dayana Cualchi
<b>Director</b>	Msc. Carlos Vásquez, Msc. Henry Farinango
<b>Desarrollador</b>	Sr. Farid Avila

*Nota.* Identificación de los actores clave del proyecto según su nivel de participación y responsabilidad. Elaboración propia.

### 3.3 Requerimientos

La norma IEEE 29148, que ofrece pautas para transformar los requerimientos de los interesados en requisitos nítidos y bien definidos, se ha tomado como base durante el desarrollo de los requerimientos del prototipo. El IEEE 29148, además, abarca todos los procedimientos vinculados con la ingeniería de requisitos, asegurando que el desarrollo esté en concordancia con las metas fijadas a lo largo de toda su evolución ISDI (2024).

#### 3.3.1 Nomenclatura de los requerimientos

Para facilitar la lectura y organización de la información se ha creado un conjunto de abreviaturas que permiten representar los distintos tipos de requerimientos del sistema de forma más eficaz. Estas abreviaturas ayudan a identificar rápidamente si un requerimiento es funcional o no funcional para el desarrollo del prototipo. La Tabla 15 presenta estas abreviaturas junto con su significado, lo cual permite una mejor comprensión del contenido a lo largo del documento.

**Tabla 15.**

Abreviatura de los requerimientos

Abreviatura	Requerimiento
StSR	Requerimiento de Stakeholders
SySR	Requerimientos del Sistema
SRSR	Requerimientos de Arquitectura

*Nota.* Abreviaturas empleadas para los distintos niveles de requerimientos del sistema. Elaboración propia.

#### 3.3.2 Requerimientos de Stakeholders

Los requisitos operacionales establecen los estándares de rendimiento y técnicos que el sistema debe cumplir para el sector agrícola, incluyendo aspectos como su integración con plataformas para monitorear a distancia, su implementación en invernaderos y la extensión del área cultivable. Los requerimientos del usuario, por otro lado se enfocan en la posibilidad de ac-

ceder a la información en tiempo real, así como en la facilidad de uso y la accesibilidad. Esto garantiza que tanto los técnicos como los agricultores puedan tomar decisiones exactas cuando surge una enfermedad. Estos componentes, en su totalidad, garantizan una solución completa y eficaz para identificar la roya en las plantaciones de rosas, tal como se puede observar en la Tabla 16.

**Tabla 16.**  
Requerimientos Operacionales y de Usuarios

#	Requerimientos	Prioridad		
		Alta	Media	Baja
<b>Requerimientos operacionales</b>				
STSR1	El sistema de detección debe ser implementado dentro de un invernadero.		X	
STSR2	El dron debe ser compacto y resistente para ambientes de exterior.			X
STSR3	El sistema debe disponer de una base de datos estructurada para almacenar y gestionar las imágenes capturadas de las rosas durante el proceso de detección.		X	
STSR4	El plan de vuelo debe cubrir toda el área del cultivo.	X		
STSR5	El sistema debe ser fácil de integrar en el entorno de cultivo.		X	
STSR6	El sistema debe asegurar la integridad y respaldo de los datos recolectados	X		
<b>Requerimientos de usuario</b>				
STSR7	El sistema debe ser fácil de usar por el agricultor o técnico.	X		
STSR8	Visualización de los resultados obtenidos durante la detección de imágenes de las rosas y la identificación de roya.		X	
STSR9	Reporte del estado del cultivo que pueda visualizar el agricultor.		X	
STSR10	El sistema debe ofrecer recomendaciones sobre el tratamiento de las plantas.		X	
STSR11	El sistema debe permitir el acceso a la información de los cultivos en cualquier momento.		X	
STSR12	La plataforma debe ser accesible desde dispositivos móviles.			X

*Nota.* Tabla de requerimientos operacionales y del usuario. Elaborado por el autor.

### 3.3.3 Requerimientos del sistema

Como se indica en la Tabla 17 se presentan los requerimientos del sistema necesarios para la ejecución adecuada del proceso de monitoreo y detección de roya en cultivos de rosas. Estos requerimientos incluyen condiciones de uso, interfaces, estados de funcionamiento y aspectos físicos que deben cumplirse para garantizar el correcto desempeño del sistema en campo.

**Tabla 17.**

Requerimientos del Sistema (SySR)

#	Requerimientos del sistema (SySR)	Prioridad		
		Alta	Media	Baja
<b>Requerimientos de uso</b>				
SySR1	El dispositivo debe estar encendido para llevar a cabo el proceso de detección de roya.	X		
SySR2	El algoritmo de detección debe ejecutarse al iniciar el plan de vuelo en el cultivo.			X
SySR3	El sistema guarda automáticamente los resultados de la detección en la base de datos.		X	
SySR4	El sistema debe contar con una interfaz gráfica para iniciar la detección.		X	
SySR5	El usuario podrá reiniciar el proceso de detección desde la aplicación web.			X
<b>Requerimientos de Performance</b>				
SySR6	El sistema debe soportar la detección simultánea.		X	
SySR7	El tiempo de respuesta del sistema de detección debe ser menor a 10 segundos.		X	
SySR8	El dron debe tener un punto de inicio de fácil acceso.			X
SySR9	El sistema debe garantizar una velocidad de procesamiento estable.	X		
SySR10	El sistema debe ser capaz de funcionar sin interrupciones durante al menos 8 horas.		X	
<b>Requerimientos de interfaces</b>				
SySR11	La tarjeta de almacenamiento debe tener una capacidad mínima de 8 GB.		X	
SySR12	El sistema debe ser compatible con múltiples resoluciones de cámaras.		X	
SySR13	La base de datos debe recopilar y almacenar los datos en un formato adecuado.		X	
SySR14	El sistema debe ser compatible con protocolos de comunicación inalámbrica.		X	
SySR15	La interfaz del dron debe mostrar el estado de conexión en tiempo real.		X	
<b>Requerimientos de estados</b>				
SySR16	La placa de procesamiento debe estar encendido para iniciar el sistema.	X		
SySR17	La cámara del dron debe capturar imágenes con la resolución y calidad necesarias para identificar signos de roya en el cultivo.	X		
SySR18	El sistema es capaz de indicar cuando la detección haya finalizado		X	
SySR19	El sistema debe indicar el estado del procesamiento en cada fase de detección.		X	
SySR20	El sistema debe indicar un estado de fallo cuando ocurra una interrupción crítica.		X	
<b>Requerimientos físicos</b>				
SySR21	El dispositivo debe ser portable para su uso en distintas zonas del cultivo.			X
SySR22	Se debe contar con un área de cultivo accesible para realizar la detección.		X	
SySR23	El dron debe operar en condiciones adecuadas por su susceptibilidad al agua.		X	
SySR24	El dispositivo debe tener protección contra golpes.			X
SySR25	El dron debe permitir el reemplazo rápido de la batería.			X

**Nota.** Requerimientos establecidos para la arquitectura del sistema. Elaborado por el autor.

### 3.3.4 Requerimientos de la arquitectura

Como se muestra en la Tabla 18 en la que se detallan los requerimientos de arquitectura del sistema, los cuales se dividen en categorías lógicas, de hardware, software, eléctricos y de diseño. Cada grupo establece las condiciones necesarias para garantizar el correcto funcionamiento del sistema de monitoreo, considerando factores como compatibilidad, rendimiento, eficiencia energética y facilidad de integración.

**Tabla 18.**

Requerimientos de Arquitectura (SRSH)

#	Requerimientos de Arquitectura (SRSH)	Prioridad		
		Alta	Media	Baja
<b>Requerimientos lógicos</b>				
SRSH1	El sistema de monitoreo debe ser compatible con diferentes sistemas operativos .		X	
SRSH2	El sistema debe ser capaz de procesar las imágenes y videos		X	
SRSH3	El sistema debe procesar los datos sin necesidad de conexión directa con el dron.		X	
SRSH4	Los datos debe ser subidos de forma manual a la aplicación web.	X		
SRSH5	El sistema debe permitir la exportación de datos en PDF para análisis posterior.		X	
<b>Requerimientos de hardware</b>				
SRSH6	El dron debe tener una cámara con alta resolución para la captura de imágenes.		X	
SRSH7	El sistema debe ser capaz de manejar y procesar grandes volúmenes de datos.		X	
SRSH8	El dron debe tener soporte GPS.	X		
SRSH9	Se debe incluir una batería eficiente para el dron.		X	
SRSH10	El dispositivo debe contar con una tarjeta gráfica dedicada NVIDIA compatible con el sistema de procesamiento y visión artificial.		X	
<b>Requerimientos de software</b>				
SRSH11	El sistema debe ser compatible con sistemas operativos basados en GNU/Linux.		X	
SRSH12	El software debe ser capaz de procesar imágenes capturadas por el dron.		X	
SRSH13	Uso eficiente de la memoria RAM para evitar sobrecargar el sistema.	X		
SRSH14	Capacidad de generar reportes para el tratamiento del cultivo.		X	
SRSH15	El software del sistema debe permitir actualizaciones seguras			X
<b>Requerimientos eléctricos</b>				
SRSH16	El sistema debe tener una fuente de energía para alimentar el sistema de detección.		X	
SRSH17	El sistema debe tener una fuente de energía de respaldo.			X
SRSH18	El consumo energético del sistema debe ser eficiente.		X	
SRSH19	El sistema debe contar con un sistema de protección contra sobrecargas eléctricas.		X	
SRSH20	El sistema de monitoreo debe tener un consumo energético bajo.			X
<b>Requerimientos de Diseño</b>				
SRSH21	La estructura del dron debe ser compacto y ligero, con un peso no superior a 2 kg.		X	
SRSH22	El dron debe ser resistente al agua y al polvo para operar en condiciones agrícolas.			X
SRSH23	El dron debe contar con una estructura robusta que soporte impactos y caídas menores.		X	
SRSH24	El sistema debe permitir un fácil acceso a la interfaz principal.	X		
SRSH25	El sistema de monitoreo debe permitir visualizar la presencia de Roya.	X		

*Nota.* Tabla de requerimientos establecidos para la arquitectura del sistema. Elaborado por el autor.

### 3.4 Selección de Hardware y Software

La elección apropiada del hardware y el software es un elemento esencial en la creación del sistema de detección de roya en plantaciones de rosales por medio de visión artificial. Esta etapa se ocupa de garantizar que todos los elementos físicos y lógicos sean capaces de satisfacer las demandas operativas y funcionales definidas en el diseño del prototipo.

#### 3.4.1 Hardware de desarrollo

Optar por el hardware para el sistema de detección de roya es un paso importante, pues establece la base en la que se llevarán a cabo las funciones esenciales del prototipo. Como el sistema está diseñado para capturar, procesar y detectar imágenes, se necesita una arquitectura que sea compatible con dispositivos periféricos de visión artificial y que consuma poca energía.

##### 3.4.1.1 Selección de la placa de desarrollo

La selección de la placa de desarrollo se realizó de acuerdo con el objetivo del proyecto, orientado al procesamiento de imágenes para la detección de roya. La Tabla 19 muestra el análisis comparativo de las placas evaluadas.

**Tabla 19.**

Selección de la placa de desarrollo

Componente	SRSH1	SRSH2	SRSH7	SRSH10	SRSH13	Valor total
Dell G15 - Ryzen 7 (PC)	2	2	2	2	2	10
Lenovo Ideapad Slim 3 - Core i5 (PC)	2	2	2	0	2	8
LattePanda Delta 3	2	2	1	0	1	6
Jetson Nano	2	2	1	2	1	8
BeagleBone Black	1	1	1	0	1	4
<b>Cumple totalmente:</b>	2					
<b>Cumple parcialmente:</b>	1					
<b>No cumple:</b>	0					

*Nota.* Tabla comparativa para la selección de la placa de desarrollo para el prototipo. Elaboración propia.

**Elección:** Se determinó que el modelo Dell G15 con procesador Ryzen 7 es la alternativa más apropiada para desarrollar el prototipo, debido a su capacidad de cómputo elevada y su desempeño óptimo en trabajos de visión artificial. La capacidad de procesamiento y la compatibilidad con GPU NVIDIA hacen posible llevar a cabo una detección más precisa de roya, tal como se explica en el Anexo 6.

### 3.4.1.2 Selección del dron

Como se muestra en la Tabla 20, se estudiaron varias opciones disponibles en el mercado para elegir el dron más apropiado para el prototipo de detección de roya. La evaluación tuvo en cuenta elementos como la estabilidad al volar, la eficiencia energética y la calidad de las capturas de imágenes, con el fin de garantizar un desempeño óptimo durante las operaciones aéreas de supervisión en el cultivo.

**Tabla 20.**

Selección del Dron Profesional

<b>Dron Profesional</b>	<b>SRSH6</b>	<b>SRSH8</b>	<b>SRSH9</b>	<b>SRSH21</b>	<b>SRSH23</b>	<b>Valor total</b>
DJI Air 3	2	2	2	2	2	10
DJI Mavic 2 Enterprise	2	2	1	1	2	8
Parrot Anafi	1	2	1	2	1	7
Autel EVO II Pro	2	2	2	1	1	8
SenseFly eBee X	2	2	2	1	2	9
<b>Cumple totalmente:</b>	2					
<b>Cumple parcialmente:</b>	1					
<b>No cumple:</b>	0					

*Nota.* Comparativa técnica entre drones profesionales. Elaborado por el autor.

**Elección:** El dron DJI Air 3 fue el elegido para el sistema de detección de roya, ya que su cámara de alta resolución, su autonomía de vuelo y su diseño liviano permiten una integración ideal con el sistema de visión artificial. Esto garantiza que las imágenes se tomen y examinen con exactitud mientras el cultivo es supervisado, como se indica en el Anexo 7.

### 3.4.2 Software

La selección del software se centró en buscar herramientas que puedan satisfacer las necesidades funcionales del sistema, en particular lo concerniente a procesar imágenes de manera efectiva, ser compatibles con sistemas operativos embebidos y combinarse con bibliotecas de visión artificial. En este escenario, fue preciso examinar numerosas opciones que pudieran adecuarse al entorno hardware establecido, maximizando tanto el rendimiento como la utilización de los recursos existentes.

### 3.4.3 Sistema Operativo

Para seleccionar el sistema operativo más adecuado, se realizó una comparación técnica entre diversas plataformas, basándose en los requerimientos específicos definidos en la arquitectura del sistema. La Tabla 21 presenta aspectos clave como la compatibilidad con herramientas de visión artificial, el rendimiento en entornos con recursos limitados, la capacidad de integración con otros componentes del sistema, y la estabilidad del sistema operativo para ejecutar algoritmos de reconocimiento de imágenes.

**Tabla 21.**

Selección del Sistema Operativo

Sistema Operativo	SySR9	SySR10	SySR12	SRS13	SRS15	Valor total
Ubuntu	2	2	2	2	2	10
Debian	2	2	2	2	1	9
CentOS	2	2	1	2	1	8
Fedora	2	1	1	2	1	7
Windows 10	1	1	1	1	1	5
<b>Cumple totalmente:</b>	2					
<b>Cumple parcialmente:</b>	1					
<b>No cumple:</b>	0					

*Nota.* Comparativa de sistemas operativos según eficiencia y compatibilidad con el sistema. Elaborado por el autor.

**Elección:** Después de contrastar diferentes alternativas, se optó por Ubuntu como sistema operativo para la creación del sistema de detección de roya. Debido a su entorno estable, su capacidad de gestión de recursos y la gran cantidad de bibliotecas de visión artificial a dispo-

sición lo que hacen posible la implementación de soluciones con un alto rendimiento. Así, se asegura que el sistema tenga un comportamiento robusto y escalable, como se detalla en el Anexo 8.

### 3.4.3.1 Entorno de programación

Con el fin de determinar el entorno de programación más adecuado para el desarrollo del sistema de detección de roya, se llevó a cabo una evaluación comparativa entre diversas plataformas de software. Tal como se presenta en la Tabla 22, el análisis consideró factores como la eficiencia en el uso de recursos, la estabilidad del entorno y su compatibilidad con librerías de visión artificial, flexible y eficiente para la ejecución y desarrollo del sistema.

**Tabla 22.**

Selección de entorno de programación

Entorno	SySR9	SySR12	SRS11	SRS13	SRS15	Valor total
Visual Studio Code	2	2	2	2	2	10
Spyder	1	2	1	2	2	8
PyCharm	2	2	2	1	2	9
<b>Cumple totalmente:</b>	2					
<b>Cumple parcialmente:</b>	1					
<b>No cumple:</b>	0					

*Nota.* Evaluación de entornos de programación. Elaborado por el autor.

**Elección:** Se eligió Visual Studio Code como el principal entorno de desarrollo porque satisface adecuadamente las necesidades del proyecto. El trabajo con librerías de visión artificial que se usan en muchas aplicaciones de inteligencia artificial es más fácil gracias a su compatibilidad con varios sistemas operativos y a un ambiente ágil y flexible, como se muestra en el Anexo 9.

### 3.4.4 Lenguaje de programación

Para establecer el lenguaje de programación más apropiado se debe analizar cada una de las opciones que se presentan en la Tabla 23. Para elegir el lenguaje de programación se apli-

can criterios como la capacidad para ejecutar procesos de detección simultánea, la rapidez de respuesta en tareas de visión artificial, la estabilidad en el procesamiento continuo, la compatibilidad con el manejo de imágenes y el soporte para tarjetas gráficas NVIDIA.

**Tabla 23.**

Selección del lenguaje de programación

Lenguaje	SySR6	SySR7	SySR9	SRS2	SRS10	Valor total
Python	2	2	2	2	2	10
C++	2	2	2	2	1	9
Java	1	1	1	2	1	6
MATLAB	2	1	1	1	1	6
<b>Cumple totalmente:</b>	2					
<b>Cumple parcialmente:</b>	1					
<b>No cumple:</b>	0					

*Nota.* Comparativa de lenguajes de programación según su compatibilidad y desempeño en entornos de visión artificial. Elaborado por el autor.

**Elección:** Según la evaluación técnica y los requisitos establecidos, se concluyó que el lenguaje más apropiado para desarrollar el sistema de detección de roya es Python. La capacidad de este lenguaje para trabajar con librerías específicas, como TensorFlow, OpenCV y Yolo, así como su eficacia en la manipulación de imágenes (que se muestran en el Anexo 10) son sus características destacadas.

### 3.4.5 Algoritmos de visión artificial

Se llevó a cabo una investigación comparativa de varios algoritmos utilizados para el procesamiento de imágenes con el objetivo de establecer cuál era el modelo de visión artificial más eficaz para detectar roya en rosas. La Tabla 24 se utilizó para evaluar las opciones estudiadas, que comprenden Haar Cascades, OpenCV, YOLOv5, MobileNet y SSD. La capacidad de estas opciones para ajustarse al sistema, la rapidez con que detectan y su precisión se consideró al evaluar su desempeño. Esta metodología hace posible identificar el modelo más adecuado para integrarse con el hardware seleccionado, lo que mejora la eficiencia general del sistema. En consecuencia, se garantiza que el reconocimiento en las pruebas de campo sea más confiable y efectivo.

**Tabla 24.**

Selección de algoritmos de visión artificial

Algoritmo / Librería	SySR6	SySR9	SySR12	SRS2	SRS12	Valor total
YOLOv5	2	2	2	2	2	10
MobileNet	2	2	1	2	1	8
SSD	2	1	1	2	1	7
Haar Cascades	1	1	1	1	1	5
OpenCV	2	1	2	2	1	8
<b>Cumple totalmente:</b>	2					
<b>Cumple parcialmente:</b>	1					
<b>No cumple:</b>	0					

*Nota.* Comparativa de algoritmos de visión artificial, considerando precisión, rapidez y compatibilidad. Elaborado por el autor.

**Elección:** YOLOv5 se seleccionó como algoritmo principal para el sistema de detección de roya debido a su alta precisión en la detección de objetos y su eficiencia en el procesamiento de imágenes en aplicaciones de monitoreo de cultivos. Su compatibilidad con distintos frameworks de inteligencia artificial lo convierte en una solución versátil y funcional para el prototipo desarrollado, como se detalla en el Anexo 11.

### 3.4.6 Aplicación web

El sistema de detección requiere una plataforma web que permita visualizar los resultados del análisis de imágenes además de gestionar los datos procesados y una interfaz accesible al usuario. La Tabla 25 muestra la comparación técnica entre las alternativas consideradas.

**Tabla 25.**

Selección de la aplicación web

Framework / Plataforma	SySR5	SySR9	SRS4	SRS14	SRS26	Valor total
Flask	2	2	2	2	2	10
WordPress	1	1	1	1	1	5
Django	2	2	2	2	1	9
Node.js + Express	2	2	2	1	2	9
<b>Cumple totalmente:</b>	2					
<b>Cumple parcialmente:</b>	1					
<b>No cumple:</b>	0					

*Nota.* Comparativa de entornos de desarrollo web. Elaborado por el autor.

**Elección:** Flask se eligió como la plataforma más apropiada para construir la aplicación web, puesto que es ligera, flexible y se integra fácilmente con varias tecnologías; por lo tanto, para proyectos que requieren un rendimiento alto y un uso eficaz de los recursos, representa una opción ideal. También, su estructura permite integrar módulos para procesar imágenes, implementar APIs, ejecutar modelos de inteligencia artificial y visualizar datos dinámicamente en una plataforma que se puede escalar y adaptar fácilmente, como se presenta en el Anexo 12.

### 3.4.7 Base de datos

Al seleccionar la base de datos para el sistema de detección de roya es necesario analizar la capacidad para gestionar grandes volúmenes de datos generados por el sistema de visión artificial y garantizar un acceso rápido y seguro a la información procesada. Como se muestra en la Tabla 26 se evaluaron distintas alternativas en función de su rendimiento, escalabilidad, compatibilidad con el entorno de monitoreo y facilidad de integración con el sistema.

**Tabla 26.**

Selección de la Base de Datos

Base de Datos	SRS11	SRS12	SRS14	SRS111	SRS115	Valor total
MySQL	2	2	2	2	2	10
PostgreSQL	2	2	2	2	1	9
SQLite	2	1	2	2	1	8
MongoDB	1	2	1	1	1	6
MariaDB	2	2	2	2	1	9
<b>Cumple totalmente:</b>	2					
<b>Cumple parcialmente:</b>	1					
<b>No cumple:</b>	0					

*Nota.* Comparativa de bases de datos evaluadas. Elaborado por el autor.

**Elección:** Tras evaluar diferentes opciones de bases de datos, se optó por MySQL porque puede gestionar grandes volúmenes de información de manera rápida y fácil, y también trabaja bien con sistemas Linux. Esta base de datos brinda estabilidad y confianza para guardar y procesar los datos que genera el sistema, como se menciona en el Anexo 13.

### 3.4.8 Aplicación para la planificación de vuelo

Para la planificación de rutas de vuelo del dron en el sistema de detección de roya, se evaluaron distintas aplicaciones de control y misión autónoma, considerando su compatibilidad con drones DJI, la capacidad de definir waypoints precisos, la estabilidad durante la misión y la facilidad de uso por parte del operador.

Como se muestra en la Tabla 27, se compararon varias alternativas en función de su integración con el dron seleccionado, sus opciones de personalización de trayectorias y su soporte para la gestión de misiones en entornos de cultivo.

**Tabla 27.**

Selección de la aplicación de planificación de vuelo

Aplicación	SRSH21	SRSH22	SRSH23	SRSH24	SRSH25	Valor total
Litchi	2	2	2	2	2	10
DJI Fly	2	1	2	1	1	7
DroneDeploy	1	2	2	1	1	7
<b>Cumple totalmente:</b>	2					
<b>Cumple parcialmente:</b>	1					
<b>No cumple:</b>	0					

*Nota.* Comparativa de aplicaciones de planificación de vuelo. Elaboración propia.

**Elección:** Después de la comparación efectuada, se eligió Litchi como la aplicación principal para el planeamiento del vuelo del dron. Esta plataforma permite gestionar misiones de manera confiable en entornos de cultivo, así como establecer rutas a través de waypoints con gran nivel de detalle y modificar la velocidad, los puntos de interés y la altura.

Asimismo, según se señala en el Anexo 14, cuenta con una interfaz intuitiva y una integración apropiada con drones DJI; esto permite realizar rutas repetibles sobre los invernaderos y ayuda a obtener imágenes de manera sistemática para el sistema de detección de roya.

### 3.5 Diseño del Sistema

Después de seleccionar los elementos básicos de hardware y software, se establecen las bases para crear el sistema propuesto. Estas pautas se crean según las normas vistas antes, para asegurar una buena y útil implementación del prototipo diseñado para detectar la roya (*Phragmidium mucronatum*) en cultivos de rosas.

En esta etapa, se establecen las funciones primordiales del sistema mediante representaciones gráficas como diagramas de flujo o diagramas de bloques. Estas herramientas visuales posibilitan una descripción ordenada de la manera en que opera el sistema. Esto facilita tanto la creación del código como la comprensión de los procesos involucrados en el funcionamiento del dron y el análisis de los datos producidos por el sistema de visión artificial.

#### Diagrama de bloques general del sistema

El funcionamiento del sistema para la detección de roya se organiza en cuatro bloques principales que cumplen funciones específicas dentro del flujo operativo del prototipo. Estos bloques estructuran el ciclo completo, desde la captura de imágenes hasta la entrega de resultados visuales como se indica en la Figura 23.

**Figura 23.**

*Diagrama de bloques del sistema*



**Nota.** Estructura general del sistema dividida en cuatro etapas: recolección de datos, procesamiento, tratamiento y visualización de resultados. Elaborado por el autor.

- **Primer bloque:** consiste en la recopilación de datos, que es el primer paso del sistema, en este punto, el dron se encarga de inspeccionar las áreas de rosas y toma imágenes

detalladas de los sitios que está analizando. Estas imágenes permiten la identificación precisa de las áreas afectadas por la roya, que se distinguen por manchas que van desde puntos amarillos y anaranjados hasta zonas con daños más severos, en las que las hojas presentan colores marrones. Asimismo, permiten identificar las áreas que permanecen saludables. Las imágenes, después de ser capturadas, son retiradas del dron y enviadas al siguiente segmento del procedimiento para su análisis y procesamiento.

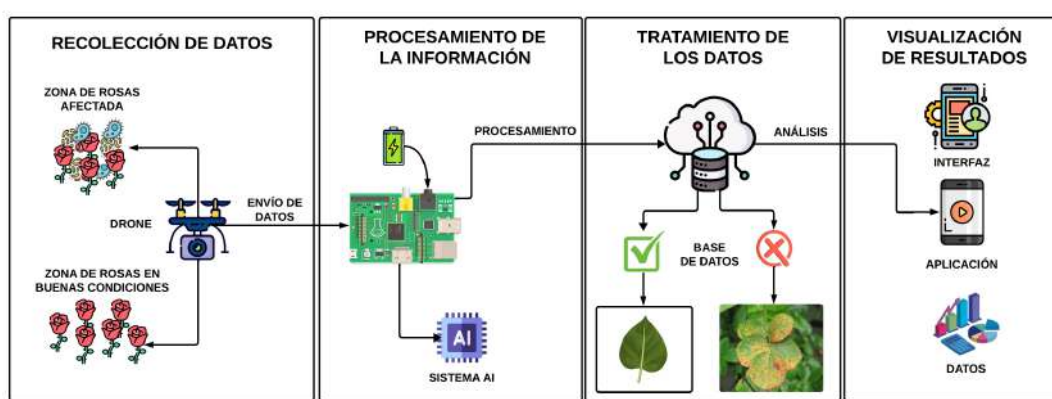
- **Segundo bloque:** Se enfoca en el procesamiento de información, como las imágenes tomadas y enviadas a un sistema basado en inteligencia artificial que utiliza el algoritmo YOLOv5, que está especializado para detectar roya. En esta etapa, se utilizan métodos sofisticados de visión artificial que examinan detalladamente cada imagen, lo que asegura la detección exacta del hongo en las zonas afectadas de la cosecha.
- **Tercer bloque:** Se fundamenta en el procesamiento de datos; la información que se obtiene después de este procesamiento se guarda en una base de datos centralizada y organizada. En esta fase, se analizan y ordenan los datos de manera detallada para identificar patrones concretos que ayuden a corroborar la presencia del hongo que causa la roya. Asimismo, se detectan las áreas críticas del cultivo que necesitan atención urgente y se producen informes minuciosos que hacen más fácil tomar decisiones informadas. Para la creación de estrategias eficaces y precisas para la gestión y control de la enfermedad, así como para proteger la salud de los cultivos y optimizar recursos, esta información es esencial.
- **Cuarto bloque:** Por último, en la etapa de visualización de los resultados, se incorporan los datos procesados a una aplicación web creada con Flask. Esta aplicación presenta una interfaz intuitiva y sencilla de utilizar. Es factible, mediante esta plataforma, observar las imágenes analizadas y los resultados producidos por el sistema de inteligencia artificial. Esta visualización simplifica el trabajo de los agricultores y técnicos al calificar la severidad de las áreas impactadas y tomar acciones adecuadas para reducir riesgos y maximizar el empleo de fungicidas.

## Arquitectura del sistema

La Figura 24 presenta la arquitectura del sistema de forma más detallada en el proceso por bloques del prototipo. Esta representación visual facilita la comprensión de cómo se desarrolla cada una de los bloques mencionadas anteriormente, desde la captura de imágenes hasta la visualización de resultados para la detección de roya.

**Figura 24.**

*Arquitectura del sistema*



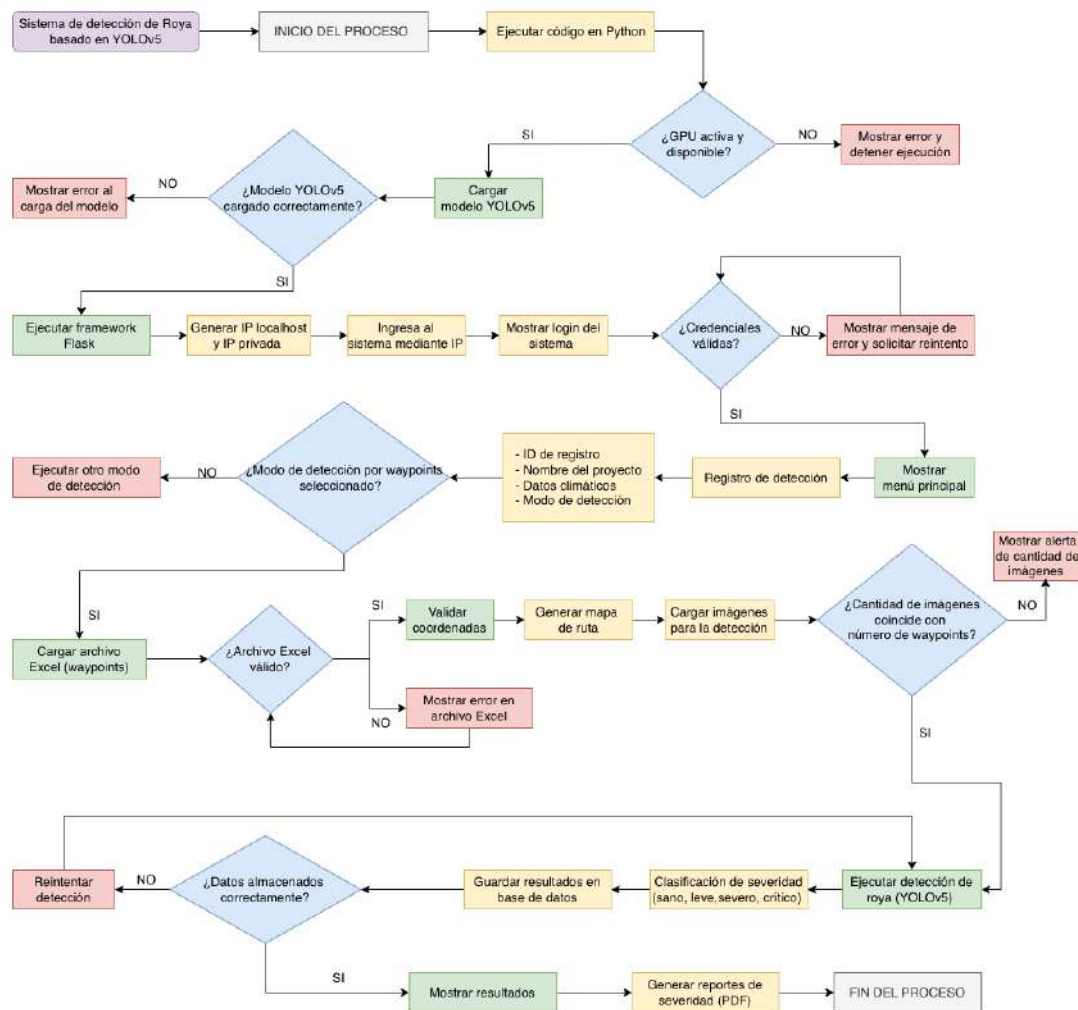
*Nota.* Representación de la arquitectura del sistema. Elaborado por el autor.

## Diagrama de flujo del sistema

El flujograma del sistema permite comprender el funcionamiento del sistema de detección de Roya. El proceso comienza con la verificación de hardware y software necesarios para iniciar el sistema. Una vez validados correctamente, el sistema permite el acceso a la aplicación web, la cual facilita la interacción con el usuario para el registro y la configuración de la detección por waypoints. Para ello se carga el archivo Excel con la ruta del trayecto y las imágenes capturadas, ya sean terrestres o aéreas, dando inicio al proceso de detección mediante técnicas de visión artificial. Posteriormente, la información generada es tratada y almacenada para finalmente presentar los resultados a través de una galería que muestra las detecciones obtenidas según el nivel de severidad, junto con el reporte pdf con resultados de los niveles de severidad detectados en el cultivo. La Figura 25 muestra este flujo de operación.

Figura 25.

Diagrama de flujo del sistema



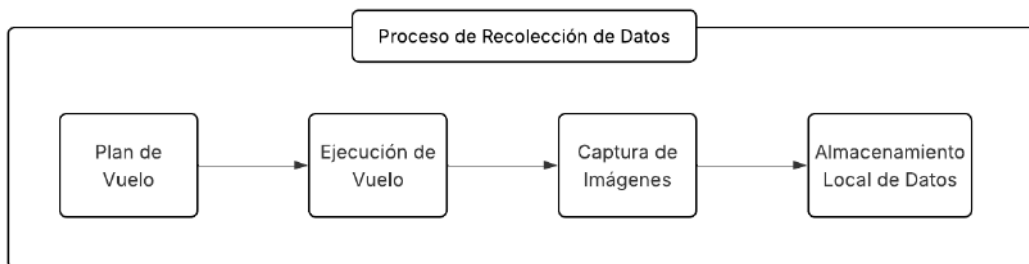
*Nota.* Elaborado por el autor.

### 3.5.1 Bloque 1: recolección de datos

El proceso de recolección de datos permite identificar áreas afectadas por la plaga mediante la planificación del vuelo, la captura de imágenes del cultivo con el dron y el almacenamiento local de la información para su posterior análisis. La Figura 26 presenta el diagrama de bloques de este proceso.

**Figura 26.**

*Diagrama de bloques de la recolección de datos*

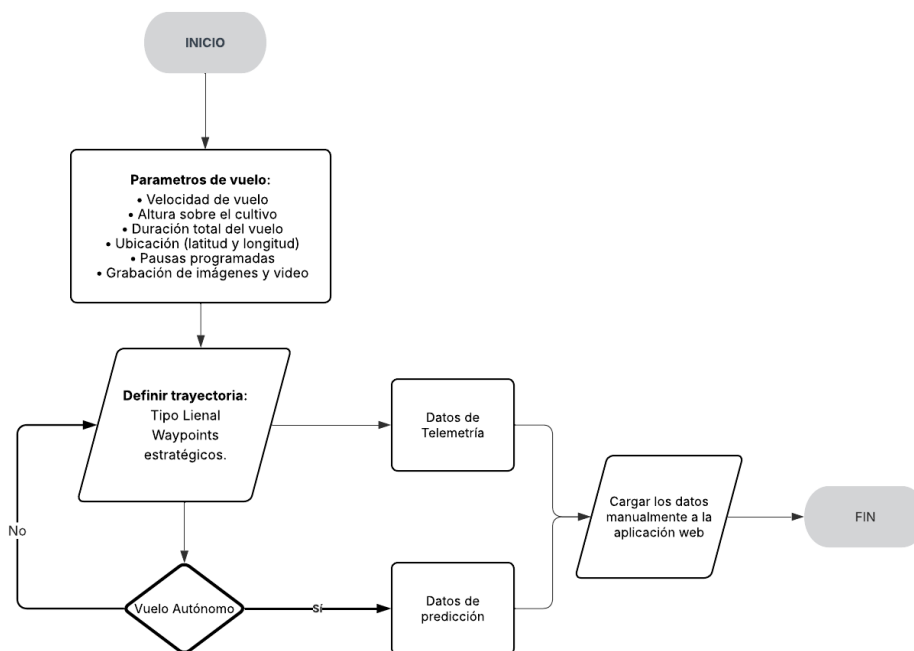


*Nota.* Diagrama de bloques del proceso de recolección de datos, desde el plan de vuelo hasta el almacenamiento local. Elaborado por el autor.

**Plan de vuelo:** Esta fase constituye el punto de partida del bloque de recolección de datos, en la cual se definen con precisión la trayectoria, los parámetros de vuelo y la ruta específica que seguirá el dron sobre el área de cultivo, la cual abarca aproximadamente 280 m<sup>2</sup>. La Figura 27 muestra el diagrama de flujo correspondiente al plan de vuelo.

**Figura 27.**

*Diagrama de flujo del plan de vuelo*



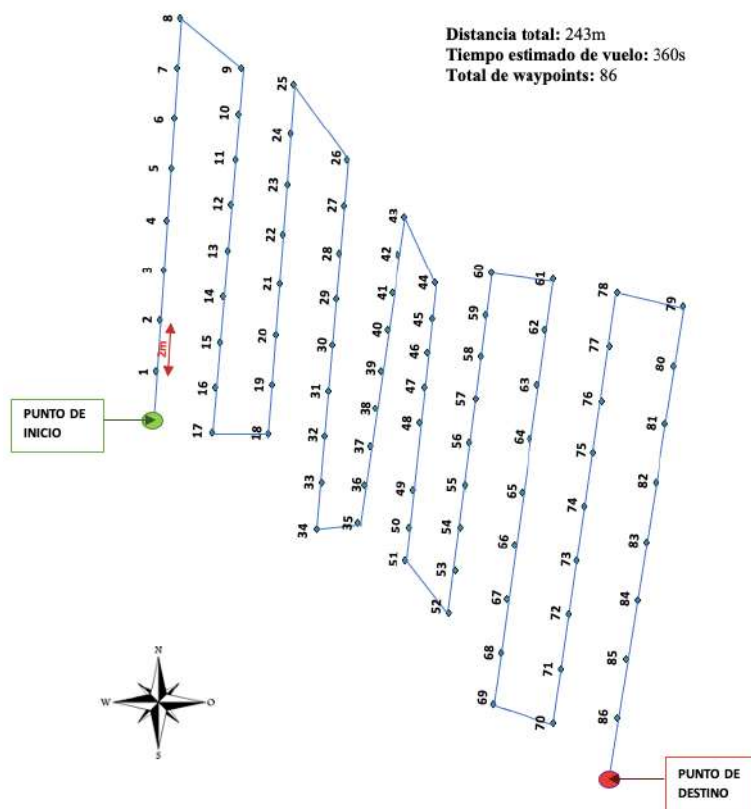
*Nota.* Proceso desde la planificación de vuelo. Elaborado por el autor.

En complemento a lo anterior, se definieron parámetros fundamentales como la ubicación exacta del terreno a inspeccionar, las coordenadas de cada waypoint y las distancias entre ellos, los cuales se detallan con mayor profundidad en el Anexo 15. Una vez establecidos estos parámetros, se procedió con el diseño del ortomapa, que serviría como guía para la ruta de vuelo del dron.

En este diseño se establecieron 86 waypoints, distribuidos a una distancia de 2 metros uno del otro, con el objetivo de evitar solapamientos en la captura de imágenes y garantizar una cobertura completa del área de cultivo. A su vez, la planificación considera la presencia de posibles obstáculos dentro del perímetro, optimizando la trayectoria para prevenir colisiones y mantener una distancia de seguridad adecuada. Como resultado, se obtuvo un diseño poligonal de trayectoria, que permite un desplazamiento sistemático y eficiente durante la misión de vuelo como se muestra en la Figura 28.

**Figura 28.**

*Ortomapa del plan de vuelo*



**Nota.** Ortomapa del plan de vuelo del dron. Elaborado por el autor.

El diseño del ortomapa hace que el dron siga un camino preestablecido, formado por varios puntos de referencia ubicados de manera estratégica para mejorar la cobertura del área que se está estudiando. Así, se disminuye el margen de error en el camino de vuelo. En la Tabla 28 se muestran los resultados obtenidos, que aseguran que cada waypoint esté bien colocado y a la distancia correcta, evitando superposiciones en la captura de imágenes y posibles choques con objetos en el entorno.

**Tabla 28.**

Coordenadas y parámetros de vuelo del dron

<b>ID</b>	<b>LONGITUD</b>	<b>LATITUD</b>	<b>ALTURA</b> <b>(m)</b>	<b>DISTANCIA</b> <b>(m)</b>	<b>VELOCIDAD</b> <b>(m/s)</b>	<b>TIEMPO DE</b> <b>VUELO (s)</b>
1	-78.193.006	0.085823	2.5	2	1	360
2	-78.193.012	0.085793	2.5	4	1	360
3	-78.193.018	0.085764	2.5	6	1	360
4	-78.193.024	0.085735	2.5	8	1	360
5	-78.193.030	0.085705	2.5	10	1	360
6	-78.193.036	0.085676	2.5	12	1	360
7	-78.193.043	0.085647	2.5	14	1	360
8	-78.193.049	0.085617	2.5	16	1	360
9	-78.193.055	0.085588	2.5	18	1	360
10	-78.193.090	0.085615	2.5	20	1	360
11	-78.193.085	0.085642	2.5	22	1	360
12	-78.193.080	0.085669	2.5	24	1	360
13	-78.193.075	0.085696	2.5	26	1	360
14	-78.193.070	0.085723	2.5	28	1	360
15	-78.193.064	0.085750	2.5	30	1	360
16	-78.193.059	0.085776	2.5	32	1	360
17	-78.193.054	0.085803	2.5	34	1	360
18	-78.193.049	0.085830	2.5	37	1	360
19	-78.193.090	0.085837	2.5	40	1	360
20	-78.193.095	0.085811	2.5	43	1	360
21	-78.193.100	0.085784	2.5	46	1	360
22	-78.193.105	0.085757	2.5	49	1	360

Continúa en la siguiente página

Cuadro 28 – continuación de la página anterior

ID	LONGITUD	LATITUD	ALTURA (m)	DISTANCIA (m)	VELOCIDAD (m/s)	TIEMPO DE VUELO (s)
23	-78.193.110	0.085730	2.5	52	1	360
24	-78.193.115	0.085704	2.5	55	1	360
25	-78.193.120	0.085677	2.5	58	1	360
26	-78.193.124	0.085650	2.5	61	1	360
27	-78.193.129	0.085624	2.5	64	1	360
28	-78.193.157	0.085670	2.5	67	1	360
29	-78.193.153	0.085691	2.5	70	1	360
30	-78.193.150	0.085713	2.5	73	1	360
31	-78.193.147	0.085734	2.5	76	1	360
32	-78.193.144	0.085755	2.5	79	1	360
33	-78.193.140	0.085777	2.5	82	1	360
34	-78.193.137	0.085798	2.5	85	1	360
35	-78.193.134	0.085819	2.5	88	1	360
36	-78.193.130	0.085840	2.5	91	1	360
37	-78.193.166	0.085835	2.5	94	1	360
38	-78.193.168	0.085817	2.5	97	1	360
39	-78.193.171	0.085800	2.5	100	1	360
40	-78.193.173	0.085783	2.5	103	1	360
41	-78.193.176	0.085765	2.5	106	1	360
42	-78.193.180	0.085731	2.5	109	1	360
43	-78.193.183	0.085713	2.5	112	1	360
44	-78.193.185	0.085696	2.5	115	1	360
45	-78.193.216	0.085715	2.5	118	1	360
46	-78.193.212	0.085734	2.5	121	1	360
47	-78.193.208	0.085753	2.5	124	1	360
48	-78.193.205	0.085772	2.5	127	1	360
49	-78.193.201	0.085791	2.5	130	1	360
50	-78.193.197	0.085811	2.5	133	1	360
51	-78.193.194	0.085830	2.5	136	1	360
52	-78.193.190	0.085849	2.5	139	1	360
53	-78.193.186	0.085868	2.5	142	1	360

Continúa en la siguiente página

Cuadro 28 – continuación de la página anterior

<b>ID</b>	<b>LONGITUD</b>	<b>LATITUD</b>	<b>ALTURA</b> <b>(m)</b>	<b>DISTANCIA</b> <b>(m)</b>	<b>VELOCIDAD</b> <b>(m/s)</b>	<b>TIEMPO DE</b> <b>VUELO (s)</b>
54	-78.193.223	0.085896	2.5	145	1	360
55	-78.193.225	0.085873	2.5	148	1	360
56	-78.193.228	0.085850	2.5	151	1	360
57	-78.193.230	0.085827	2.5	154	1	360
58	-78.193.232	0.085804	2.5	157	1	360
59	-78.193.235	0.085781	2.5	160	1	360
60	-78.193.237	0.085758	2.5	163	1	360
61	-78.193.239	0.085735	2.5	166	1	360
62	-78.193.242	0.085712	2.5	169	1	360
63	-78.193.273	0.085759	2.5	172	1	360
64	-78.193.271	0.085784	2.5	175	1	360
65	-78.193.268	0.085809	2.5	178	1	360
66	-78.193.266	0.085834	2.5	181	1	360
67	-78.193.264	0.085859	2.5	184	1	360
68	-78.193.261	0.085884	2.5	187	1	360
69	-78.193.259	0.085909	2.5	190	1	360
70	-78.193.257	0.085934	2.5	193	1	360
71	-78.193.291	0.085942	2.5	196	1	360
72	-78.193.293	0.085919	2.5	199	1	360
73	-78.193.295	0.085896	2.5	202	1	360
74	-78.193.298	0.085874	2.5	205	1	360
75	-78.193.300	0.085851	2.5	208	1	360
76	-78.193.303	0.085828	2.5	211	1	360
77	-78.193.305	0.085805	2.5	214	1	360
78	-78.193.308	0.085783	2.5	217	1	360
79	-78.193.310	0.085760	2.5	220	1	360
80	-78.193.348	0.085766	2.5	223	1	360
81	-78.193.346	0.085791	2.5	226	1	360
82	-78.193.343	0.085816	2.5	229	1	360
83	-78.193.341	0.085841	2.5	232	1	360
84	-78.193.339	0.085866	2.5	235	1	360

Continúa en la siguiente página

Cuadro 28 – continuación de la página anterior

ID	LONGITUD	LATITUD	ALTURA (m)	DISTANCIA (m)	VELOCIDAD (m/s)	TIEMPO DE VUELO (s)
85	-78.193.336	0.085892	2.5	238	1	360
86	-78.193.334	0.085917	2.5	241	1	360
87	-78.193.332	0.085942	2.5	244	1	360
88	-78.193.330	0.085967	2.5	247	1	360

**Nota.** Tabla de coordenadas con los waypoints programados para el dron, incluyendo la altura, la distancia y el tiempo estimado de vuelo.

Una vez diseñado el plan de vuelo, es necesario obtener una visualización completa del trayecto que recorrerá el dron, es por eso que se utilizó el software Google Earth, que permite trazar y analizar con precisión la ruta planificada, proporcionando una representación geoespacial del recorrido.

Esta visualización facilita la verificación del punto de inicio, la trayectoria completa y el punto final del vuelo, tal como se muestra en la Figura 29.

### Figura 29.

*Diseño de la ruta de vuelo en Google Earth*

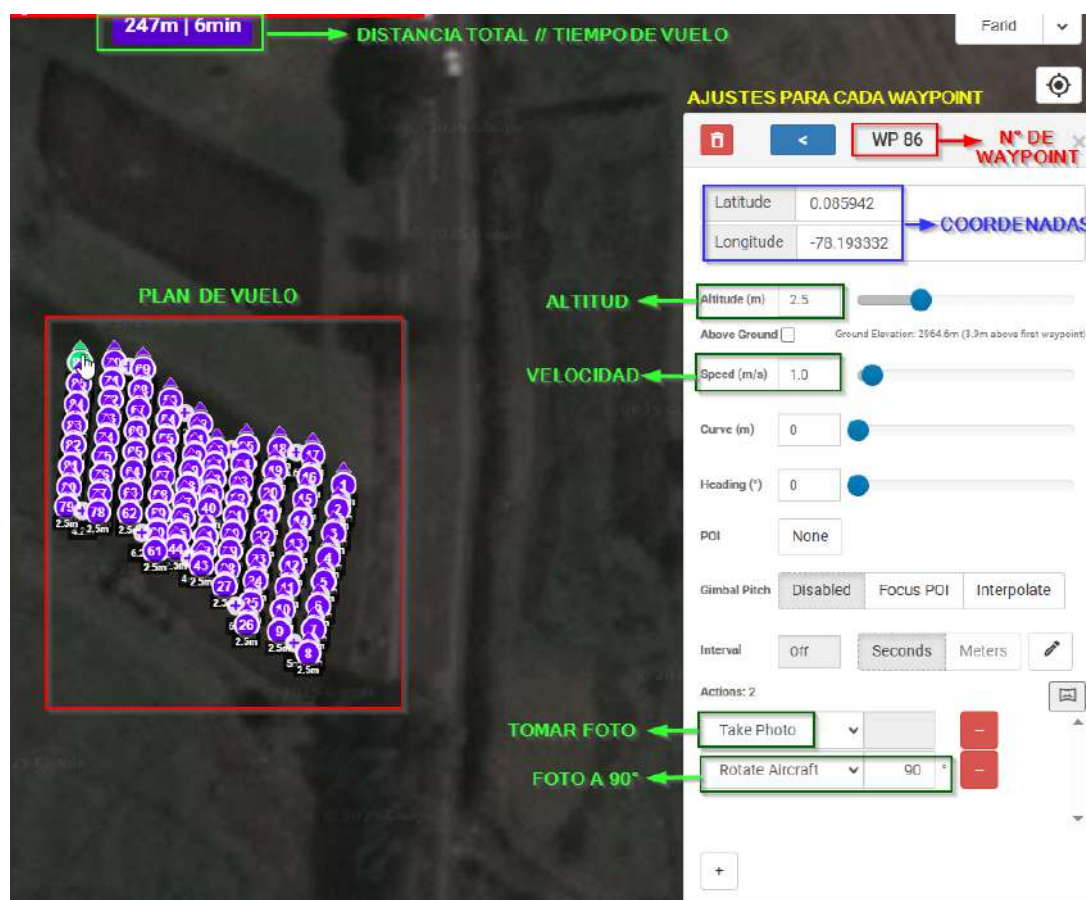


**Nota.** Diseño de la ruta de vuelo del dron en Google Earth, indicando los puntos de inicio y destino para el monitoreo del cultivo. Elaborado por el autor.

Finalmente para la planificación del vuelo se lo realizó mediante la plataforma Litchi, dentro de la zona de Tabacundo. En su interfaz se visualiza la disposición de los 86 waypoints que definen el recorrido del dron, garantizando una cobertura completa del área de cultivo. Además, Litchi permite obtener una vista tridimensional del entorno y ajustar parámetros esenciales, como la altura de vuelo, la velocidad de desplazamiento, el tiempo total de vuelo, así como la configuración de captura de imágenes y el posicionamiento preciso de cada waypoint. Estos elementos se observan en la Figura 30.

**Figura 30.**

*Interfaz de planificación de ruta de vuelo en Litchi*



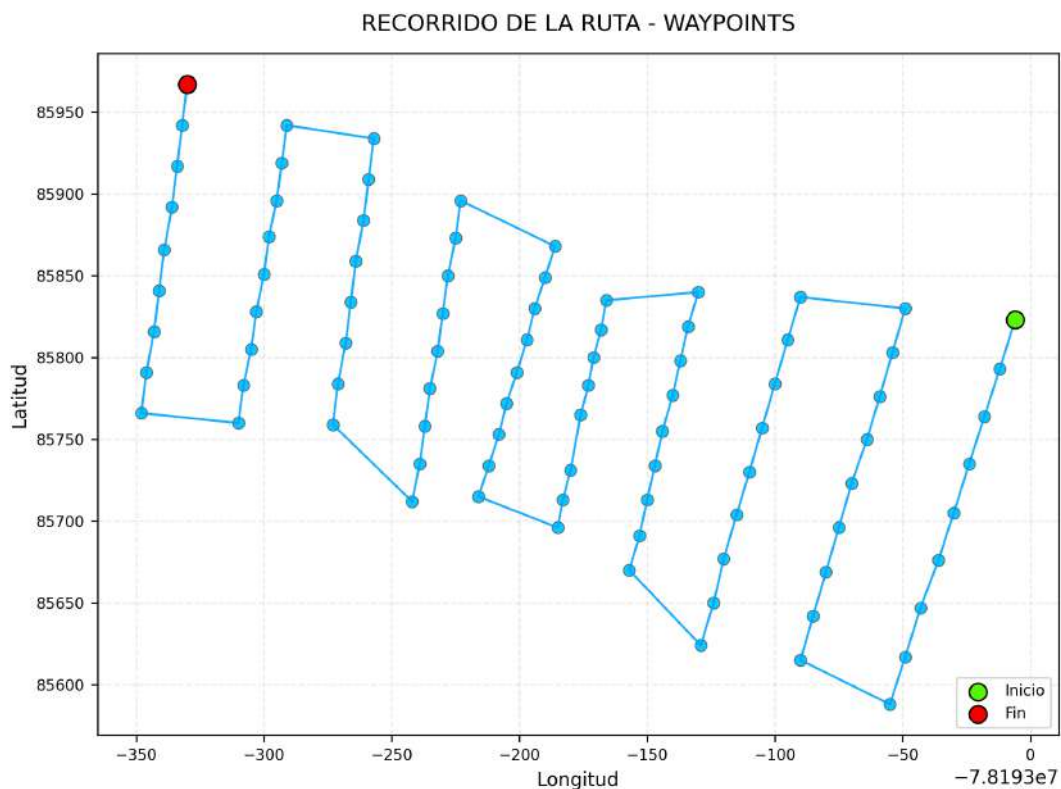
**Nota.** Configuración de la misión aérea en Litchi para el monitoreo en el cultivo. Elaborado por el autor.

Es posible importar los datos de las coordenadas de los waypoints, que se encuentran en el archivo .csv creado en Excel, a Python para analizarlos y representarlos gráficamente. Utilizando bibliotecas específicas, se puede ver claramente el trayecto planificado, reconociendo cada waypoint de manera secuencial y los puntos inicial y final de la ruta.

Esta representación facilita la validación de la planificación del vuelo, asegurando una adecuada cobertura del área de estudio y un desplazamiento coherente del dron, tal como se muestra en la Figura 31.

**Figura 31.**

*Visualización del recorrido de la ruta mediante waypoints en Python*



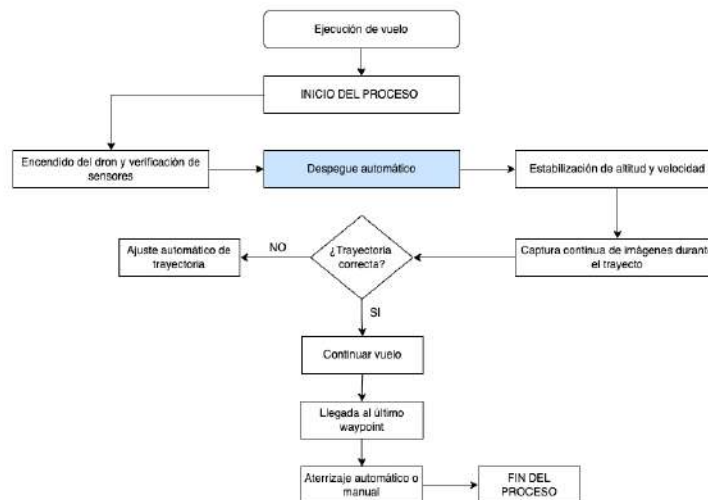
**Nota.** Mapa del recorrido generado en Python.

**Ejecución de Vuelo:** Es la segunda fase del proceso de recopilación de datos. Aquí el dron sigue de forma autónoma la ruta previamente planeada, volando por el área de estudio a través de los waypoints establecidos y conservando invariables las condiciones de vuelo, como la velocidad, la altura y la dirección.

El sistema de control es responsable de supervisar y corregir el movimiento en tiempo real, asegurando así un desplazamiento seguro y estable. A la misma vez, los sensores del dron que está a bordo recogen datos sobre la altitud, la dirección y las condiciones ambientales. Esto posibilita garantizar la exactitud del vuelo y la calidad de los datos adquiridos. El flujo de esta fase del proceso se muestra en la Figura 32.

**Figura 32.**

*Diagrama de flujo de la ejecución de vuelo*



**Nota.** Proceso autónomo del dron durante el vuelo, desde el despegue hasta el aterrizaje, con monitoreo continuo de su trayectoria. Elaborado por el autor.

Durante la fase de ejecución del vuelo es necesario colocar el dron exactamente en el punto de inicio definido previamente en el plan de vuelo. Para garantizar un despegue seguro, el dron debe colocarse sobre una superficie plana, permitiendo que el GPS se calibre correctamente y obtenga las coordenadas de latitud y longitud asociadas al primer waypoint. Esto asegura que el recorrido siga la trayectoria establecida y evita posibles colisiones con objetos presentes en la ruta de vuelo. La Figura 33 muestra el punto de despegue desde el cual se inicia la misión de vuelo.

**Figura 33.**

*Punto de inicio para la ejecución del vuelo*

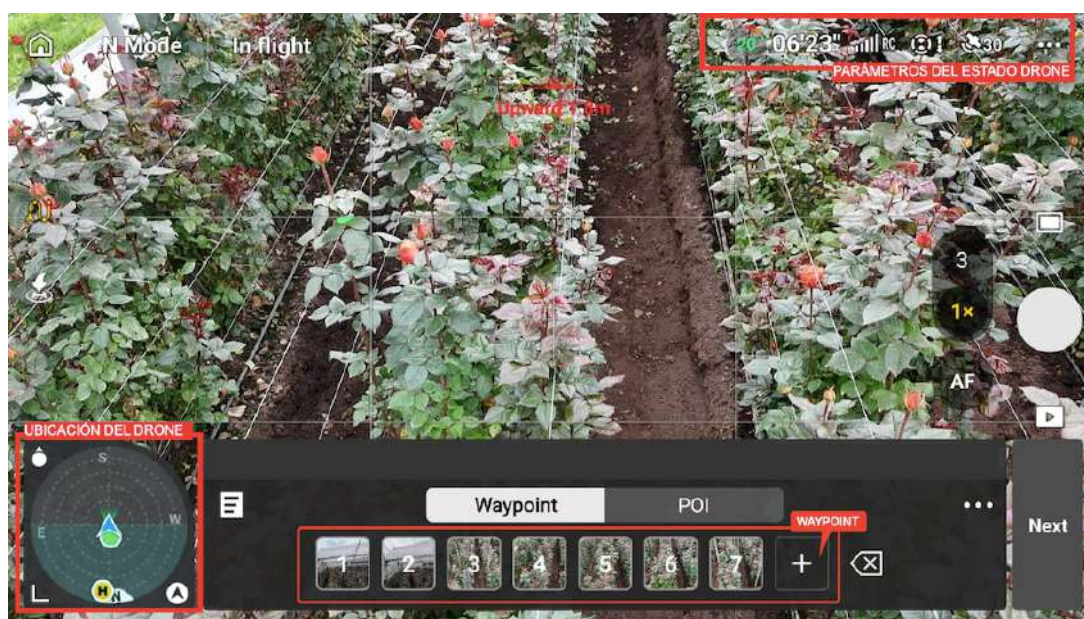


**Nota.** Zona designada como punto de inicio del vuelo del dron durante la misión aérea. Elaborado por el autor.

Una vez ubicado el dron en el punto de inicio, se procede a ejecutar la misión programada en la plataforma Litchi. Durante el vuelo, es posible visualizar en tiempo real el recorrido del dron y el avance por cada waypoint. El dron sigue la ruta respetando los parámetros configurados, como la altura, la distancia entre puntos y la captura de imágenes del cultivo con el ángulo definido para obtener fotografías precisas. La Figura 34 muestra el trayecto que realiza el dron dentro de la plataforma Litchi.

**Figura 34.**

*Inicio de pruebas de vuelo en el invernadero*



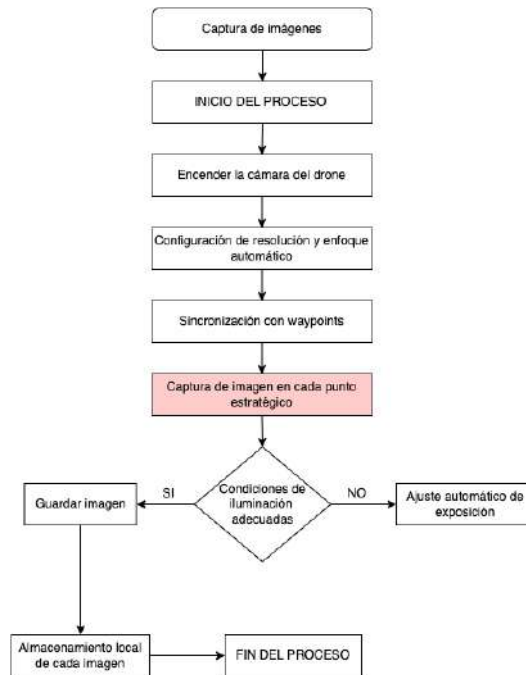
**Nota.** Momento previo al despegue del dron, posicionado para iniciar la misión de monitoreo aéreo. Elaborado por el autor.

**Captura de imágenes:** Una vez que el dron se encuentra en pleno vuelo, comienza a tomar imágenes del cultivo siguiendo la trayectoria planificada. Durante esta fase, la cámara del dron captura continuamente imágenes del cultivo, con especial atención en las hojas, donde suelen aparecer los primeros signos visibles de roya.

La frecuencia de captura está sincronizada con los waypoints, asegurando que se obtenga al menos una imagen representativa por cada punto estratégico del trayecto. La Figura 35 muestra el proceso de captura de imágenes realizado por el dron durante el vuelo.

**Figura 35.**

*Diagrama de flujo de la captura de imágenes*



**Nota.** Proceso automatizado de captura de imágenes por parte del dron durante el vuelo, sincronizado con los waypoints. Elaborado por el autor.

Como se observa en la Figura 36, el dron sigue la trayectoria planificada, recorriendo el cultivo y capturando imágenes en cada waypoint establecido. Esto permite obtener una cobertura completa del área y asegurar una adecuada calidad en las fotografías registradas.

**Figura 36.**

*Captura de imagen del dron.*

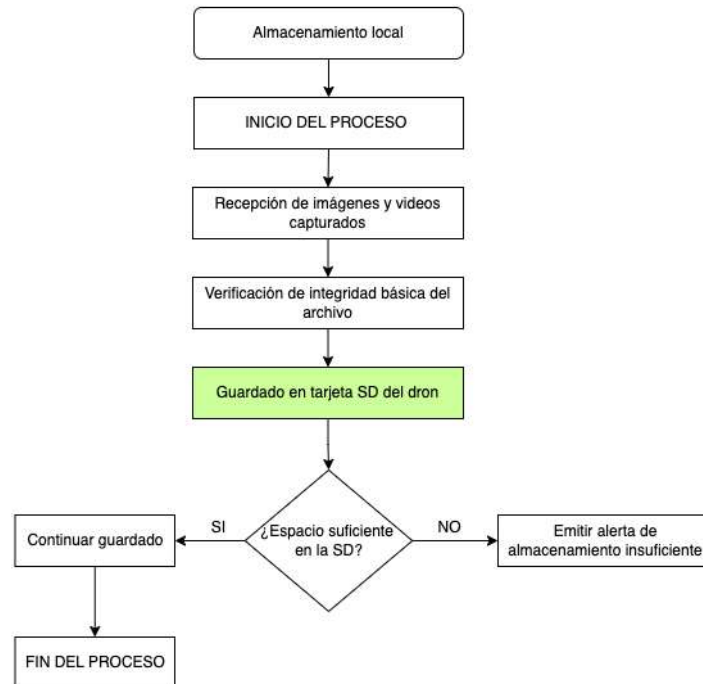


**Nota.** Configuración para la captura de imágenes. Elaborado por el autor.

**Almacenamiento local de datos:** representa la fase final del bloque de recolección, donde toda la información recopilada durante el vuelo principalmente imágenes y metadatos de vuelo es guardada directamente en el sistema de almacenamiento del dron. La Figura 37 presenta el flujo de almacenamiento local realizado al finalizar la captura de imágenes.

**Figura 37.**

*Diagrama de flujo del almacenamiento local de datos*



**Nota.** Proceso de almacenamiento de imágenes y videos en la tarjeta SD del dron durante el vuelo. Elaborado por el autor.

Este proceso garantiza que los datos capturados, como fotografías y videos, queden respaldados de forma segura antes de ser transferidos al sistema de detección para su procesamiento.

La Figura 38 muestra una fotografía tomada por el dron y almacenada en su memoria, lo que asegura la integridad de la información y contribuye a que el análisis posterior sea preciso y confiable.

**Figura 38.**

*Fotografía del cultivo de rosas almacenada en la memoria local del dron.*



DJI_20251127115445_0019_D.JPG	
Imagen JPEG - 8,9 MB	
<b>Información</b> <a href="#">Mostrar menos</a>	
Creación	27 de noviembre de 2025, 11:54
Modificación	27 nov. 2025 11:54
Creación del contenido	27/11/25 11:54
Dimensiones	4032x2268
Resolución	72x72
Espacio de color	RGB
Perfil de color	sRGB IEC61966-2.1
Marca del dispositivo	DJI
Modelo del dispositivo	FC8282
Valor de apertura	1,53
Tiempo de exposición	1/500
Longitud focal	6,72 mm
Velocidad ISO	100
Flash	No
Número f	f/1,7
Programa de exposición	Normal
Modalidad de medición	Promedio
Balances de blancos	Automático
Creador del contenido	10.08.09.57
Longitud	78° 11' 35,064" O
Latitud	0° 5' 8,471" N
<b>Etiquetas</b>	
<a href="#">Agregar etiquetas...</a>	

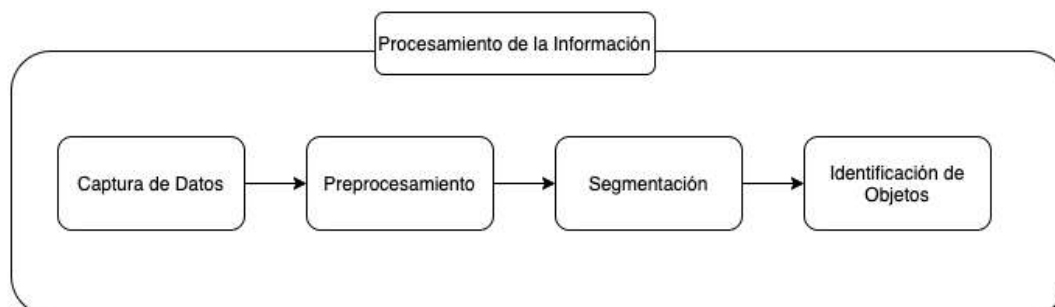
*Nota.* La fotografía del cultivo es almacenada en la memoria local del dron. Elaborado por el autor.

### 3.5.2 Bloque 2: procesamiento de la información

Para llevar a cabo el procesamiento de la información, se definieron cuatro etapas principales: captura de datos, preprocesamiento, segmentación e identificación de objetos. Este procedimiento se resume en la Figura 39, donde se representa de forma estructurada cada paso del proceso.

**Figura 39.**

*Diagrama de bloques del procesamiento de la información*



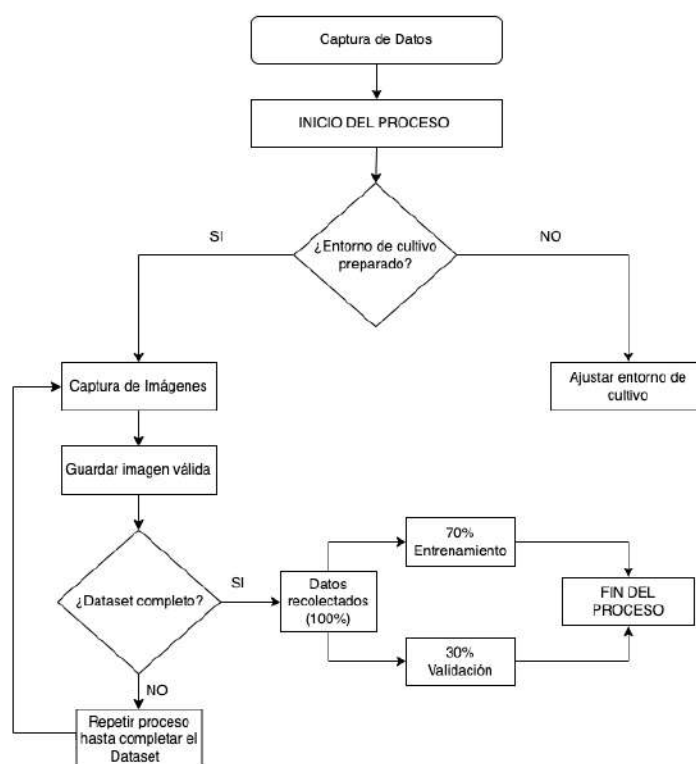
*Nota.* Ilustración propia que representa las cuatro etapas del procesamiento de datos en el sistema.

**Captura de datos:** Se reunió un grupo de 2,000 fotografías elegidas meticulosamente para ilustrar diferentes grados de daño por roya en las hojas de rosa. Este conjunto de datos se dividió de manera estratégica, con el 70 % de las imágenes destinadas a entrenar el modelo y el 30 % restante para validarlo. Esta distribución posibilita optimizar el aprendizaje automático y aumentar la exactitud en la detección del hongo *Phragmidium mucronatum*, lo que mejora la capacidad de generalización del sistema inteligente.

La Figura 40 ilustra el procedimiento total, que incluye la recolección de imágenes, la validación del ambiente de captura y la clasificación sistemática de los datos obtenidos.

**Figura 40.**

*Diagrama de flujo del proceso de captura de datos*



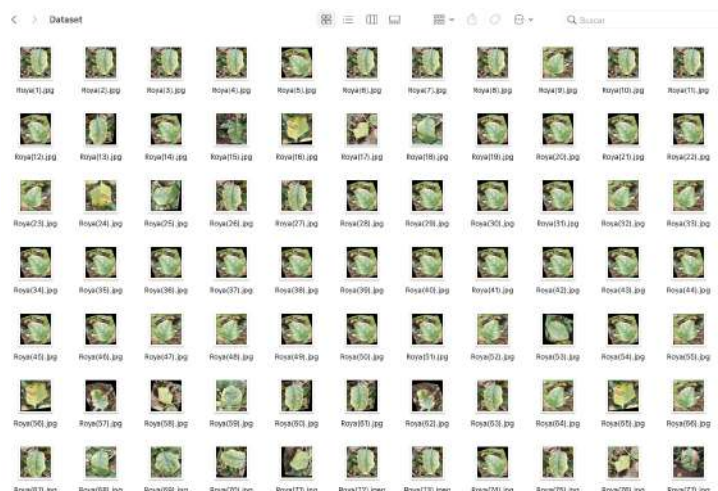
**Nota.** Ilustración propia que muestra cómo se recolecta y valida la información visual.

Es fundamental asegurarse de que las condiciones del entorno de cultivo sean las adecuadas antes de recolectar datos, ya que esto garantiza que las imágenes obtenidas sean claras y precisas. Una buena calidad visual es clave para que las imágenes puedan incorporarse al conjunto de entrenamiento y para detectar de manera confiable la presencia del hongo en las hojas de los rosales.

La Figura 41 ilustra cómo se recopilan las imágenes necesarias para completar el conjunto de datos, mostrando todo el proceso, desde la verificación del entorno hasta la conformación del conjunto final.

**Figura 41.**

*Imágenes capturadas para el dataset de roya*

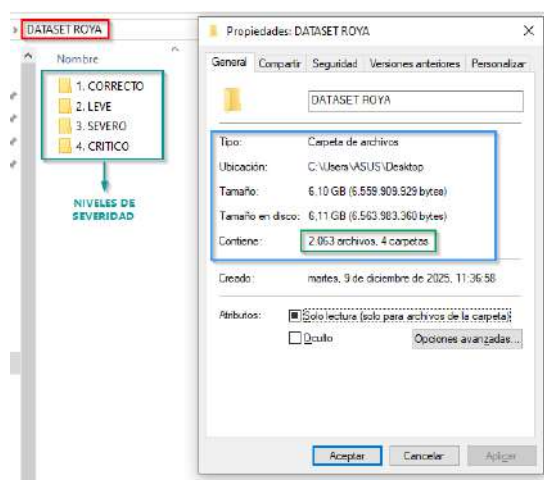


**Nota.** Vista general del dataset de imágenes de hojas con síntomas de roya, utilizado en la etapa de procesamiento del sistema. Elaborado por el autor.

Tras completar la captura de datos para el dataset, se procede a organizar las imágenes de acuerdo con el nivel de severidad de la roya. En la figura 42 se aprecia cómo el material recopilado se distribuye en carpetas independientes para cada categoría (correcto, leve, severo y crítico). También se visualiza información general del conjunto de datos, como la cantidad total de archivos como lo es las 2000 imágenes, lo cual proporciona una base suficientemente amplia para que el modelo de detección alcance un desempeño adecuado.

**Figura 42.**

*Organización del dataset de roya por niveles de severidad*



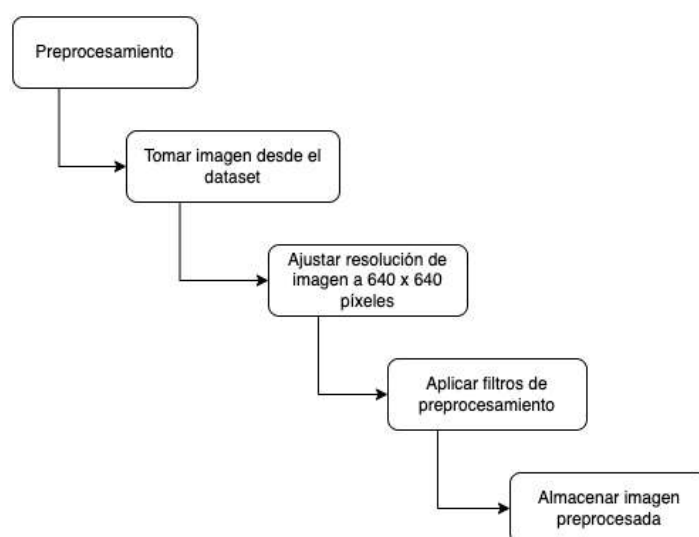
**Nota.** Estructura del dataset de imágenes de roya agrupadas en carpetas según el nivel de severidad. Elaborado por el autor.

**Preprocesamiento:** en este apartado se aplicaron filtros de suavizado para reducir el ruido presente en las fotografías, eliminando detalles irrelevantes que podrían interferir con la detección precisa de la roya. Posteriormente se procedió al ajuste de resolución de todas las imágenes a un formato estándar de 640 x 640 píxeles. Esta estandarización permite que el modelo procese los datos de manera uniforme, reduciendo la carga computacional y mejorando la eficiencia del entrenamiento.

Estas acciones no solo mejoran la calidad visual de los datos, sino que también facilitan etapas posteriores como la corrección de color, la segmentación y la identificación de patrones visuales. Gracias a este proceso, se garantiza que las imágenes sean compatibles con los algoritmos de visión artificial y que el sistema pueda operar con mayor precisión y consistencia en la detección de la roya en hojas de rosales. El proceso descrito se indica en la Figura 43, donde se representa la secuencia de pasos aplicada para la preparación de las imágenes.

**Figura 43.**

*Diagrama de bloques del proceso de preprocesamiento*

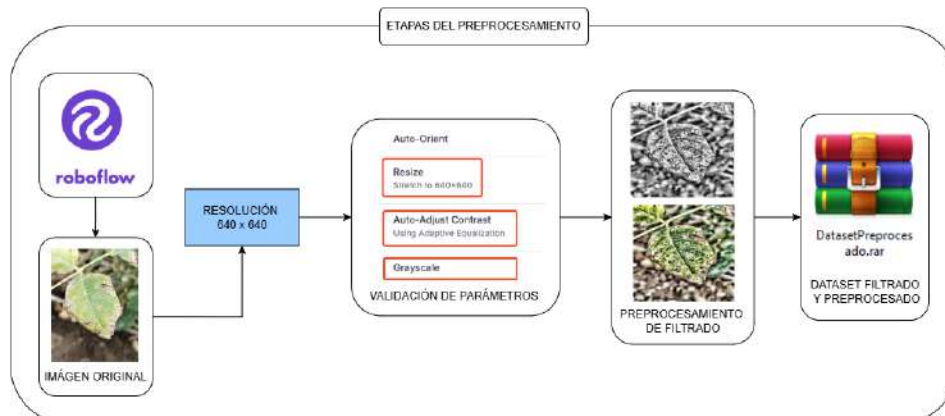


**Nota.** Ilustración propia que muestra cómo se filtran, ajustan y almacenan las imágenes antes del entrenamiento del modelo.

Este proceso se aplica a las imágenes capturadas por el dron y es fundamental para optimizar su calidad antes de ser analizadas por el sistema de detección de roya. Incluye operaciones como el redimensionamiento, mejora de contraste, filtrado y normalización, lo que permite preparar las imágenes para una detección más precisa y eficiente. Para una mejor comprensión, la Figura 44 presenta de forma estructurada el flujo de las etapas que conforman el preprocesamiento de datos.

**Figura 44.**

Diagrama del proceso de preprocesamiento



**Nota.** Ilustración propia que muestra el flujo de preprocesamiento aplicado a las imágenes recolectadas.

El proceso inicia con la carga de la imagen original en la plataforma Roboflow, representada en la parte izquierda del diagrama. Esta imagen corresponde a una imagen de una hoja capturada en el cultivo que se encuentra con posibles signos de infección por *Phragmidium mucronatum*.

Posteriormente, se aplica una etapa de redimensionamiento a una resolución estándar de 640×640 píxeles, lo cual garantiza uniformidad en el dataset y mejora la eficiencia durante el entrenamiento del modelo de detección. A esta transformación le sigue una fase de validación de parámetros, donde se especifican los ajustes que serán aplicados a todas las imágenes, tales como:

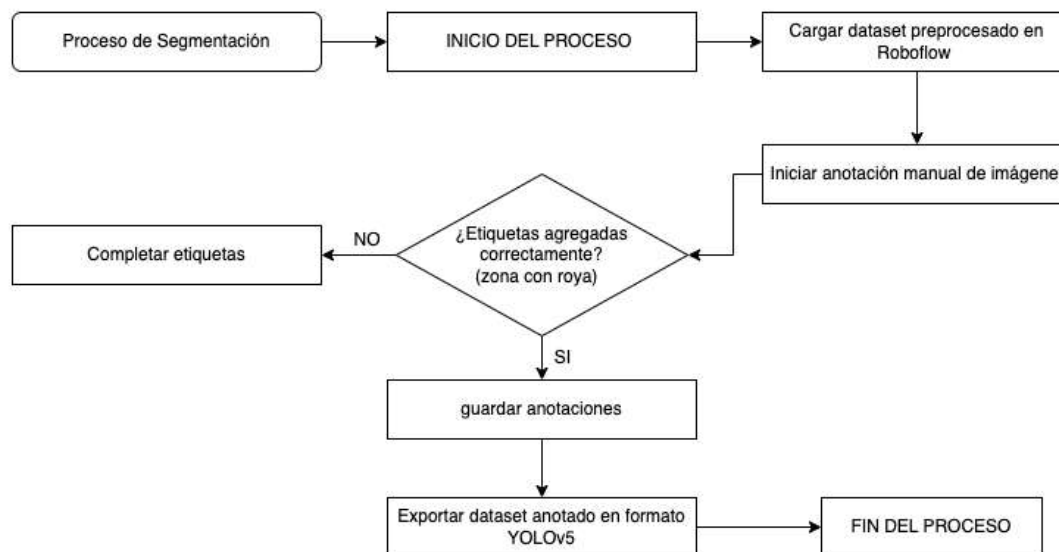
- Auto-Orientación de las imágenes para mantener una alineación coherente,
- Redimensionamiento (Resize) al tamaño 640×640,
- Ajuste automático de contraste mediante ecualización adaptativa (Auto-Adjust Contrast),
- Conversión a escala de grises (Grayscale) para facilitar el procesamiento posterior si se requiere.

Una vez validados los parámetros, las imágenes pasan a la etapa final del diagrama: el preprocesamiento de filtrado, donde ya se visualizan las imágenes con las transformaciones aplicadas. Este bloque representa el conjunto de imágenes optimizadas que conforman el dataset final para el entrenamiento del modelo YOLOv5.

**Segmentación:** una vez finalizado el proceso de preprocesamiento, el conjunto de imágenes fue cargado en dicha plataforma para continuar con la segmentación manual de las regiones de interés. El flujo de este proceso puede observarse en la Figura 45.

**Figura 45.**

*Diagrama de flujo del proceso de segmentación*

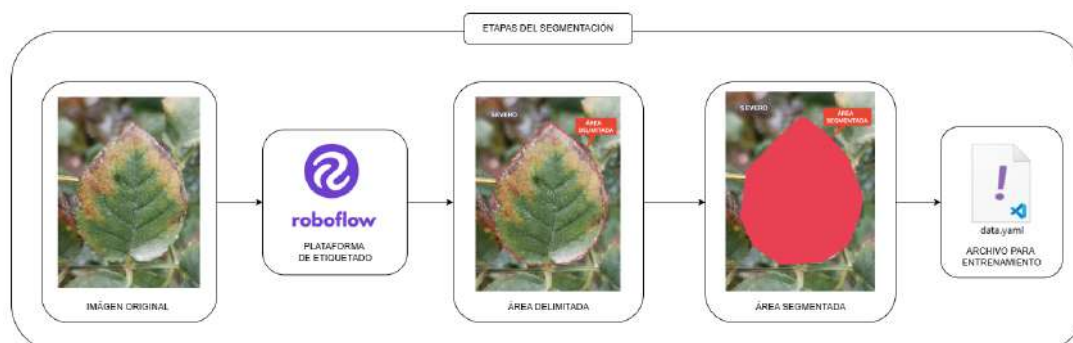


**Nota.** Ilustración propia que muestra cómo se realiza la anotación manual y la exportación del dataset segmentado.

Roboflow permite etiquetar con precisión las zonas afectadas por la roya (*Phragmidium mucronatum*) en las imágenes analizadas. Estas etiquetas (labels) son fundamentales, ya que definen los objetos que el modelo de inteligencia artificial deberá reconocer durante la fase de entrenamiento. En el Anexo 16 se describe de manera detallada y secuencial el procedimiento seguido para realizar este proceso de etiquetado en la plataforma. Asimismo, la figura 46 ilustra de forma más clara el proceso de anotación manual de las imágenes dentro de Roboflow, correspondiente a la etapa de segmentación del dataset.

**Figura 46.**

*Diagrama de flujo del proceso de segmentación*



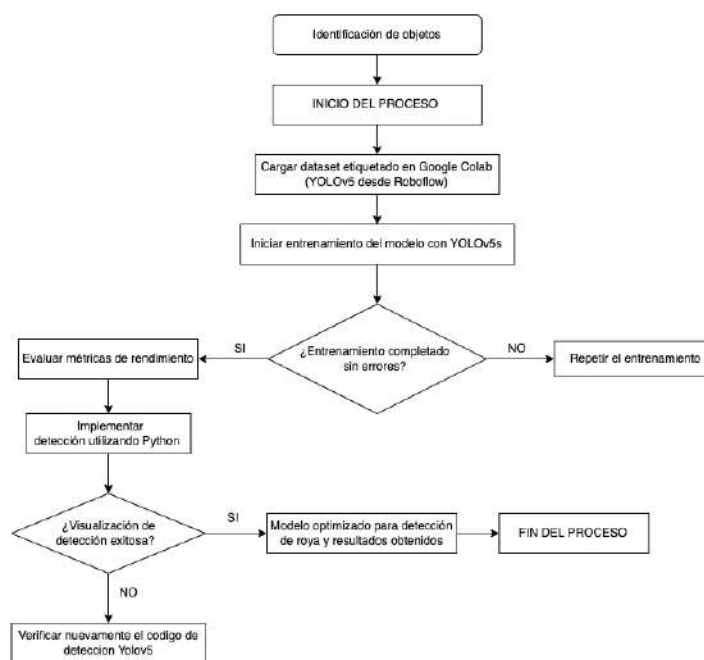
**Nota.** Ilustración propia que muestra cómo se realiza la anotación manual y la exportación del dataset segmentado.

Tras completar la anotación del dataset, Roboflow permite exportar el conjunto de datos segmentado en un formato compatible con YOLOv5s, modelo elegido para el entrenamiento de la red neuronal convolucional. De esta manera, el sistema puede aprender a reconocer visualmente las zonas afectadas por la roya con mayor precisión, lo que fortalece su desempeño en escenarios reales.

**Identificación de objetos:** constituye una fase fundamental dentro del sistema, ya que se encarga de detectar la presencia de roya en las imágenes capturadas por el dron. Esta tarea se realiza mediante el modelo de visión artificial YOLOv5s, previamente entrenado con el conjunto de datos etiquetado. A partir de este modelo, es posible localizar con precisión las zonas afectadas por roya y clasificar visualmente los síntomas identificados. De esta manera, la etapa de identificación de objetos no solo automatiza el análisis visual, sino que también incrementa la rapidez y la consistencia del diagnóstico fitosanitario. La Figura 47 ilustra de forma general este proceso, desde el ingreso de las imágenes al modelo hasta la visualización de las detecciones obtenidas.

**Figura 47.**

*Diagrama de flujo del proceso de identificación de objetos*



**Nota.** Ilustración propia que muestra el flujo de entrenamiento, detección y validación del modelo YOLOv5s.

Elaborado por el autor.

Desde la plataforma Google Colab, aprovechando el soporte de GPU para acelerar el procesamiento y optimizar el rendimiento durante el entrenamiento, se utilizó el dataset previamente preparado en Roboflow, compuesto por imágenes segmentadas y debidamente anotadas. Este conjunto de datos se dividió en subconjuntos de entrenamiento, validación y prueba, lo que permitió a la red neuronal aprender a identificar con precisión las zonas



neuronas son compartidas entre diferentes capas mediante el uso de filtros, lo que dificulta determinar de manera exacta el número de neuronas activas en cada capa oculta. Sin embargo, es posible realizar una estimación aproximada del número de neuronas presentes en dichas capas, para lo cual se aplica la Ecuación 3.1.

$$h = \frac{2}{3} in + out \quad (3.1)$$

donde:

- $h$  = número de neuronas en las capas ocultas,
- $in$  = número de neuronas en la capa de entrada,
- $out$  = número de neuronas en la capa de salida.

A continuación, se sustituyen los valores correspondientes, considerando una entrada de  $in = 1\,228\,800$  neuronas y una salida de  $out = 4$  neuronas (las cuatro clases de severidad):

$$h = \frac{2}{3} \cdot 1\,228\,800 + 4 \quad (3.2)$$

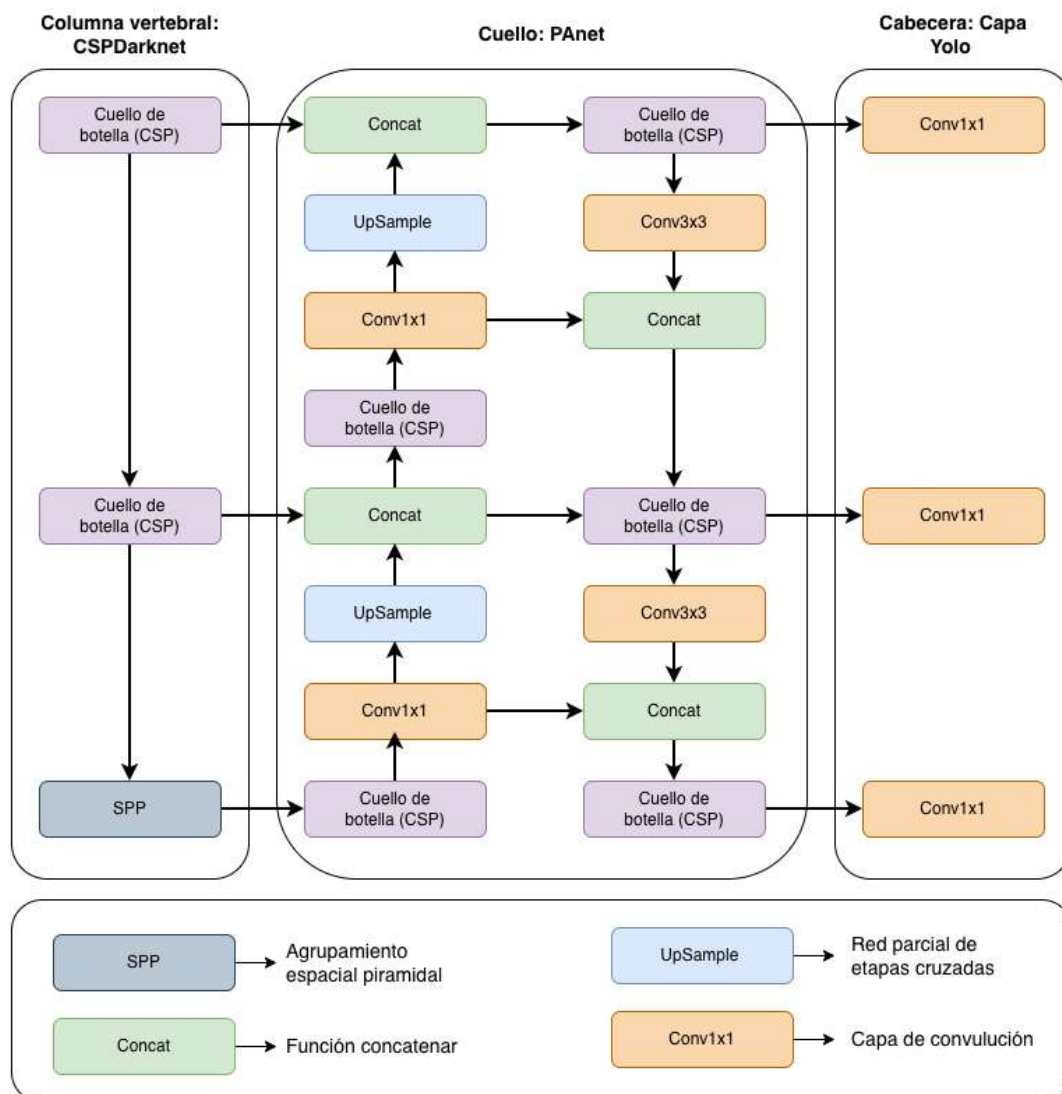
$$= 819\,200 + 4 \quad (3.3)$$

$$= 819\,204 \quad (3.4)$$

La arquitectura de YOLOv5 se estructura en tres bloques principales que trabajan de forma conjunta. En primer lugar, la columna vertebral o Backbone se basa en la arquitectura CSPDarknet53, la cual está compuesta por 29 capas convolucionales de tamaño  $3 \times 3$  encargadas de extraer de manera progresiva las características más importantes de la imagen de entrada. A continuación, el bloque conocido como Neck emplea la arquitectura PANet, cuya función es combinar información proveniente de diferentes escalas, lo que permite mejorar la identificación de objetos de diversos tamaños; para este fin, se utilizan 10 capas. Finalmente, la cabeza de la red o Head corresponde a la capa YOLO, integrada por 3 capas convolucionales responsables de generar las predicciones finales, tanto de las cajas delimitadoras como de las clases detectadas. La organización general de esta arquitectura se presenta en la Figura 49.

Figura 49.

Arquitectura general del modelo YOLOv5



**Nota.** Diagrama de la arquitectura del modelo YOLOv5, donde se identifican la columna vertebral CSPDarknet, el cuello PANet y la capa de detección YOLO. Adaptado de Chowdhury et al. (2023).

Para un mejor entendimiento se explican mejor algunos bloques:

- Cuello de botella CSP (BottleNeckCSP): Estos bloques forman parte de la arquitectura CSPDarknet53 y se encargan de mejorar progresivamente la extracción de características de la imagen. Su diseño permite profundizar el análisis visual sin aumentar de forma excesiva el costo computacional, logrando un balance adecuado entre eficiencia y rendimiento del modelo.
- SPP (Spatial Pyramid Pooling): El bloque de agrupamiento espacial piramidal permite analizar la información visual en diferentes escalas dentro de una misma imagen. Gracias a esto, la red logra una mayor

capacidad para reconocer objetos que varían en tamaño o que se encuentran en distintas posiciones, fortaleciendo su desempeño en escenarios complejos.

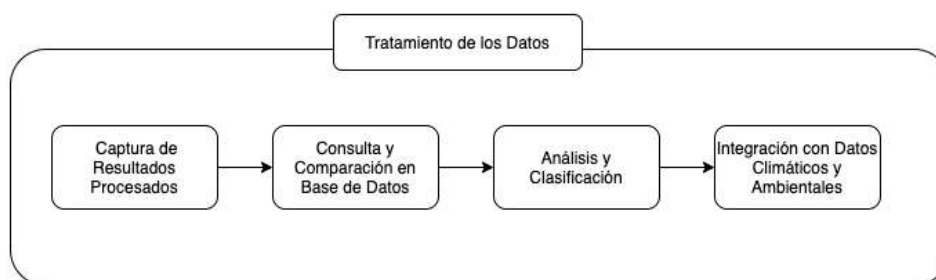
- **UpSample:** Este bloque tiene como función aumentar la resolución de los mapas de características. Al escalar las características a un tamaño mayor, se facilita la integración de información detallada proveniente de capas superficiales con información más abstracta obtenida en capas profundas.
- **Concat:** El bloque de concatenación se utiliza para unir mapas de características generados en diferentes niveles de la red. Esta fusión permite combinar información global del contexto con detalles más precisos, enriqueciendo la representación final utilizada para la detección.
- **Conv 1×1:** Las convoluciones de tamaño 1×1 se emplean principalmente para ajustar la cantidad de canales de los mapas de características, sin alterar su resolución espacial. Este tipo de operación actúa como un mecanismo de optimización que permite reorganizar y reducir la información de manera eficiente.
- **Conv 3×3:** Las capas de convolución con kernels de 3×3 permiten capturar patrones locales presentes en la imagen. Al analizar la relación entre píxeles vecinos, estas capas aportan una mayor capacidad descriptiva al modelo, mejorando su habilidad para identificar estructuras y formas relevantes.

### 3.5.3 Bloque 3: tratamiento de los datos

Para el tercer bloque del tratamiento de datos se establecieron cuatro etapas orientadas a transformar los resultados en información útil para la toma de decisiones. En primer lugar, se capturan y comparan las detecciones con los registros de la base de datos para mantener un historial de monitoreo coherente; posteriormente, la información es analizada y clasificada para determinar el nivel de severidad de la roya. Finalmente, se incorporan variables climáticas y ambientales, como temperatura y humedad, que influyen en la propagación de la enfermedad, tal como se muestra en el diagrama de bloques de la Figura 50.

**Figura 50.**

*Diagrama de flujo del tratamiento de los datos*



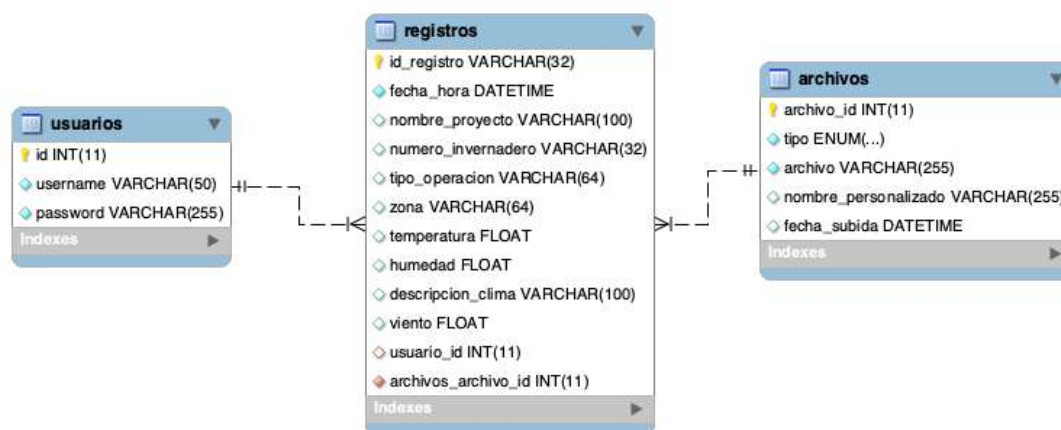
**Nota.** Ilustración propia que muestra cómo se analizan, comparan e integran los datos recolectados para ser tratados.

**Captura de resultados procesados:** Cuando el procesamiento de la información se ha completado, los datos que produce el sistema son administrados a través de una base de datos estructurada en MySQL. Esta fue diseñada para asegurar un almacenamiento eficaz, seguro y ordenado. Esta base de datos está formada sobre todo por tres tablas: archivos, usuarios y registros.

Esta estructura no solo permite almacenar los datos de manera segura, sino que también facilita la clasificación, consulta y visualización ordenada de los resultados en la interfaz del sistema. La Figura 51 muestra la distribución lógica de estas tablas y sus relaciones principales dentro del sistema.

**Figura 51.**

*Modelo entidad-relación para el tratamiento de datos*



**Nota.** Ilustración propia que representa el modelo entidad-relación de la base de datos MySQL, donde se gestionan los usuarios, registros de detección y archivos relacionados, incluyendo variables ambientales.

La tabla de usuarios guarda los datos requeridos para la autenticación y el acceso al sistema, lo que posibilita una gestión segura de los perfiles que utilizan la plataforma. Por otro lado, la tabla de registros contiene la información asociada a cada detección que se ha realizado, como el tipo de operación, la fecha y hora, la ubicación del cultivo y las variables climáticas relacionadas. Esta información posibilita que cada análisis se sitúe en un contexto preciso. Por último, la tabla de archivos juega un rol fundamental en el procesamiento y organización de los datos visuales. Administra de manera organizada todos los videos e imágenes procesados por el sistema, conservando un registro minucioso de cada archivo analizado.

Cada entrada se vincula tanto con un registro específico como con el usuario que ejecutó la operación, garantizando así una trazabilidad completa y segura del flujo de información. Tal como se representa en la Figura 52, los resultados procesados se almacenan de forma organizada en esta estructura, lo que facilita su posterior análisis y uso dentro de la plataforma de detección de roya.

Figura 52.

Consulta de registros en la tabla archivos de la base de datos MySQL

```
mysql>
mysql> USE DBRoya;
Reading table information for completion of table and column names
|You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
mysql> SELECT * FROM archivos;
```

archivo_id	usuario_id	tipo	archivo	nombre_personalizado	fecha_subida
1	1	imagen	det_20250601_011625_IMG_20250523_102845.jpg	FFF	2025-05-31 23:16:31
2	1	imagen	det_20250601_011641_IMG_20250523_102906.jpg		2025-05-31 23:16:43
3	1	imagen	det_20250601_011655_IMG_20250523_102922.jpg	NICE	2025-05-31 23:16:57
4	2	imagen	det_20250601_014911_IMG_20250523_104310.jpg	fffff	2025-05-31 23:49:16
5	2	imagen	det_20250601_014935_IMG_20250523_103809.jpg		2025-05-31 23:49:37
6	1	video		ffff	2025-06-04 15:35:56
7	1	video		jh	2025-06-04 16:48:57

**Nota.** Ilustración propia que muestra los datos almacenados en la tabla archivos, incluyendo tipo de archivo, nombre personalizado, usuario asociado y fecha de subida.

**Consulta y comparación en la base de datos:** una vez almacenados los resultados del procesamiento, el sistema ejecuta la etapa de consulta y comparación en la base de datos mediante sentencias SQL, con el objetivo de contrastar las nuevas detecciones con registros previos. Este proceso permite identificar patrones recurrentes de roya dentro del invernadero.

La organización de la base de datos en tablas relacionales facilita búsquedas eficientes al cruzar variables como fecha, nivel de severidad, ubicación y condiciones climáticas, lo que contribuye a la identificación de tendencias o anomalías en el comportamiento de la roya. La Figura 53 presenta el fragmento de código utilizado para registrar los archivos procesados y asociarlos con su respectivo usuario.

Figura 53.

Fragmento de código para registrar archivos en la base de datos

```
# Guarda, consulta y compara el resultado en BD
nuevo_archivo = Archivo(
    usuario_id=session['user_id'],
    tipo=tipo,
    archivo=output_filename,
    nombre_personalizado=nombre_personalizado
)
db.session.add(nuevo_archivo)
db.session.commit()
```

**Nota.** Ilustración propia que muestra la inserción de un nuevo archivo en la base de datos MySQL desde una aplicación Python.

**Análisis y clasificación:** este proceso se realiza a partir de los resultados generados por el modelo de detección, el cual identifica la presencia de roya. En esta etapa, el sistema extrae de cada imagen o video el valor aso-

ciado al nivel de severidad detectado y lo clasifica de manera automática según criterios previamente definidos, basados en los porcentajes de área afectada obtenidos por el modelo YOLOv5.

Es por ello que, la Tabla 31 presenta la escala utilizada para determinar el grado de severidad de la infección por *Phragmidium mucronatum*. Dicha escala fue construida a partir de observaciones visuales y criterios técnicos de afectación en el follaje, y establece cuatro niveles de clasificación: Correcto, Leve, Severo y Crítico.

**Tabla 29.**

*Escala para evaluar el grado de severidad de Phragmidium mucronatum en cultivos de rosales.*

Grado de Severidad	% de afectación	Síntomas observables en el follaje	Evidencia visual
<b>Correcto (I)</b>	0.0–0.5	Las hojas presentan un aspecto sano, sin presencia de pústulas, manchas amarillas o anaranjadas, ni signos visibles de infección por roya.	
<b>Leve (II)</b>	0.50–0.60	Se observan pequeñas pústulas aisladas de color anaranjado en las hojas.	
<b>Severo (III)</b>	0.60–0.70	Las hojas muestran múltiples lesiones de roya, pústulas densas y avance evidente de la infección.	
<b>Crítico (IV)</b>	Mayor a 0.70	La infección es crítica, las hojas están completamente afectadas, con necrosis, marchitez y pérdida de funcionalidad fisiológica.	

**Nota.** La presente escala fue elaborada para la validación visual del grado de severidad de *Phragmidium mucronatum* en hojas de rosales, como parte del sistema de detección automatizada desarrollado en esta tesis.

También, la Tabla 32 muestra la estructura de la matriz de confusión empleada para validar la clasificación automática del grado de severidad. En esta matriz se comparan las predicciones generadas por el sistema de de-

tección con los valores reales observados, organizando los resultados en cuatro categorías: verdaderos positivos, falsos negativos, falsos positivos y verdaderos negativos. Adicionalmente, cada cuadrante se vincula con una condición visual específica, lo que permite interpretar de forma más clara el desempeño del modelo en la identificación de la severidad de la roya.

**Tabla 30.**

*Matriz de confusión para la clasificación automática del grado de severidad de *Phragmidium mucronatum*.*

	Valoración	Predicciones	
		Positivas	Negativas
Valor Real	Positivos	<b>Verdaderos Positivos</b> Marchito	<b>Falsos Negativos</b> Enfermo
	Negativos	<b>Falsos Positivos</b> Potencialmente enfermos	<b>Verdaderos Negativos</b> Saludable

**Nota.** Se presenta la estructura de la matriz de confusión adaptada a la clasificación de roya en rosales.

Para determinar los niveles de severidad a partir de la matriz de confusión, se deben considerar las siguientes ecuaciones, las cuales permiten calcular los grados de exactitud, precisión, sensibilidad y el tiempo de respuesta del sistema:

**Ecuación de exactitud:** Esta métrica permite evaluar el nivel de acierto del sistema, ya que representa la proporción entre el número de predicciones correctamente realizadas y el total de predicciones efectuadas, como se indica en la Ecuación 3.5.

$$\text{Exactitud} = \frac{\text{Verdaderos positivos} + \text{Verdaderos negativos}}{\text{Total}} \quad (3.5)$$

**Ecuación de precisión:** La precisión permite evaluar qué tan confiables son los elementos identificados dentro de los cuadros delimitadores, indicando si las predicciones realizadas por el sistema son relevantes y correctas. Este parámetro se calcula dividiendo el número de verdaderos positivos entre la suma de verdaderos positivos y falsos positivos, tal como se presenta en la Ecuación 3.6.

$$\text{Precisión} = \frac{\text{Verdadero positivo}}{\text{Verdadero positivo} + \text{Falso positivo}} \quad (3.6)$$

**Ecuación de sensibilidad:** Este indicador se utiliza para analizar la capacidad del sistema para reconocer correctamente los casos positivos, expresando la relación entre los aciertos obtenidos y el total de ejemplos positivos evaluados. Sus valores se encuentran en un rango de 0 a 1, donde los valores más altos representan un mejor desempeño. Este parámetro se obtiene al dividir el número de verdaderos positivos entre la suma de verdaderos positivos y falsos negativos, tal como se muestra en la Ecuación 3.7.

$$\text{Sensibilidad} = \frac{\text{Verdadero positivo}}{\text{Verdadero positivo} + \text{Falso negativo}} \quad (3.7)$$

**Ecuación de tiempo de respuesta:** En aplicaciones que requieren procesamiento en tiempo real, el tiempo de respuesta se emplea como un indicador del desempeño del sistema, ya que mide el lapso comprendido entre el ingreso de los datos y la generación de la imagen de salida con la detección realizada. Este parámetro se calcula según lo descrito en la Ecuación 3.8.

$$t = t_{\text{fin}} - t_{\text{inicio}} \quad (3.8)$$

Posteriormente, este valor de severidad es almacenado en la base de datos junto al archivo procesado, lo que permite no solo organizar la información de forma más eficiente, sino también establecer filtros de búsqueda, generar reportes y ofrecer recomendaciones según el grado de afectación identificado. La Figura 54 ilustra el fragmento de código donde se realiza el análisis y la comparación de resultados, permitiendo clasificar la severidad de la roya en función de la cantidad de detecciones registradas.

#### Figura 54.

*Fragmento de código para el análisis y clasificación de severidad*

```
# Según el número de detecciones, se clasifica el nivel de severidad de la roya en el cultivo.
def clasificar_severidad(cantidad_detecciones):
    # Si se detectan 2 o menos casos de roya, se considera una afectación leve.
    if cantidad_detecciones <= 2:
        return "leve"

    # Si hay entre 3 y 5 detecciones, se considera una afectación grave.
    elif 3 <= cantidad_detecciones <= 5:
        return "grave"

    # Si hay más de 5 detecciones, se clasifica como una afectación moderada.
    else:
        return "crítica"
```

**Nota.** Ilustración propia que muestra la lógica de análisis para clasificar el nivel de severidad de la roya según la cantidad de detecciones.

Como etapa final del procesamiento, el sistema integra datos ambientales obtenidos en tiempo real desde la API OpenWeatherMap, incluyendo temperatura, humedad y condiciones climáticas, los cuales se almacenan junto con los resultados de detección. La integración de esta API se presenta en la Figura 55.

**Figura 55.**

*Fragmento de código para la obtención de datos climáticos en tiempo real*

```
try:
    # API key para OpenWeatherMap (guárdala segura!)
    api_key = "0bd18e6005c0ad9212375feb2ce273e"

    # URL para consulta con parámetros:
    # ciudad, unidades métricas, idioma español, y API key
    url = f"https://api.openweathermap.org/data/2.5/weather?q={ciudad}&units=metric&appid={api_key}&lang=es"

    # Realiza la petición GET
    response = requests.get(url)

    # Si la respuesta es exitosa, extrae los datos
    if response.status_code == 200:
        data = response.json()
        return {
            "temperatura": data["main"]["temp"], # Temperatura actual (°C)
            "humedad": data["main"]["humidity"], # Humedad relativa (%)
            "descripcion": data["weather"][0]["description"], # Descripción textual (e.g. "cielo claro")
            "viento": data["wind"]["speed"] # Velocidad del viento (m/s)
        }
    else:
        # En caso de respuesta inválida, retorna None
        return None

except Exception as e:
    # Captura cualquier excepción y la imprime para debugging
    print(f"❌ Error al obtener clima: {e}")
    return None
```

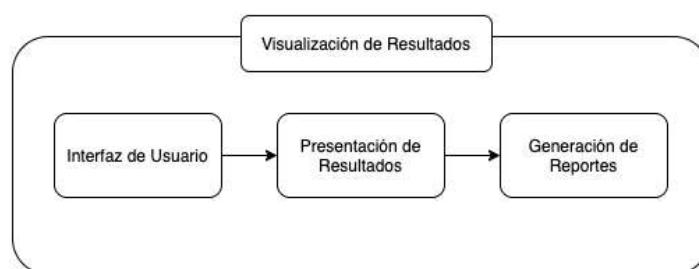
**Nota.** Ilustración propia que muestra la integración de la API de OpenWeatherMap para consultar condiciones climáticas del lugar donde se realizará la detección.

### 3.5.4 Bloque 4: visualización de resultados

Finalmente, el bloque de visualización de resultados se estructura en tres etapas: la interfaz de usuario desarrollada en Flask y Python para la presentación visual de las detecciones mediante análisis de archivos, y la generación de reportes. Estas etapas se ilustran en el diagrama de bloques de la Figura 56.

**Figura 56.**

*Diagrama de bloques de la visualización de resultados*

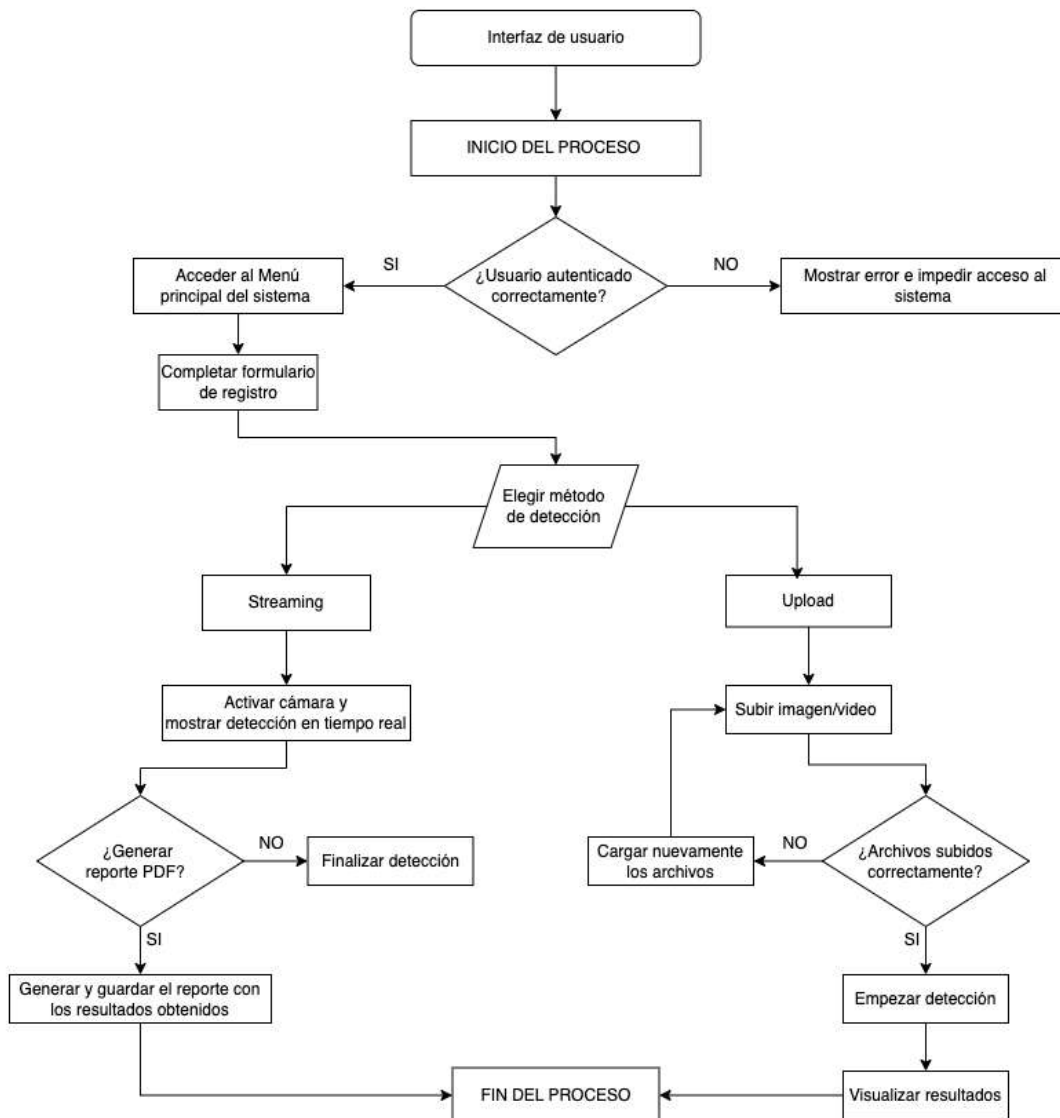


**Nota.** Ilustración propia que muestra el flujo desde la interfaz de usuario hasta la presentación de resultados y generación de reportes.

**Interfaz de usuario:** la interfaz fue desarrollada con Flask en Python, integrando el modelo de detección YOLOv5 en una aplicación web basada en HTML y CSS, con acceso seguro mediante login. El sistema se organiza en tres secciones principales: panel de registros, transmisión en vivo para detección en tiempo real y carga de archivos para el análisis de imágenes o videos. El flujo general de funcionamiento de la interfaz se muestra en la Figura 57.

**Figura 57.**

*Diagrama de flujo de la interfaz de usuario*



**Nota.** Ilustración propia que muestra el proceso realizado por la interfaz del usuario.

La Figura 58 presenta de manera clara la interfaz principal del sistema inteligente para la detección automatizada de roya, la cual ha sido diseñada para facilitar el registro de nuevos eventos de monitoreo y capturar los datos necesarios antes del análisis.

Figura 58.

Interfaz de usuario

**Nota.** Interfaz de registro del sistema inteligente de la detección de Roya. Elaborado por el autor.

El sistema incluye un formulario de registro intuitivo, en el que el usuario tiene la posibilidad de introducir datos fundamentales para cada sesión de monitoreo. Estos datos comprenden el nombre del proyecto, la fecha y hora del análisis, el ID del registro, la zona supervisada, el número del invernadero y la categoría de operación. Esta organización facilita la detección eficaz de cada elemento.

A la derecha del formulario, hay un panel que presenta información climática en tiempo real, incluyendo la humedad, la temperatura, la velocidad del viento y una breve reseña de las condiciones climáticas. Para contextualizar las condiciones ambientales del cultivo y ajustar la evaluación de la severidad de la roya, es fundamental esta información que se incorpora automáticamente al sistema.

Los botones de acción se encuentran en la parte inferior de la interfaz, y desde allí el usuario tiene la posibilidad de comenzar una detección en vivo, generar una nueva sesión de monitoreo o subir archivos para su análisis posterior. La interfaz completa ha sido diseñada para proporcionar una experiencia que sea fluida, clara y se ajuste al funcionamiento operativo del sistema inteligente.

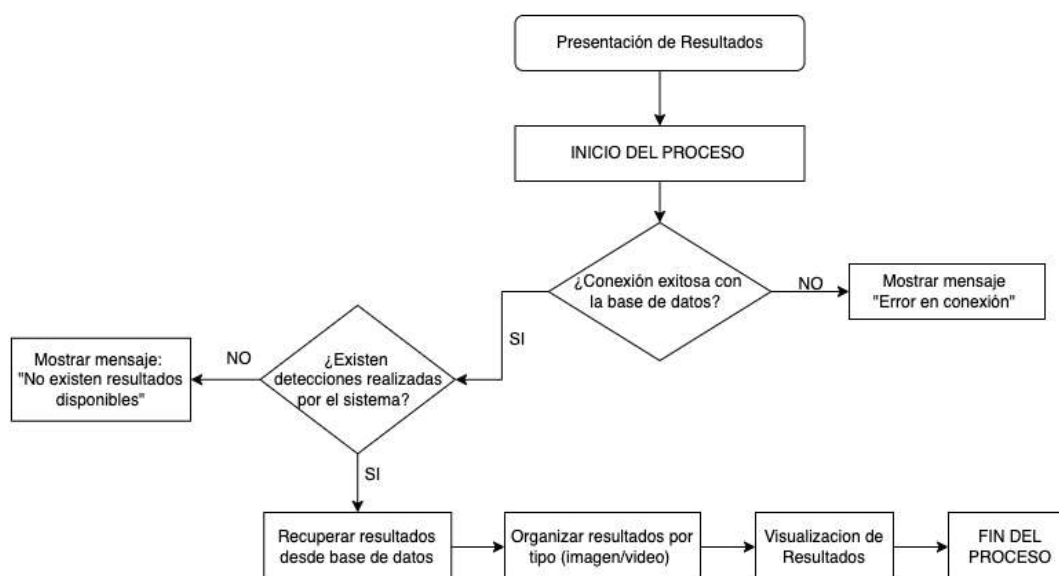
**Presentación de resultados:** en esta sección se visualizan las imágenes y videos procesados por el sistema inteligente de detección de roya, cada uno correspondiente a una detección realizada en la que se resaltan las zonas afectadas de manera precisa mediante el modelo entrenado. Esta visualización permite al usuario evaluar gráficamente el estado del cultivo y analizar cada caso de forma individual. Para una mejor comprensión, el sistema organiza los archivos por tipo (imagen o video), fecha de procesamiento y nivel de severidad, lo que facilita

el seguimiento histórico de las detecciones y la trazabilidad del análisis.

El proceso inicia con la validación de la conexión con la base de datos y la verificación de la existencia de resultados disponibles. Luego, los datos se gestionan y clasifican según su tipo antes de ser presentados al usuario para su revisión visual. Este flujo operativo se resume en la Figura 59, donde se muestra de manera esquemática cómo se procesan, organizan y despliegan los resultados generados por el sistema.

**Figura 59.**

*Diagrama de flujo de la presentación de resultados*



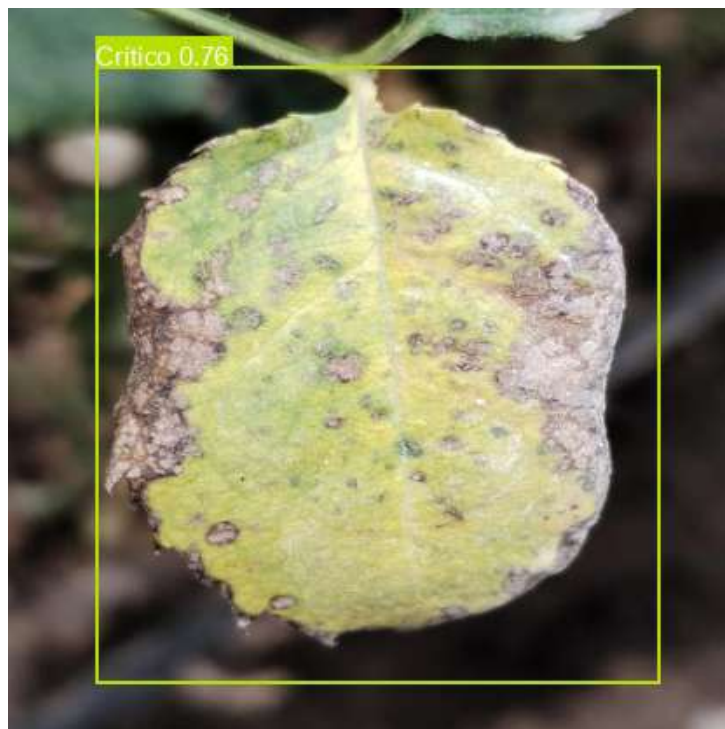
**Nota.** Proceso de consulta y visualización de resultados. Elaborado por el autor.

Una vez que el sistema concluye el análisis de las imágenes, se identifica de manera clara la presencia de roya en las hojas evaluadas. Cada imagen es procesada por el modelo YOLOv5, el cual permite detectar y resaltar visualmente las áreas afectadas por la enfermedad. Estas zonas se representan mediante cuadros delimitadores y etiquetas que indican el nivel de severidad identificado, facilitando la comprensión del grado de afectación observado en el cultivo.

Esta representación gráfica permite al usuario interpretar de forma rápida y sencilla la magnitud del problema, apoyando una evaluación más precisa del estado fitosanitario. En la Figura 59 se observa un ejemplo de detección correspondiente a un nivel crítico de severidad, donde se evidencian claramente las áreas afectadas por roya.

**Figura 60.**

*Resultados de detección en nivel Crítico*



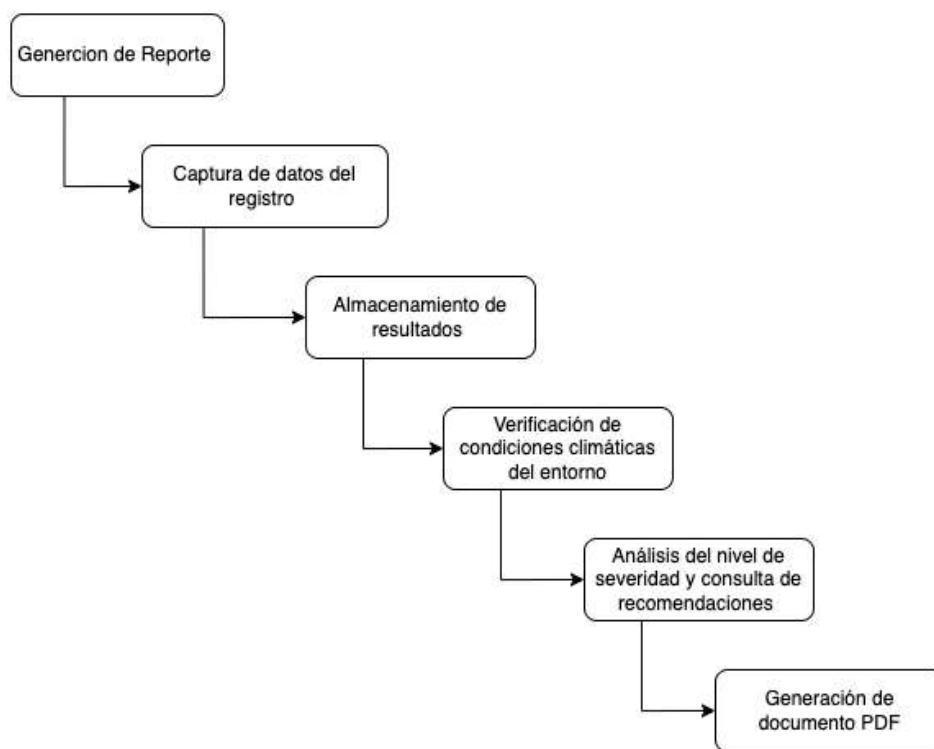
**Nota.** Visualización de resultados con presencia de roya. Elaborado por el autor.

**Generación de reportes:** la generación de reportes es una funcionalidad esencial dentro del sistema inteligente, ya que permite compilar y presentar de manera estructurada la información obtenida en cada sesión de detección. Cada documento incluye los datos del registro inicial fecha, hora, ubicación y condiciones climáticas junto con los resultados del modelo, destacando el nivel de severidad identificado y la evidencia visual correspondiente.

Según el grado de afectación, el sistema incorpora recomendaciones técnicas, como el uso de fungicidas o prácticas agronómicas para controlar la roya. Los reportes se exportan en formato PDF, facilitando su almacenamiento y consulta posterior. La Figura 61 muestra el flujo del proceso, desde la captura de datos hasta la generación del documento final.

**Figura 61.**

*Diagrama de bloques de la generación de reportes*



**Nota.** Ilustración propia que muestra el proceso de generación de reportes, desde la captura de datos hasta la creación del documento PDF.

En este proceso, se recogen los datos que ya están en la base de datos. Estos incluyen el ID del registro, la fecha y hora de detección, el área monitorizada, las condiciones climáticas del entorno (como humedad, temperatura, velocidad del viento y descripción del clima), además de las métricas que el sistema ha calculado (como los frames procesados, las detecciones y el promedio por cuadro). Asimismo, se presenta un resumen de registros previos para proporcionar una perspectiva comparativa y se estima una tendencia que muestra cómo ha cambiado la gravedad con el tiempo.

La Figura 62 muestra el fragmento de código encargado de la generación automática de reportes en formato PDF, a partir de los resultados obtenidos por el sistema de detección de roya. Esta funcionalidad ha sido implementada utilizando Flask junto con la librería xhtml2pdf, lo que permite transformar una plantilla HTML predefinida en un documento PDF dinámico y personalizado por sesión de análisis.

Figura 62.

Fragmento de código para la generación de reportes en PDF

```
html = render_template("plantilla_pdf.html",
                      id=datos.get("id"),
                      fecha_hora=datos.get("fecha_hora"),
                      zona=datos.get("zona"),
                      tipo_operacion=datos.get("tipo_operacion"),
                      nombre_proyecto=datos.get("nombre_proyecto"),
                      numero_invernadero=datos.get("numero_invernadero"),
                      temperatura=float(datos.get("temperatura", 0)),
                      humedad=float(datos.get("humedad", 0)),
                      descripcion_clima=datos.get("descripcion_clima"),
                      viento=datos.get("viento"),
                      estado=datos.get("estado"),
                      avg=datos.get("avg"),
                      frames=datos.get("frames"),
                      detecciones=datos.get("detecciones"),
                      ultimos_registros=ultimos_registros,
                      tendencia=tendencia)

result = BytesIO()
pisa.CreatePDF(html, dest=result)
response = Response(result.getvalue(), content_type='application/pdf')
response.headers['Content-Disposition'] = f"inline; filename=reporte_{datos.get('id')}.pdf"
return response
```

*Nota.* Ilustración propia que muestra el proceso de generación de un reporte en PDF a partir de datos obtenidos, utilizando una plantilla HTML.

El reporte final es generado de forma automática y devuelto al usuario para su visualización o descarga, permitiendo conservar un respaldo formal y estructurado de cada monitoreo realizado. Esta herramienta aporta valor al proceso de gestión agrícola, facilitando la toma de decisiones basadas en evidencia técnica.

## CAPÍTULO IV

### RESULTADOS Y ANÁLISIS

En este capítulo se presentan los resultados del sistema desarrollado para la detección de roya (*Phragmidium mucronatum*) en rosales usando visión artificial y redes neuronales convolucionales. El modelo se entrenó y probó con imágenes tomadas en dos escenarios diferentes: imágenes aéreas capturadas con un dron, que proporcionan una vista general del cultivo, e imágenes terrestres, para capturar detalles de las hojas enfermas. A partir de estas pruebas se mide el rendimiento del sistema con un análisis cualitativo y cuantitativo para establecer su exactitud y eficacia en el reconocimiento de la enfermedad.

#### 4.1 Resultados del modelo de detección

Los resultados obtenidos del proceso de entrenamiento de la red neuronal convolucional se presentan en la Figura 63. En esta etapa se generan los archivos necesarios para la validación y funcionamiento del modelo, entre los cuales se incluye el archivo data.yml, que define la estructura y configuración del conjunto de datos utilizado. Posteriormente, al ejecutar el entrenamiento en la plataforma Google Colab, se obtiene el archivo yolov5s.pt, correspondiente al modelo entrenado. Este modelo se encuentra preparado para su posterior integración en el sistema de detección de roya, permitiendo su aplicación en la identificación automática de la enfermedad en los cultivos. Estos resultados se obtienen de planificación, entrenamiento y validación que se detalla en los Anexos 15 al Anexo 17.

**Figura 63.**

*Resultados de los archivos generados del modelo de detección*



**Nota.** Archivos del modelo entrenado. Elaborado por el autor

## 4.2 Casos de pruebas

En esta sección se analizan y explican los casos de prueba establecidos para comprobar el funcionamiento del sistema de detección de roya. Se verifica la captura y procesamiento de imágenes, la identificación y categorización de la enfermedad, la elaboración de reportes y la exactitud del modelo, garantizando que cumpla con los requisitos establecidos para demostrar su eficacia y rendimiento.

**Tabla 31.**

Validación de hardware del dron

Hardware	¿Prueba de despliegue?	Sí / No
<b>Descripción:</b>		
Verificar que el dron cuenta con los componentes de hardware necesarios para la captura de imágenes y geolocalización durante su operación.		
<b>Prerrequisitos:</b>		
<ol style="list-style-type: none"> <li>1. Batería del dron con nivel de carga adecuado para la prueba.</li> <li>2. Sistema GPS operativo y correctamente configurado.</li> <li>3. Dron con cámara en correcto funcionamiento.</li> </ol>		
<b>Pasos:</b>		
<ul style="list-style-type: none"> <li>• Comprobar la autonomía de batería del dron.</li> <li>• Encender el dron y comprobar el correcto funcionamiento de la cámara.</li> <li>• Verificar la captura adecuada de imágenes y la geolocalización del dron.</li> </ul>		
<b>Resultados esperados:</b>		
Verificar el correcto funcionamiento de la cámara para la captura de imágenes, así como el estado operativo del sistema de geolocalización (GPS) y de la batería durante la operación.		

*Nota.* Elaborado por el autor.

### Resultados obtenidos:

Para garantizar la autonomía del dron al tomar imágenes, se verificó las características de la batería del DJI Air 3. Este dron está equipado con una batería inteligente de ion-litio, para un uso eficiente de energía y un vuelo estable. La batería tiene una capacidad de alrededor de 4241 mAh, 62.6 Wh y 14.76 V, lo que permite vuelos largos y continuos en condiciones normales de trabajo.

Gracias a estas especificaciones, el dron puede tener una fuente de energía continua y así poder tomar imágenes del cultivo sin interrupciones y tener un vuelo seguro y estable durante toda la detección, la Figura 64 presenta las características de la batería del dron.

**Figura 64.**

*Validación de las características de la batería del dron DJI Air 3*



*Nota.* Elaborado por el autor.

Además, se verificó el estado de la batería, midiendo directamente el voltaje con un multímetro, ajustado en la escala de voltaje continuo (DC). Para esta prueba se conectaron las puntas del multímetro al pin negativo y al pin de salida de voltaje de la batería para verificar que el valor obtenido se mantuviera alrededor de los 14 V, que es su rango normal de funcionamiento. Como se puede observar en la Figura 65, la medida dio 13.89 V, lo que indica que la batería está en buen estado y con carga suficiente para realizar la de toma de imágenes sin problemas.

**Figura 65.**

*Verificación del voltaje de la batería*



*Nota.* Elaborado por el autor.

Para estimar el tiempo de uso de la batería del dron DJI Air 3, se consideraron las especificaciones técnicas proporcionadas por el fabricante y las condiciones reales de operación. De acuerdo con la información técnica disponible, el DJI Air 3 posee un tiempo máximo teórico de vuelo de aproximadamente 46 minutos, valor que corresponde a condiciones ideales de funcionamiento, tales como ausencia de viento, maniobras mínimas y sin carga adicional.

Sin embargo, en escenarios reales de operación, como el monitoreo de cultivos de rosas, la captura de imágenes y la transmisión continua de video influyen directamente en la reducción del tiempo de vuelo del dron. Por esta razón, se considera que el tiempo real de uso de la batería corresponde aproximadamente al 70 % – 80 % del tiempo máximo especificado por el fabricante.

Con base en lo anterior, el tiempo real de uso de la batería se lo realizó mediante la siguiente ecuación:

$$T_{real} = T_{mx} \times \eta \quad (4.1)$$

donde  $T_{real}$  representa el tiempo real de uso de la batería,  $T_{mx}$  corresponde al tiempo máximo de vuelo indicado por el fabricante y  $\eta$  es el factor de eficiencia, cuyo valor se encuentra entre 0.7 y 0.8.

Aplicando esta expresión al dron DJI Air 3, y considerando un factor de eficiencia del 75 %, se obtiene:

$$T_{real} = 46 \text{ min} \times 0,75 = 34,5 \text{ min} \quad (4.2)$$

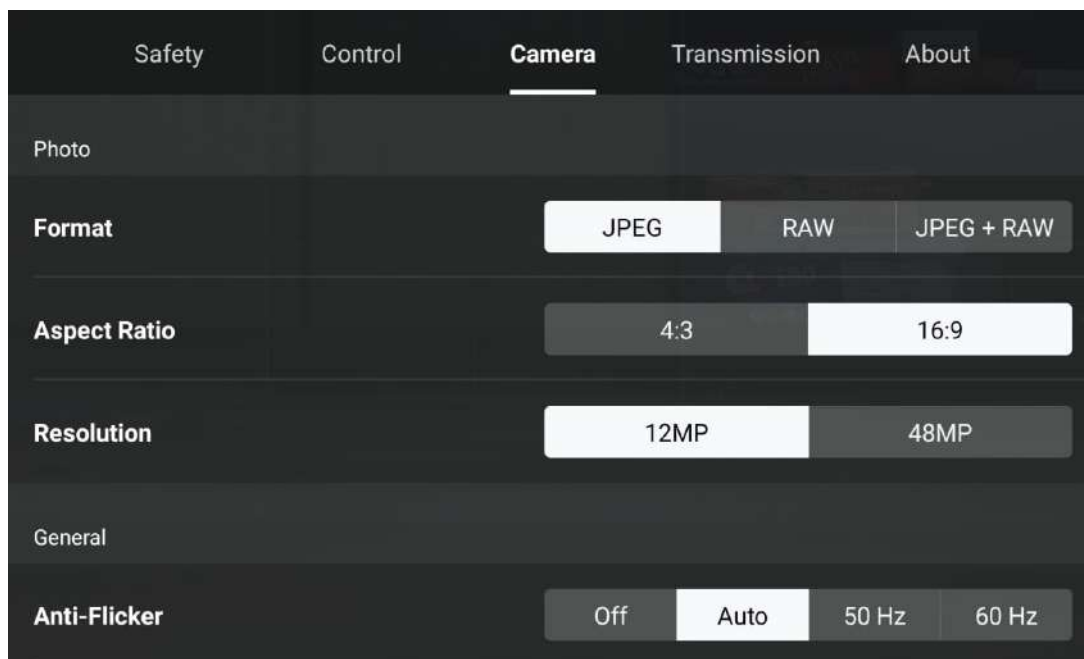
Por lo tanto, el tiempo real estimado de vuelo del dron es de aproximadamente 34 a 36 minutos por batería, valor considerado adecuado para monitorear el cultivo y realizar la captura de imágenes.

A partir de este tiempo de operación, se validan las configuraciones de la cámara del dron DJI Air 3. En esta parte se observan las configuraciones utilizadas en la captura de imágenes. Se definió el formato JPEG porque facilita una gestión más rápida de los archivos y su procesamiento posterior en la plataforma de detección. Se seleccionó la relación de aspecto 16:9 para un campo de visión amplio, adecuado para la toma aérea del cultivo.

También se estableció una resolución de 12MP, que asegura un balance entre calidad de imagen y tamaño del archivo, proporcionando un nivel de detalle apropiado para detectar la enfermedad sin comprometer el funcionamiento del sistema. Finalmente, Anti-Flicker se dejó en automático, permitiendo que la cámara se ajuste a los cambios de luz y así evitar artefactos en las fotografías. La Figura 66 evidencia estas configuraciones.

**Figura 66.**

*Validación de los parámetros de la cámara del dron DJI Air 3*



*Nota.* Elaborado por el autor.

Al verificar que los ajustes de la cámara, comprobaremos que el dron esté encendido para confirmar que todo funciona correctamente. Durante esta prueba se logro verificar que la cámara del dron es capaz de tomar las imágenes del cultivo de forma correcta como se lo planteo en la configuración inicial. Esto garantiza la obtención de imágenes adecuadas para el primer bloque de captura de datos, como se muestra en la Figura 67.

**Figura 67.**

*Validación del funcionamiento de la cámara del dron*

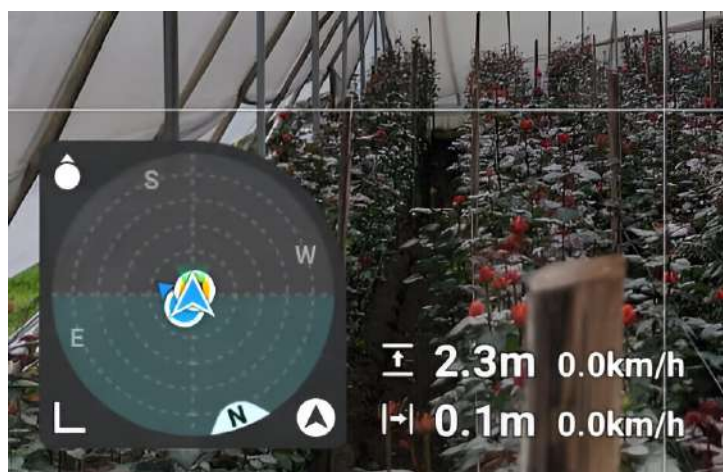


*Nota.* Elaborado por el autor.

Para validar el correcto funcionamiento del sistema GPS se verifica que el mapa de ubicación permita visualizar en tiempo real la posición del dron, para confirmar que el sistema GPS del dron funciona correctamente. El dron registra con precisión sus coordenadas geográficas, lo que posibilita saber su ubicación exacta en todo momento. Esto se pudo verificar mediante este mapa. Así, el dron garantiza que la captura de datos sea confiable, como se ilustra en la Figura 68.

**Figura 68.**

*Validación del funcionamiento de geolocalización del GPS*



*Nota.* Elaborado por el autor.

**Tabla 32.**

Validación en la ejecución del vuelo y captura de imágenes

Hardware	¿Prueba de despliegue?	Sí / No
<b>Descripción:</b>		
Validar la correcta ejecución del vuelo del dron y la captura continua de imágenes del cultivo durante el recorrido definido.		
<b>Prerrequisitos:</b>		
<ol style="list-style-type: none"> <li>1. Mapa de ruta de vuelo previamente definido.</li> <li>2. Cámara del dron calibrada y en correcto funcionamiento.</li> <li>3. Almacenamiento disponible para el guardado de las imágenes capturadas.</li> </ol>		
<b>Pasos:</b>		
<ul style="list-style-type: none"> <li>• Iniciar el vuelo del dron siguiendo la ruta establecida.</li> <li>• Capturar imágenes del cultivo en cada waypoint definido.</li> <li>• Almacenar las imágenes capturadas de manera local en el dron durante la ejecución del vuelo.</li> </ul>		
<b>Resultados esperados:</b>		
El dron ejecuta el vuelo de forma estable y captura imágenes de manera continua en cada waypoint, almacenándolas correctamente para su posterior procesamiento.		

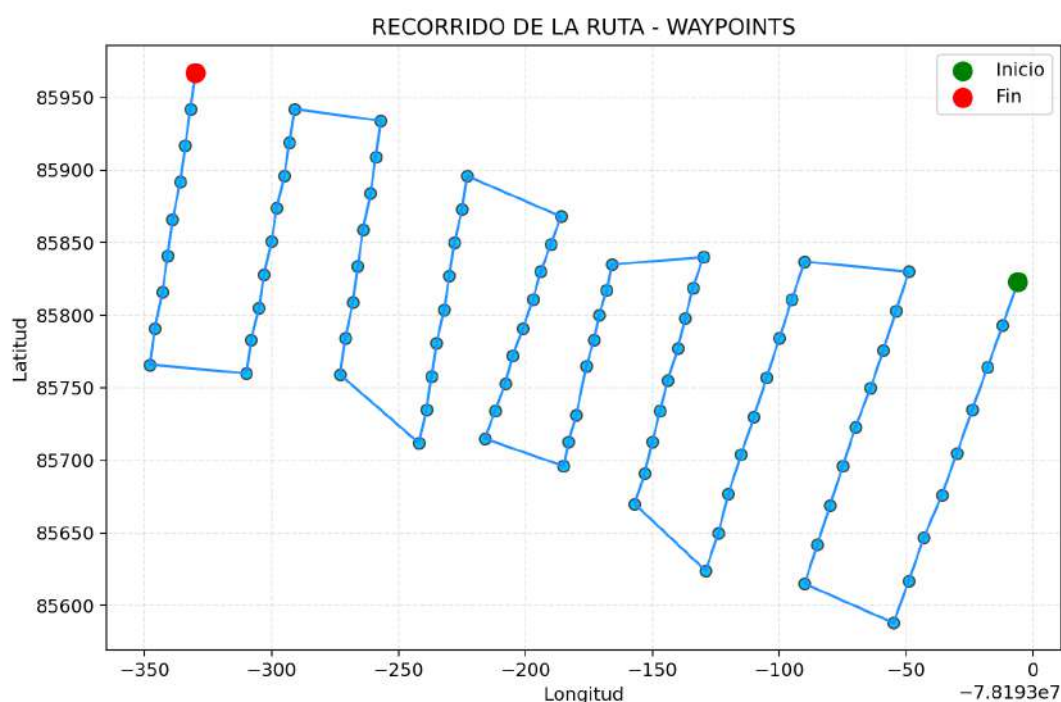
*Nota.* Elaborado por el autor.

### Resultados obtenidos:

La verificación de la ejecución del vuelo se basó en la planificación de la ruta, la cual se visualiza con el mapa establecido por los waypoints. Esta planificación garantiza el control de la trayectoria del dron en el invernadero y que sea la adecuada para la toma de datos en cada punto. La ruta consta de 86 waypoints que cubren la totalidad del área de interés, asegurando una toma de información sistemática en cada punto del cultivo, como se muestra en la Figura 69.

**Figura 69.**

*Validación del recorrido del dron y captura de imágenes en cada waypoint*

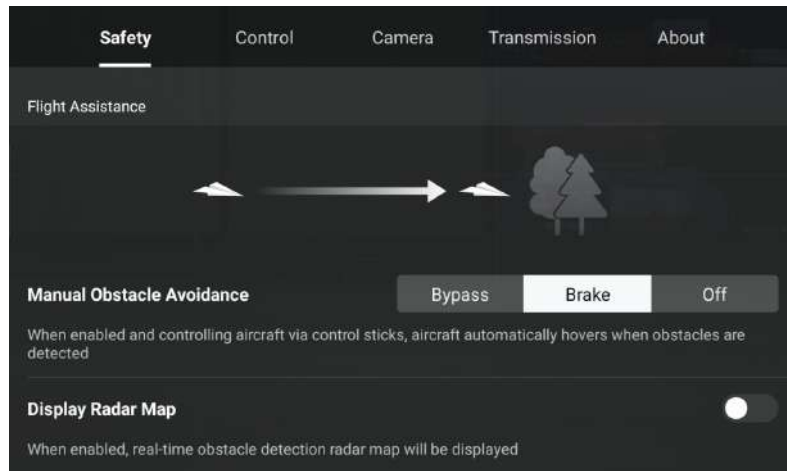


**Nota.** Elaborado por el autor.

Antes de iniciar la ejecución del vuelo se debe verificar que las configuraciones del dron sean las adecuadas con el fin de garantizar una ruta de vuelo segura dentro del invernadero. Este proceso permite minimizar el riesgo accidentes con objetos y obstáculos presentes en el cultivo que puedan interrumpir la captura de datos o comprometer la integridad del equipo. Para ello se configura el dron en modo Brake, con el fin de que los sensores de detección de obstáculos del DJI Air 3, los cuales permiten frenar automáticamente y evitar posibles daños durante el vuelo. La Figura 70 evidencia el proceso de validación de estas configuraciones.

**Figura 70.**

*Verificación de la configuración de seguridad del dron antes del vuelo*

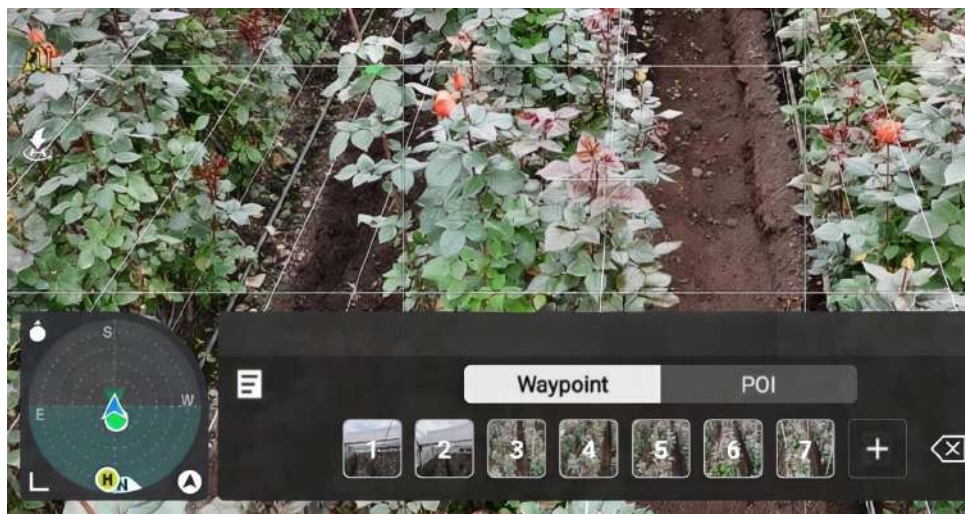


*Nota.* Elaborado por el autor.

Una vez que se ha comprobado el mapa de la ruta y que la configuración del vuelo es adecuada se da inicio a la ejecución del vuelo del dron. Este continúa el trayecto y toma las imágenes en cada uno de los puntos de ruta establecidos. Este procedimiento garantiza que la recolección de datos en cada uno de los puntos se realice adecuadamente. El vuelo y la captura de imágenes permiten garantizar que la información se lo realice de forma completa sobre el cultivo, para utilizarla más adelante en el proceso de detección y análisis del cultivo. La Figura 71 ilustra este proceso.

**Figura 71.**

*Validación en la ejecución del vuelo y captura de imágenes en los waypoints establecidos*

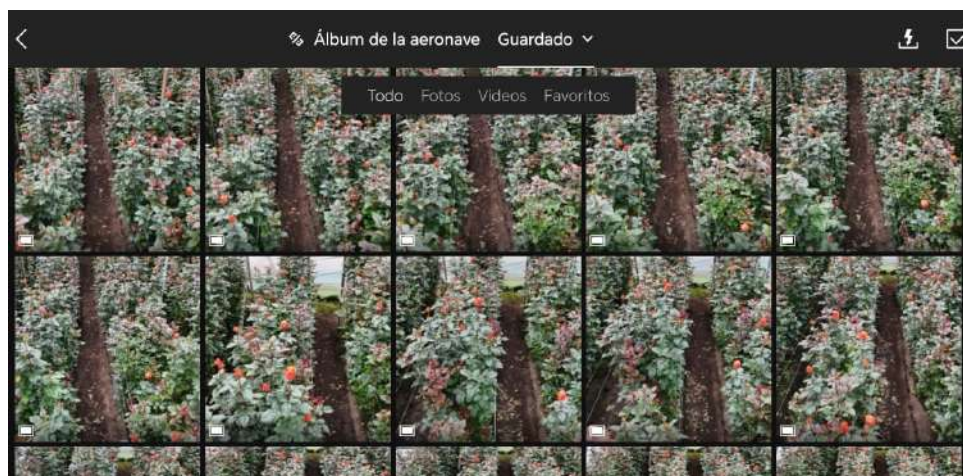


*Nota.* Elaborado por el autor.

Al finalizar la captura de datos se visualiza en la galería de la aeronave todas las imágenes obtenidas del cultivo que corresponden a cada uno de los waypoints, lo que permite verificar que la captura de imágenes se realizó exitosamente, tal como se muestra en la Figura 72.

**Figura 72.**

*Validación en la visualización de las imágenes capturadas en la galería del dron*



*Nota.* Elaborado por el autor.

**Tabla 33.**

Validación del modelo Yolov5

Software	¿Prueba de despliegue?	Sí / No
<b>Descripción:</b>		
Verificar que el sistema procesa automáticamente las imágenes capturadas y detecta la presencia de roya, clasificando los resultados según el nivel de severidad mediante el modelo YOLOv5.		
<b>Prerrequisitos:</b>		
<ol style="list-style-type: none"> <li>1. Modelo YOLOv5 previamente entrenado y cargado en el sistema.</li> <li>2. Aplicación web operativa y accesible.</li> <li>3. Conjunto de imágenes disponibles para su procesamiento.</li> </ol>		
<b>Pasos:</b>		
<ul style="list-style-type: none"> <li>● Cargar el conjunto de imágenes en el sistema mediante la aplicación web.</li> <li>● Ejecutar el procesamiento automático de las imágenes utilizando el modelo YOLOv5.</li> <li>● Clasificar las detecciones obtenidas según el nivel de severidad de la roya.</li> <li>● Registrar los resultados generados en la base de datos del sistema.</li> <li>● Visualizar los resultados de detección en la interfaz de la aplicación web.</li> </ul>		
<b>Resultados esperados:</b>		
El sistema procesa automáticamente las imágenes, detecta correctamente la presencia de roya, clasifica el nivel de severidad y presenta los resultados de forma visual en la aplicación web, almacenando la información generada.		

*Nota.* Elaborado por el autor.

### Resultados obtenidos:

Para verificar el funcionamiento del modelo de detección YOLOv5, se realizó la ejecución del aplicativo web a través del archivo principal app.py. Como se puede ver en la Figura 73, el sistema demostró que la ejecución del modelo fue exitosa.

### Figura 73.

*Verificación de la ejecución del modelo YOLOv5*

```

/home/ubuntu/Escritorio/PythonFinal2026/Python/Roya/lib/python3.8/site-packages/torch/cuda/
warnings.warn("Can't initialize NVML")
Cantidad de GPUs: 0
Cargando modelo YOLOv5...
Using cache found in /home/ubuntu/.cache/torch/hub/ultralytics_yolov5_master
YOLOv5 2025-6-3 Python-3.8.20 torch-2.4.1+cu121 CPU

Fusing layers...
Model summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs
Adding AutoShape...

```

*Nota.* Elaborado por el autor.

La plataforma muestra que las imágenes han sido recolectadas y procesadas adecuadamente por el sistema de detección para verificar su procesamiento en el sistema. Como resultado, el sistema presenta 86 imágenes en total correspondientes a cada waypoint, las que han sido mapeadas correctamente para asegurar una gestión y organización apropiadas de los datos, como se muestra en la Figura 74.

### Figura 74.

*Verificación del procesamiento de imágenes en el sistema*



*Nota.* Elaborado por el autor

Para clasificar las detecciones según su nivel de severidad, el sistema presenta la tabla de waypoints en la que se evidencia los resultados realizados tras el análisis de las imágenes capturadas en cada uno de los puntos estratégicos a lo largo del recorrido. Esta tabla permite la identificación del grado de afectación observado en cada waypoint, lo que ayuda a determinar con exactitud las áreas del cultivo donde hay presencia de roya. Así el sistema ordena los datos obtenidos en la detección lo que favorece a un análisis más exhaustivo y eficaz como se muestra en la Figura 75.

**Figura 75.**

*Clasificación de detecciones por nivel de severidad por waypoints*

**Tabla Waypoints**

ID	LAT	LON	Severidad
19	85837.000000	-78193090.000000	Correcto (I)
20	85811.000000	-78193095.000000	Correcto (I)
21	85784.000000	-78193100.000000	Correcto (I)
22	85757.000000	-78193105.000000	Correcto (I)
23	85730.000000	-78193110.000000	Correcto (I)
24	85704.000000	-78193115.000000	Correcto (I)
25	85677.000000	-78193120.000000	Severo (III)
26	85650.000000	-78193124.000000	Correcto (I)
27	85624.000000	-78193129.000000	Leve (II)
28	85679.000000	-78193157.000000	Severo (III)
29	85691.000000	-78193153.000000	Severo (III)
30	85713.000000	-78193150.000000	Correcto (I)
31	85734.000000	-78193147.000000	

*Nota.* Elaborado por el autor.

Una vez que el sistema termina la detección, los resultados se guardan automáticamente en la base de datos MySQL. Esta base de datos es una parte esencial del sistema. La base de datos contiene la tabla usuarios que permite el ingreso a usuarios registrados en la plataforma, la tabla registros que crea y estructura nuevos procesos de detección y la tabla archivos que almacena la información de cada detección. En la figura 76 se puede observar cómo esta arquitectura permite organizar, controlar y consultar los datos generados para dar seguimiento a cada detección.

Figura 76.

Verificación del almacenamiento de datos en la base de datos MySQL

REGISTROS ALMACENADOS EN LA BASE DE DATOS

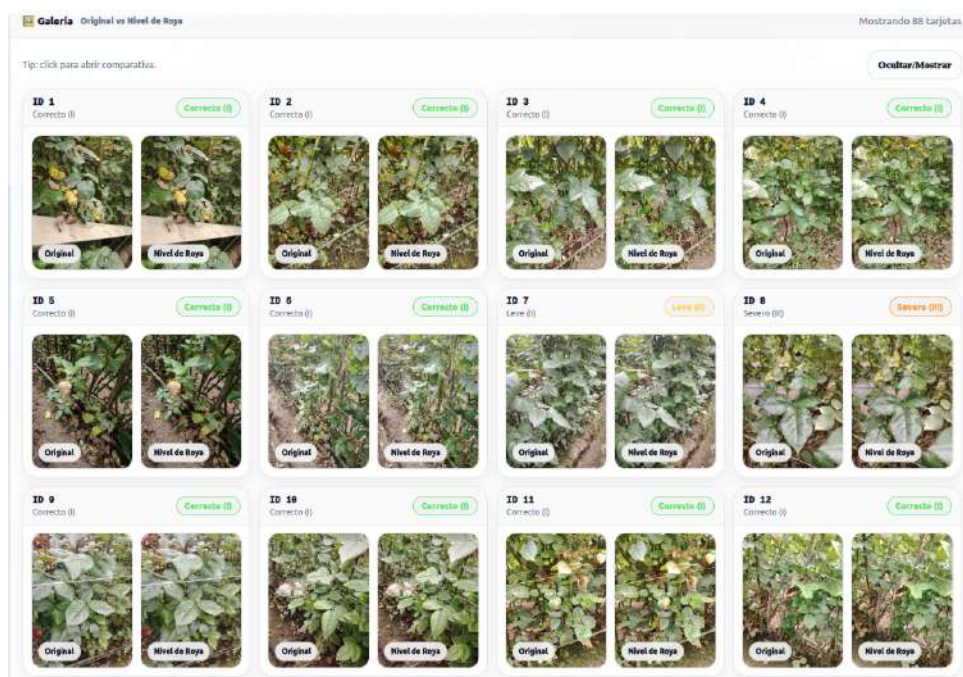
#	archivo_id	usuario_id	tipo	archivo	nombre_personalizado	fecha_subida
1	15	1	imagen	det_20260113_192709_IMG_2026...	Prueba 1	2026-01-14 00:27:09
2	16	1	imagen	det_20260113_192751_IMG_2026...	Prueba 2	2026-01-14 00:27:53
3	17	1	imagen	det_20260113_192907_IMG_2026...	Prueba 3	2026-01-14 00:29:08
4	18	1	imagen	det_20260113_192923_IMG_2026...	Prueba 4	2026-01-14 00:29:24
5	19	1	imagen	det_20260113_192940_IMG_2026...	Prueba 5	2026-01-14 00:29:41
6	20	1	imagen	det_20260113_193016_IMG_2026...	Prueba 6	2026-01-14 00:30:18
7	21	1	imagen	det_20260113_193053_IMG_2026...	Prueba 7	2026-01-14 00:30:34
8	22	1	imagen	det_20260113_193053_IMG_2026...	Prueba 7	2026-01-14 00:30:54
9	23	1	imagen	det_20260113_193117_IMG_2026...	Prueba 8	2026-01-14 00:31:19

**Nota.** Elaborado por el autor.

El sistema facilita la comparación entre la imagen original y la detectada para validar el proceso de detección permitiendo establecer la gravedad de cada waypoint. El proceso se muestra en la figura 77, lo cual facilita un análisis más preciso del estado de las cosechas en el invernadero, dado que se puede observar cuáles áreas están afectadas y cuán severa es la afectación. Con esta información, es posible elaborar reportes acerca de la situación actual del cultivo y determinar las medidas más adecuadas para evitar que la enfermedad se propague.

Figura 77.

Verificación de los resultados de detección en la interfaz web.



**Nota.** Elaborado por el autor.

**Tabla 34.**

Validación de reportes

Software	¿Prueba de despliegue?	Sí / No
<b>Descripción:</b>		
Validar la generación de resultados visuales del sistema, incluyendo la visualización del estado del cultivo mediante mapas de severidad y la generación de reportes en formato PDF.		
<b>Prerrequisitos:</b>		
1. Resultados de detección previamente almacenados en el sistema. 2. Procesamiento de imágenes completado correctamente. 3. Módulo de generación de reportes habilitado.		
<b>Pasos:</b>		
<ul style="list-style-type: none"> <li>• Generar el mapa visual del cultivo a partir de los resultados de detección.</li> <li>• Generar el reporte del análisis en formato PDF.</li> </ul>		
<b>Resultados esperados:</b>		
El sistema genera el mapa de severidad del cultivo y produce un reporte PDF claro de las zonas afectadas.		

*Nota.* Elaborado por el autor.

#### Resultados obtenidos:

El mapa de severidad se genera a partir de los resultados del proceso de detección y las coordenadas geográficas de la ruta y los waypoints definidos. En él se representan cuatro niveles de gravedad: leve, grave, crítico y sano. Además, se muestran 86 puntos estratégicos con su ubicación y nivel de severidad, lo que permite analizar el estado del cultivo en las distintas áreas del invernadero como se indica en la Figura 78.

#### Figura 78.

*Verificación de los resultados de detección en la interfaz web.*



*Nota.* Elaborado por el autor.

Como se observa en la figura 79, el sistema genera el reporte según el nivel de gravedad y el waypoint de cada detección, incorporando coordenadas geográficas para visualizar con precisión las áreas afectadas y apoyar la decisión sobre fungicidas.

**Figura 79.**

*Verificación de los resultados de detección en la interfaz web.*

UNIVERSIDAD TÉCNICA DEL NORTE			
Sistema: Detección automatizada de roya con IA			
Detección ID: 177 - Fecha: 11/12/2026, 19:38:05			
Autor: Farid Avila - Director Msc. Carlos Viquez			
<b>Resumen de severidad</b>			
● Crítico (IV)			
<b>Detalle (Waypoints)</b>			
ID	LAT	LONG	Severidad
18	85839.099000	-78103440.000998	Correcto (I)
19	85837.099000	-78103886.000998	Correcto (I)
20	85821.099000	-78103895.000998	Correcto (I)
21	85784.099000	-78103186.000998	Correcto (I)
22	85757.099000	-78103185.000998	Correcto (I)
23	85730.099000	-78103119.000998	Correcto (I)
24	85704.099000	-78103115.000998	Correcto (I)
25	85677.099000	-78103128.000998	Severo (III)
26	85650.099000	-78103124.000998	Correcto (I)
27	85624.099000	-78103128.000998	Lleve (II)
28	85670.099000	-78103157.000998	Severo (III)
29	85691.099000	-78103153.000998	Severo (III)
30	85713.099000	-78103126.000998	Correcto (I)
31	85734.099000	-78103147.000998	Correcto (I)
32	85755.099000	-78103144.000998	Severo (III)
33	85777.099000	-78103148.000998	Correcto (I)
34	85798.099000	-78103137.000998	Correcto (I)
35	85819.099000	-78103134.000998	Correcto (I)

*Nota.* Elaborado por el autor.

**Tabla 35.**

Test de funcionamiento en escenario real de cultivo

Subsistema Prueba	¿Prueba de despliegue?	Entorno
Escenario de pruebas en el cultivo	Sí	Campo real
<b>Descripción:</b>		
Validar el correcto funcionamiento del prototipo dentro del cultivo de rosas, bajo condiciones controladas, para la captura de imágenes mediante detección aérea o terrestre.		
<b>Prerrequisitos:</b>		
<ol style="list-style-type: none"> <li>1. Área de cultivo de rosas disponible para la ejecución de pruebas.</li> <li>2. Dron operativo y en condiciones seguras de vuelo.</li> <li>3. Sistema de detección configurado y listo para su operación.</li> </ol>		
<b>Pasos:</b>		
<ul style="list-style-type: none"> <li>● Seleccionar el tipo de detección a emplear (aérea o terrestre).</li> <li>● Ejecutar las pruebas en el horario establecido entre las 08h30 y 10h30.</li> </ul>		
<b>Resultados esperados:</b>		
El prototipo opera correctamente en el escenario real de cultivo, permitiendo la captura de imágenes con condiciones de iluminación adecuadas para la detección aérea o terrestre.		

*Nota.* Elaborado por el autor.

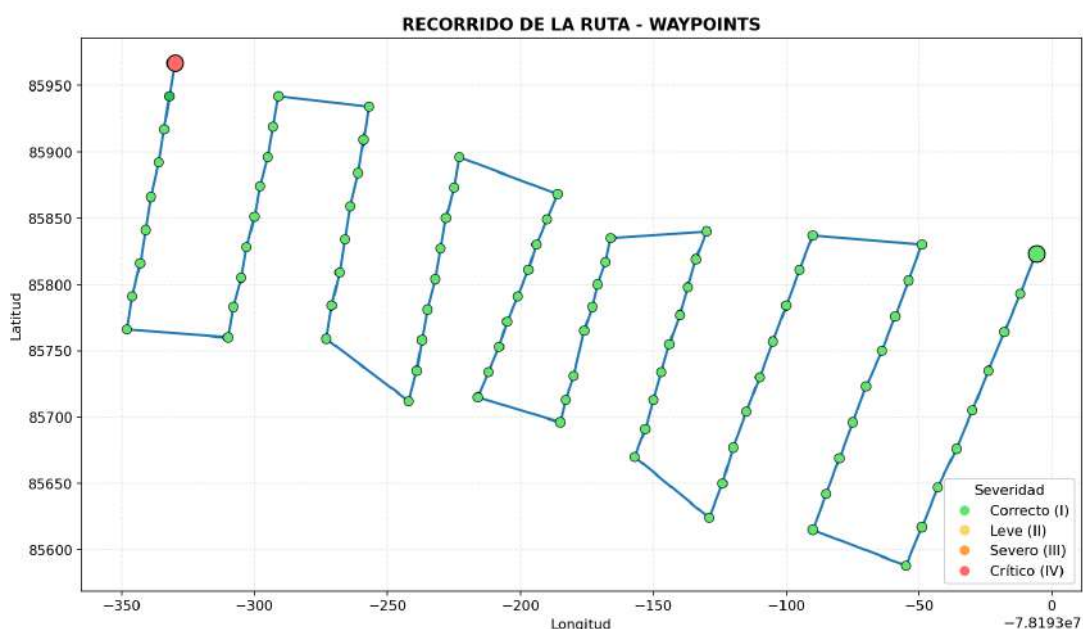
## Resultados obtenidos:

### Escenario de pruebas A: Captura aérea

El rendimiento del sistema de detección de roya fue evaluado durante las pruebas con el dron DJI Air 3. La toma aérea se llevó a cabo de 08:00 a 09:00, cuando había buena luz, lo que permitió obtener imágenes nítidas y con un buen enfoque. El dron voló a unos 3 metros de altura, evitando los obstáculos y proporcionando una visión amplia del área. La figura 80 ilustra el resultado con los 86 puntos de referencia, lo que demuestra que la mayoría de las plantas están sanas, sin manifestar niveles leves, severos ni críticos de roya. Esto confirma el estado óptimo general del área inspeccionada.

**Figura 80.**

*Mapa de severidad del cultivo – Prueba 1 aérea*



**Nota.** Elaborado por el autor

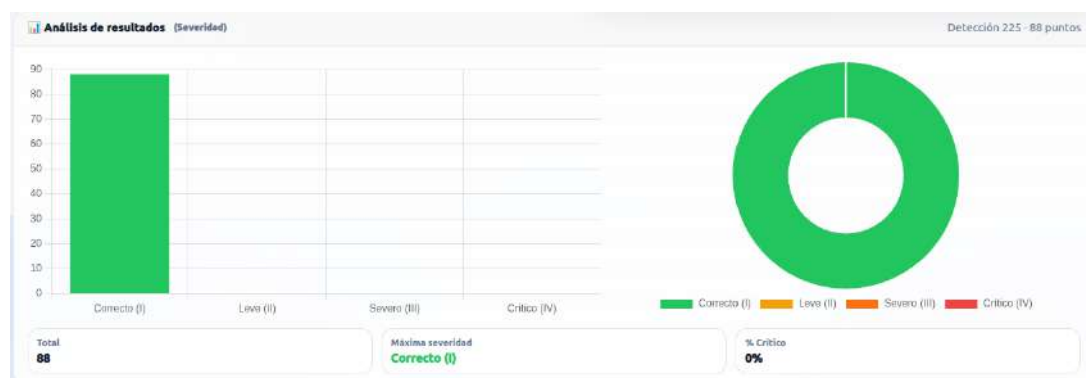
En relación con los resultados representados en el mapa de la Figura 81, se identifican los porcentajes correspondientes a los grados de severidad evaluados. Para esta toma de muestras, los resultados evidencian que el 100 % de los puntos analizados corresponde a un estado sano, sin presencia de niveles potencialmente enfermos, marchitos o enfermos. Este comportamiento indica que, bajo las condiciones de evaluación establecidas, el cultivo presenta un estado fitosanitario óptimo, sin indicios de afectación por roya en el área analizada.

**Figura 81.***Niveles severidad del cultivo – Prueba 1 aérea*

**Nota.** Elaborado por el autor

La Figura 82 demuestra que no se registraron casos en los niveles Leve (II), Severo (III) o Crítico (IV), presentándose un 0% de severidad crítica. Esto indica que, durante el proceso de captura y análisis de las imágenes, el cultivo se encontraba en buenas condiciones fitosanitarias, sin evidencias visibles de síntomas asociados a la roya (*Phragmidium mucronatum*).

De igual manera, los resultados permiten observar un funcionamiento estable del sistema de visión artificial, ya que no se generaron alertas innecesarias ni clasificaciones en niveles de mayor gravedad. Esto refuerza la confiabilidad del modelo para reconocer correctamente estados saludables del cultivo, lo cual resulta fundamental en un sistema de monitoreo preventivo.

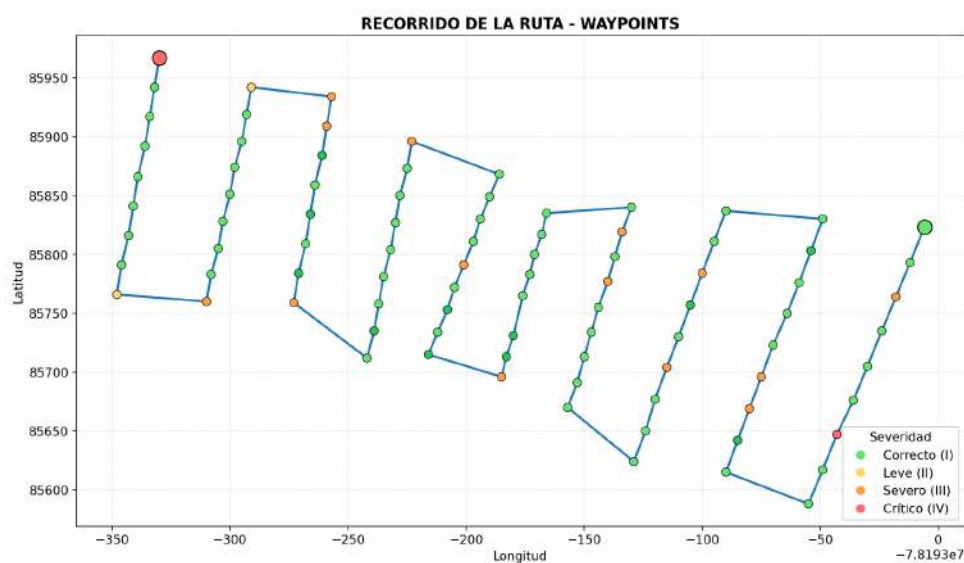
**Figura 82.***Análisis de severidad del cultivo – Prueba 1 aérea*

**Nota.** Elaborado por el autor

La segunda prueba se demuestra la capacidad del sistema, evaluando distintas horas del día. Se consideró para el análisis de 09:00 a 10:00, a una altura de tres metros. En la Figura 83 se muestra el mapa con los 86 waypoints indicados por colores según el nivel de severidad. Las áreas con gravedad III y IV representan zonas críticas, mientras que la mayoría permanece saludable, con solo una pequeña proporción muestran síntomas de la enfermedad por roya. Los resultados indican que cerca del 80 % del cultivo está en condición fitosanitaria adecuada, aunque el 20 % restante presenta distintos grados de afectación.

**Figura 83.**

*Mapa de severidad del cultivo – Prueba 2 aérea*



**Nota.** Elaborado por el autor

Según lo que se muestra en el gráfico de la Figura 84, el 80,20 % del área examinada permanece en estado saludable, lo que indica una condición fitosanitaria favorable. No obstante, el 16,27 % muestra una severidad grave, mientras que los niveles leve y crítico representan el 2,32 % y el 1,26 %, respectivamente.

A pesar de que el estado sano es el predominante, la existencia de áreas con severidad crítica y severa es importante porque podrían transformarse en focos de propagación. Por lo tanto, se aconseja fortalecer las medidas de control y seguimiento en las zonas más damnificadas para evitar que la enfermedad progrese.

**Figura 84.**

*Niveles severidad del cultivo – Prueba 2 aérea*



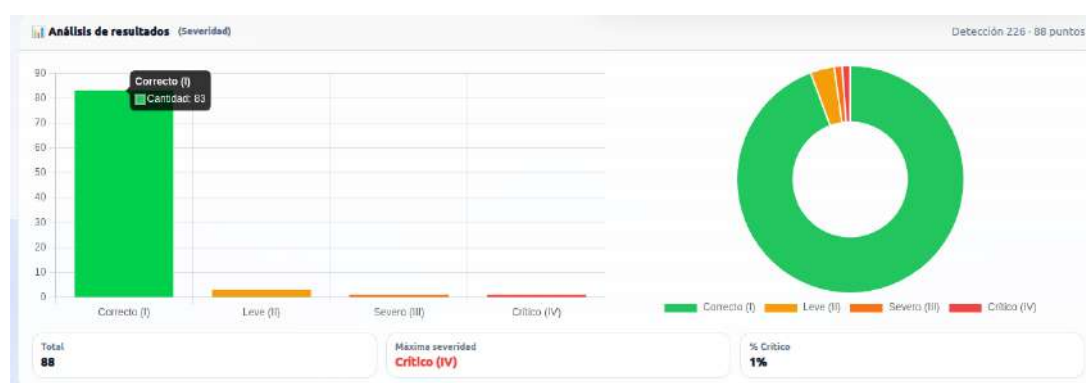
**Nota.** Elaborado por el autor

La Figura 85 demuestra que la mayoría de los puntos evaluados fueron clasificados en el nivel Correcto (I); sin embargo, también se registraron algunos casos en los niveles Leve (II), Severo (III) y Crítico (IV), presentándose un 1 % de severidad crítica. Esto indica que el cultivo presentaba síntomas de roya (*Phragmidium mucronatum*), aunque en una proporción mínima.

De igual manera, los resultados evidencian que el sistema de visión artificial logró identificar correctamente estas variaciones en la severidad, diferenciando entre áreas saludables y zonas afectadas. Esto confirma su capacidad para detectar oportunamente la presencia de la enfermedad dentro del proceso de monitoreo.

**Figura 85.**

*Análisis de severidad del cultivo – Prueba 2 aérea*



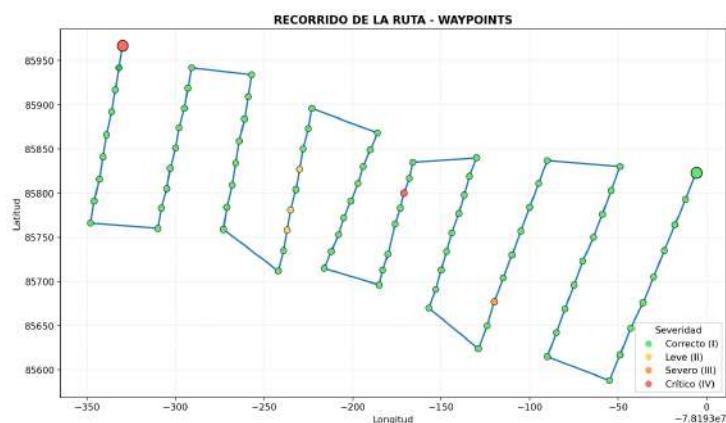
**Nota.** Elaborado por el autor

La tercera prueba se realizó entre las 10:00 y 11:00 por lo que todavía se detectó roya en niveles leve, severo y crítico. Aun así, la mayor parte del resultado del cultivo se ve saludable. En la Figura 86 se puede ver cómo las

áreas críticas y severas son pocas, mientras que la mayoría de la plantación mantiene un buen estado general bajo las condiciones actuales de luz y manejo.

**Figura 86.**

*Mapa de severidad del cultivo – Prueba 3 aérea*



*Nota.* Elaborado por el autor

Los niveles de severidad obtenidos en la tercera prueba muestran que el 94,18 % del cultivo se encuentra en estado sano, mientras que el 4,48 % presenta un nivel leve y el 1,16 % corresponde tanto a los niveles severo como crítico, tal como se observa en la Figura 87.

**Figura 87.**

*Niveles severidad del cultivo – Prueba 3 aérea*



*Nota.* Elaborado por el autor

La Figura 88 demuestra que la mayoría de los puntos evaluados fueron clasificados en el nivel Correcto (I); no obstante, se identificaron algunos casos en el nivel Severo (III), sin registrarse presencia en los niveles Leve (II) ni Crítico (IV), presentándose un 0% de severidad crítica. Esto indica que el cultivo presentaba áreas puntuales con un grado importante de afectación, aunque sin alcanzar un estado crítico.

De igual manera, los resultados evidencian que el sistema de visión artificial logró detectar correctamente estos focos de mayor severidad, diferenciándolos de las zonas saludables. Esto confirma la capacidad del modelo para identificar distintos niveles de avance de la enfermedad dentro del proceso de monitoreo fitosanitario.

**Figura 88.**

*Análisis de severidad del cultivo – Prueba 3 aérea*



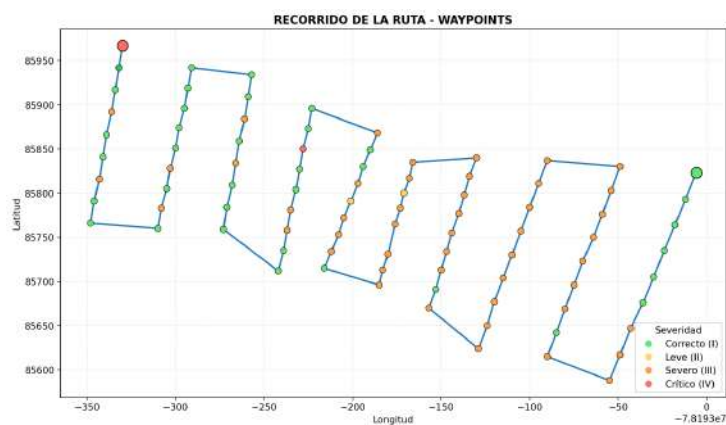
*Nota.* Elaborado por el autor

### Escenario de pruebas B: Captura terrestre

Para el segundo escenario, las pruebas se realizaron en los mismos horarios establecidos previamente. La roya tiende a generarse desde el nivel del suelo, debido a los cambios de temperatura, la humedad por lo que influyen directamente en su desarrollo sobre las plantas de rosa. Tras la captura de imágenes entre las 08:00 y 09:00, se evidenció una mayor presencia de la enfermedad en los 86 waypoints, indicando que estas zonas son más vulnerables bajo las condiciones ambientales mencionadas. Esto permite establecer medidas de manejo oportunas para prevenir la propagación de la roya como se presenta en la Figura 89.

**Figura 89.**

*Mapa de severidad del cultivo – Prueba 1 terrestre*



*Nota.* Elaborado por el autor

Una vez realizado el proceso de detección, se determinaron los porcentajes de los niveles de severidad para la primera prueba a nivel del suelo. Como se observa en la Figura 90, se evidencia una mayor afectación del cultivo por la enfermedad: el 50 % corresponde a detección severa, mientras que el 41 % del cultivo se mantiene en buen estado sano. Además, un 2 % presenta afectación leve y un 1 % se encuentra en estado crítico.

**Figura 90.**

*Niveles severidad del cultivo – Prueba 1 terrestre*



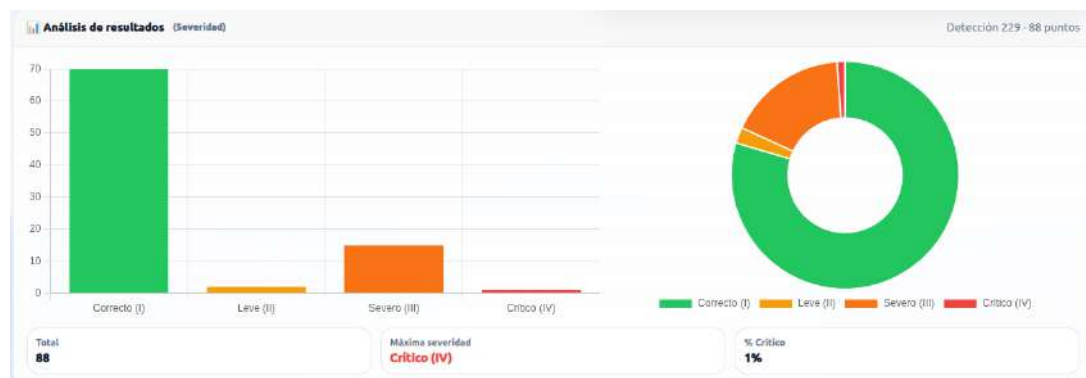
*Nota.* Elaborado por el autor

En la Figura 91 se observa que, dentro de las pruebas realizadas mediante captura terrestre, la mayor parte de los puntos evaluados fueron clasificados en el nivel Correcto (I). No obstante, también se identificaron casos en los niveles Leve (II), Severo (III) y Crítico (IV), registrándose un 1 % de severidad crítica.

Estos resultados evidencian que, a nivel de inspección terrestre, el sistema logró detectar diferentes grados de afectación dentro del cultivo, incluyendo focos con mayor avance de la enfermedad. La presencia de niveles severos y críticos, aunque en baja proporción, indica que existían zonas específicas que requerían atención fitosanitaria.

**Figura 91.**

*Análisis de severidad del cultivo – Prueba 1 terrestre*

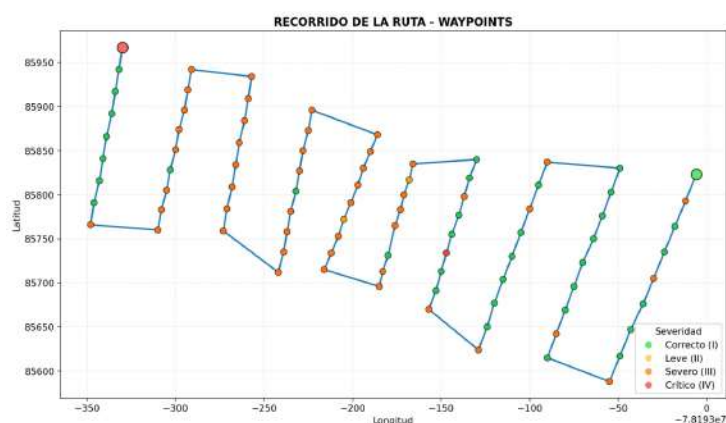


*Nota.* Elaborado por el autor

Para la segunda prueba, el análisis se llevó a cabo en el horario de 09:00 a 10:00. Durante este periodo, las condiciones de iluminación fueron favorables para la captura de imágenes, lo que permitió identificar con mayor claridad la presencia de roya. Como se muestra en la Figura 92, se observa un aumento en el número de detecciones, destacándose principalmente niveles de severidad altos, correspondientes a estados severos y críticos. En contraste, solo una parte reducida de las plantas evaluadas se mantiene en buen estado.

**Figura 92.**

*Mapa de severidad del cultivo – Prueba 2 terrestre*

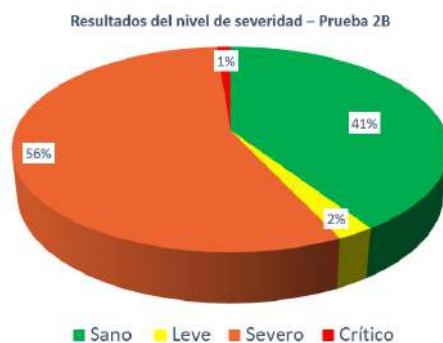


*Nota.* Elaborado por el autor

La presencia de la roya en el cultivo se confirma a través de los resultados de la segunda prueba de detección que se realizó, según este análisis, el 56 % de las detecciones tiene un nivel alto de severidad y el 41 % está en buen estado. En menor proporción, el 2 % presenta síntomas leves, y solo el 1 % se encuentra en una condición crítica como se evidencia en la Figura 93, es necesario enfocar los esfuerzos de control en las áreas donde se encontró una mayor incidencia y hacerlas más intensivas.

**Figura 93.**

*Niveles severidad del cultivo – Prueba 2 terrestre*



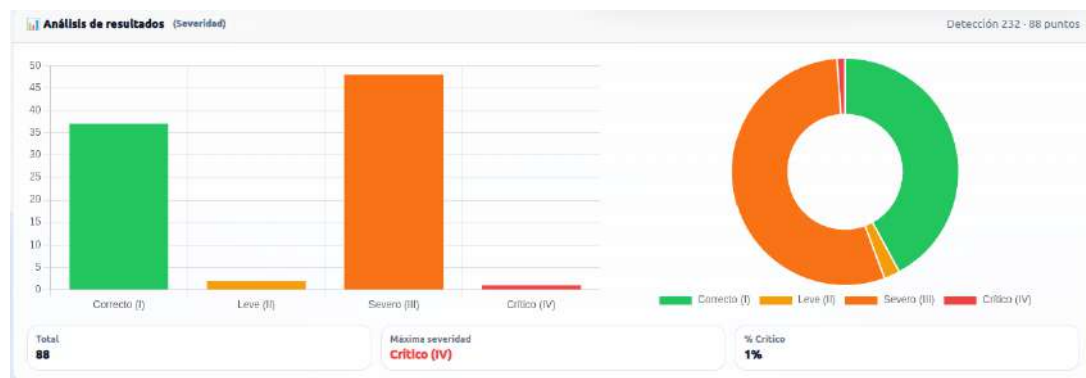
*Nota.* Elaborado por el autor

En la Figura 94 correspondiente a las pruebas realizadas mediante captura terrestre, se evidencia una mayor presencia de niveles de severidad intermedios y altos dentro del cultivo.

Este comportamiento indica que, durante esta evaluación, el cultivo presentaba focos más desarrollados de la enfermedad, reflejando un escenario de mayor afectación en comparación con otras pruebas.

**Figura 94.**

*Análisis de severidad del cultivo – Prueba 2 terrestre*

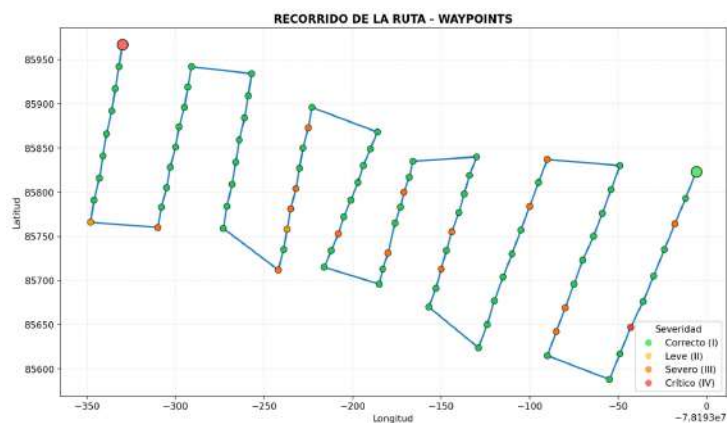


*Nota.* Elaborado por el autor

Finalmente, tras realizar las detecciones correspondientes a la tercera prueba, se pudo verificar que, en el horario de 10:00 a 11:00, el sistema evidenció una menor presencia de roya durante el análisis del cultivo. Tal como se observa en la Figura 95, los resultados permiten apreciar el impacto de la enfermedad en el cultivo, mostrando una disminución en los casos detectados en comparación con pruebas anteriores.

**Figura 95.**

*Mapa de severidad del cultivo – Prueba 3 terrestre*



*Nota.* Elaborado por el autor

Los resultados de esta prueba de detección muestran que el cultivo se encuentra en buen estado en su mayoría, dentro del análisis se obtuvo que el 79 % del área muestreada está en buenas condiciones esto significa que la mayoría de las detecciones no evidencian presencia de roya. Sin embargo, un 18 % del cultivo presenta niveles altos de la enfermedad, lo que indica que aún existen áreas donde la roya sigue siendo un problema. Además, un 2 % tiene síntomas leves y un 1 % está en estado crítico tal como se evidencia en la Figura 96. Estos resultados exponen la necesidad de un monitoreo continuo y de medidas de control localizadas en las zonas más afectadas para prevenir que la enfermedad se propague al resto del cultivo.

**Figura 96.**

*Niveles severidad del cultivo – Prueba 3 terrestre*

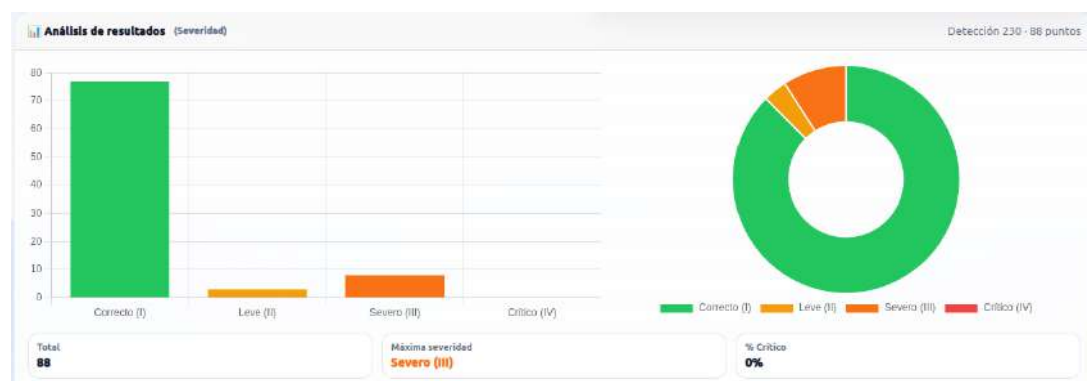


**Nota.** Elaborado por el autor

En la Figura 97, correspondiente a la tercera prueba de captura terrestre, se aprecia que la mayor parte de los puntos evaluados se mantienen en el nivel Correcto (I). Sin embargo, también se identificaron casos en los niveles Leve (II) y Severo (III), sin registrarse presencia en el nivel Crítico (IV), reflejando un 0 % de severidad crítica.

**Figura 97.**

*Análisis de severidad del cultivo – Prueba 3 terrestre*



**Nota.** Elaborado por el autor

**Tabla 36.**

Test de validación de precisión de las detecciones por escenario

<b>Subsistema Prueba</b>	<b>¿Prueba de validación?</b>	<b>Entorno</b>
Sistema de detección de roya	Sí	Campo real
<b>Descripción:</b>		
Validar la precisión de los resultados del sistema de detección de roya en los distintos escenarios de prueba mediante la matriz de confusión, evaluando su desempeño en la detección.		
<b>Prerrequisitos:</b>		
<ol style="list-style-type: none"> <li>1. Imágenes capturadas en los distintos escenarios de prueba.</li> <li>2. Sistema de detección correctamente entrenado y configurado.</li> <li>3. Observación y validación visual en campo para comparación de resultados.</li> </ol>		
<b>Pasos:</b>		
<ul style="list-style-type: none"> <li>• Comparar los resultados obtenidos con la observación real del cultivo.</li> <li>• Construir la matriz de confusión para cada escenario.</li> </ul>		
<b>Resultados esperados:</b>		
El sistema proporciona métricas cuantitativas confiables para cada escenario de prueba, permitiendo evaluar la precisión del modelo, su capacidad de detección y el tiempo de respuesta durante la ejecución de las predicciones.		

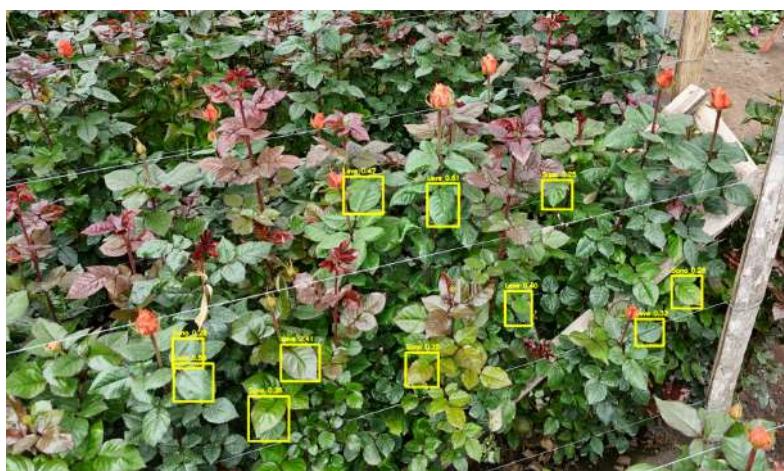
*Nota.* Elaborado por el autor.

#### **Resultados obtenidos:**

Se examinan los resultados que se han conseguido a partir de las pruebas efectuadas en el invernadero, donde el sistema fue evaluado en un ambiente controlado, durante esta fase de validación. Como se puede ver en la Figura 98, el sistema es capaz de realizar detecciones y visualizar el estado del cultivo en los diferentes waypoints, lo que demuestra un rendimiento apropiado en la detección de dicha enfermedad.

#### **Figura 98.**

*Validación de precisión de las detecciones en el cultivo*



*Nota.* Elaborado por el autor

Los resultados se presentan para cada uno de los escenarios de prueba, considerando sus respectivos parámetros y utilizando la matriz de confusión para obtener los valores de verdaderos positivos (VP), verdaderos negativos (VN), falsos positivos (FP) y falsos negativos (FN). A partir de estos datos se calculan las métricas de precisión (Pr), recall o sensibilidad (Rec), exactitud (Exact) y el tiempo de respuesta (Ts) de cada una de las muestras de predicción generadas por el sistema, las cuales se resumen en la Tabla 37.

**Tabla 37.**

Validación del desempeño del sistema de detección

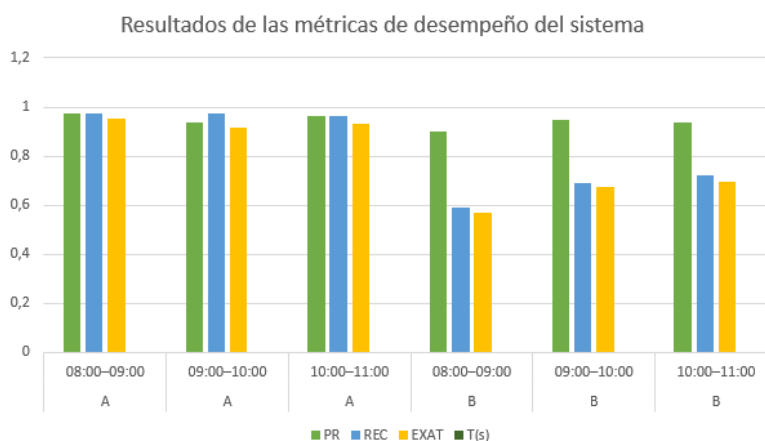
Escenario	Descripción	VP	VN	FP	FN	Pr	Rec	Exac	T (s)
A	08:00–09:00; 3 m;	79	3	2	2	0.9753	0.9753	0.9535	30.20
A	09:00–10:00; 3 m;	77	2	5	2	0.9390	0.9747	0.9186	32.09
A	10:00–11:00; 3 m;	78	2	3	3	0.9630	0.9630	0.9302	46.75
B	08:00–09:00; 3 m;	46	3	5	32	0.9020	0.5897	0.5698	32.07
B	09:00–10:00; 3 m;	56	2	3	25	0.9492	0.6914	0.6744	30.19
B	10:00–11:00; 3 m;	57	3	4	22	0.9344	0.7215	0.6977	33.12

**Nota.** Elaborado por el autor.

Una vez validadas las métricas de rendimiento, se pudo verificar que el escenario A, analizado entre las 08:00 y 11:00, tuvo un buen rendimiento. En este caso, la exactitud se mantuvo por encima de 0.93, con un máximo de 0.9753, y la sensibilidad también fue alta, hasta 0.9753. La precisión se mantuvo alrededor del 95 %, lo que demuestra un alto número de clasificaciones correctas. Por el contrario, en el escenario B se observó una caída del rendimiento, sobre todo en la sensibilidad, que llegó a su mínimo (0.5897), evidenciando falsos negativos. En ambos casos, el tiempo de procesamiento fue constante, con un promedio de 32 segundos, lo que demuestra la efectividad del sistema. Estos resultados se visualizan de manera más clara en la Figura 99.

**Figura 99.**

Validación de precisión de las detecciones en el cultivo



**Nota.** Elaborado por el autor

## CONCLUSIONES

- El sistema de visión artificial desarrollado e implementado a través de un dron fue efectivo para detectar roya (*Phragmidium mucronatum*) en cultivos de rosa. A través de imágenes tomadas en puntos estratégicos del cultivo (waypoints), el sistema pudo visualizar el estado fitosanitario de las plantas y advertir con tiempo sobre la enfermedad, ayudando en la toma de decisiones del agricultor.
- Los resultados de la evaluación del sistema demostraron que en el escenario A, en condiciones de buena iluminación y en el horario de 08:00 a 11:00, el rendimiento fue muy alto. En esta se obtuvieron altos valores de precisión, sensibilidad y exactitud, lo que indica que el sistema es fiable cuando trabaja en condiciones controladas.
- En el escenario B, el sistema se desempeñó inferiormente, sobre todo en la capacidad de identificar la enfermedad (sensibilidad), ya que aumentaron los falsos negativos. Aun así, la exactitud se mantuvo alta, lo que significa que cuando el sistema detecta roya, en realidad la detecta correctamente. Estos datos demuestran cómo las condiciones ambientales (la luz) afectan el sistema.
- El uso de redes neuronales convolucionales permite dar paso a una solución tecnológica para el monitoreo agrícola para la detección de roya en cultivos de rosas por lo que el sistema pudo identificar con éxito los patrones visuales relacionados con la roya, logrando un buen equilibrio entre precisión en la detección y eficiencia en el procesamiento de imágenes.
- Para el diseño del sistema se lo realizó siguiendo los parámetros como altura de vuelo, número de waypoints y condiciones ambientales, esto permitió evaluar el estado del cultivo en un ambiente controlado. Esta metodología integral permitió alcanzar resultados consistentes con los objetivos planteado para el prototipo.
- Como resultado del uso de 86 waypoints estratégicos y la ejecución de vuelos controlados, se logró obtener imágenes con la calidad apropiada para el análisis. Esto permitió asegurar un alto nivel de detalle y enfoque en las muestras empleadas para detectar la roya.
- El conjunto de datos se compone de cerca de 3000 imágenes etiquetadas, las cuales fueron organizadas bajo criterios visuales y técnicos preestablecidos, por lo que este conjunto de datos es útil para el sistema de detección por lo que incluyen información importante acerca de las propiedades propias de la roya, como su color, estructura y forma, para identificar la enfermedad en el cultivo.
- Finalmente los tiempos de procesamiento obtenidos, son valores promedio cercanos a los 32 segundos, muestran que el sistema puede operar de manera eficiente en escenarios reales de monitoreo agrícola. Esto permite realizar evaluaciones periódicas del cultivo sin comprometer el rendimiento ni la operatividad del sistema.

## RECOMENDACIONES

- Se recomienda para obtener las imágenes en los diversos waypoints de manera más eficiente, se sugiere planear el recorrido aéreo del dron teniendo en cuenta un ambiente despejado y condiciones controladas. Esto permite mejorar la calidad de imágenes durante la captura de datos dentro del monitoreo del cultivo y a optimizar el tiempo de vuelo.
- Para garantizar que el sistema de detección esté funcionando correctamente durante las pruebas es aconsejable llevar a cabo calibraciones del sistema y vuelos previos, con la finalidad de obtener una estabilidad adecuada durante el vuelo y una cobertura total del área cultivada, estas pruebas deben incluir tanto la captura de imágenes como el control del dron.
- se recomienda verificar que el conjunto de datos correspondientes al dataset tenga imágenes que representen adecuadamente las diferentes manifestaciones visuales de la roya, asegurando así que el dataset contenga diferencias en color, textura y forma que posibiliten una mejor generalización del aprendizaje del modelo.
- También se recomienda que la aplicación web permita visualizar los resultados correctamente, esto permite establecer que las detecciones que realiza el sistema se muestren de un modo organizado y entendible, lo cual evitará posibles contratiempos en su implementación.
- Para realizar el entrenamiento del modelo de detección se recomienda disponer de una placa con una capacidad de procesamiento alta al igual de una tarjeta gráfica adecuada y suficiente memoria RAM, esto garantiza que el sistema pueda procesar las imágenes a detectar dentro del sistema.
- Finalmente se aconseja mejorar la precisión del modelo por lo que permite minimizar el número de falsos negativos y falsos positivos en la detección de la roya, para ello se recomienda también aumentar el volumen de imágenes utilizadas para el entrenamiento, garantizando un buen nivel de detección para el sistema.

## BIBLIOGRAFÍA

- Aguilar, J., & Campoverde, M. (2019). *Introducción al aprendizaje automático con yolo*. Instituto Tecnológico de Costa Rica. <https://revistas.ulatina.ac.cr/index.php/tecnologiavital/article/download/250/260/547>
- Alessandro Giusti. (2024). *An Introduction to Convolutional Neural Networks*. [https://boracchi.faculty.polimi.it/teaching/IC/IC\\_Lez4-5-6\\_convolutional%20nets.pdf](https://boracchi.faculty.polimi.it/teaching/IC/IC_Lez4-5-6_convolutional%20nets.pdf)
- Amazon Web Services, I. (2024). ¿Qué es la visión artificial? - Explicación de la IA y el aprendizaje automático de imágenes. <https://aws.amazon.com/es/what-is/computer-vision/>
- Andrés Ares, J. L. (2015). *Plantas leñosas ornamentales: control de enfermedades producidas por hongos y cromistas*. Ediciones Mundi Prensa. [https://books.google.com.ar/books?id=-j3cCgAAQBAJ&printsec=frontcover&source=gbs\\_vpt\\_read](https://books.google.com.ar/books?id=-j3cCgAAQBAJ&printsec=frontcover&source=gbs_vpt_read)
- Aparicio del Río, I. (2005). *Algoritmo para la detección automática de umbral para la segmentación de imágenes en color utilizando conjuntos borrosos*. Universidad de Sevilla. <https://biblus.us.es/bibing/proyectos/use/abreproy/10795/direccion/>
- Bajaña Calle, L. F., & Pin Guerrero, E. B. (2024). Reconocimiento de la plaga mosca blanca (Aleuródidos) en los cultivos hortícolas aplicando técnicas de visión artificial. <https://cia.uagraria.edu.ec/Archivos/BAJA%20C3%91A%20CALLE%20LESLIE%20FIORELLA%20Y%20PIN%20GUERRERO%20ERICKA%20BRIGGITTE.pdf>
- Bansal, P. (2025). AI Pattern Recognition and its Features. *International Journal of Engineering and Advanced Technology*, 14, 18. [https://www.researchgate.net/publication/389173710\\_AI\\_Pattern\\_Recognition\\_and\\_its\\_Features](https://www.researchgate.net/publication/389173710_AI_Pattern_Recognition_and_its_Features)
- Bonillo Carrasco, M. (2021). *Una propuesta de gestión de enfermedades en el Parque del Oeste de la ciudad de Valencia* [Trabajo de fin de grado]. Universidad Politécnica de Valencia. <https://riunet.upv.es/server/api/core/bitstreams/c6d63796-e599-4d2a-8a76-1962e2aad665/content>
- Borrella, B. (2022). Introducción a la visión artificial: Procesos y aplicaciones. *Universidad Complutense de Madrid*. <https://docta.ucm.es/bitstreams/ddebb1a6-8474-4a20-91cb-78247481577f/download>
- Chowdhury, R., Green, A., Park, C., Maclennan, J., & Clark, N. (2023). Topological defect coarsening in quenched smectic- C films analyzed using artificial neural networks. *Physical Review E*, 107. <https://doi.org/10.1103/PhysRevE.107.044701>
- Corporación Financiera Nacional. (2020). *Fichas sectoriales: Cultivo de flores*. [https://www.cfn.fin.ec/wp-content/uploads/downloads/biblioteca/2019/Fichas-sectoriales-3-Trimestre-2019/FS\\_Cultivo-de-Flores.pdf](https://www.cfn.fin.ec/wp-content/uploads/downloads/biblioteca/2019/Fichas-sectoriales-3-Trimestre-2019/FS_Cultivo-de-Flores.pdf)
- Da Jiang Innovations. (2024). *Agricultural drone industry insight report*. [https://www1.djicdn.com/cms\\_uploads/ckeditor/attachments/9171/03e81f9a23cf4df447b66c91c43d929a.pdf](https://www1.djicdn.com/cms_uploads/ckeditor/attachments/9171/03e81f9a23cf4df447b66c91c43d929a.pdf)
- Digital Talent Agency. (2018). Modelo Waterfall o en Cascada. [https://www.dtagency.tech/cursos/metodologias\\_gestion\\_proyectos/tema\\_1-ModeloWaterfall.pdf](https://www.dtagency.tech/cursos/metodologias_gestion_proyectos/tema_1-ModeloWaterfall.pdf)

- DJI. (2023). *DJI Air 3 User Manual v1.0*. Da-Jiang Innovations Science y Technology Co., Ltd. [https://dl.djicdn.com/downloads/DJI\\_Air\\_3/UM/DJI\\_Air\\_3\\_User\\_Manual\\_v1.0\\_ES.pdf](https://dl.djicdn.com/downloads/DJI_Air_3/UM/DJI_Air_3_User_Manual_v1.0_ES.pdf)
- Ekos Negocios. (2024). *Ecuador es el tercer exportador mundial de flores*. <https://ekosnegocios.com/articulo/ecuador-es-el-tercer-exportador-mundial-de-flores>
- Espinoza, M. (2017). GOBIERNO DE LAS TECNOLOGÍAS DE LA INFORMACIÓN (TI): ANALISIS EXPLORATORIO EN EL SECTOR PÚBLICO DEL ECUADOR.
- Exflores. (2025). *Reporte estadístico mensual – Enero 2025*. [https://www.expoflores.com/\\_files/ugd/d0e607\\_01b2a53be5c84a76ab8771bde881b25e.pdf](https://www.expoflores.com/_files/ugd/d0e607_01b2a53be5c84a76ab8771bde881b25e.pdf)
- Fernandez Acevedo, V. (2022). *Cultivo y manejo de rosa*. Universidad Nacional de La Plata. [https://aulavirtual.agro.unlp.edu.ar/pluginfile.php/116112/mod\\_resource/content/1/Cultivo%20y%20Manejo%20de%20Rosa.pdf](https://aulavirtual.agro.unlp.edu.ar/pluginfile.php/116112/mod_resource/content/1/Cultivo%20y%20Manejo%20de%20Rosa.pdf)
- Gill, S., Shrewsbury, P., Rane, K., Clement, D., & Harding, N. (2021). Landscape and Nursery IPM Report. *Universidad de Maryland*. <https://extension.umd.edu/sites/extension.umd.edu/files/2021-07/21Jul02L.pdf?utm>
- González Marcos. (2006). *Técnicas y algoritmos básicos de visión artificial* (Vol. 24). Universidad de La Rioja, Servicio de Publicaciones. <https://publicaciones.unirioja.es/catalogo/online/VisionArtificial.pdf>
- Herrera, D., Escudero, P., Cárdenas, E., Ortiz, M., & Varela Aldás, J. (2024). Combining Image Classification and Unmanned Aerial Vehicles to Estimate the State of Explorer Roses. *AgriEngineering*, 6, 1008-1021. [https://www.researchgate.net/publication/379864411\\_Combining\\_Image\\_Classification\\_and\\_Unmanned\\_Aerial\\_Vehicles\\_to\\_Estimate\\_the\\_State\\_of\\_Explorer\\_Roses](https://www.researchgate.net/publication/379864411_Combining_Image_Classification_and_Unmanned_Aerial_Vehicles_to_Estimate_the_State_of_Explorer_Roses)
- Horst, R. K., & Cloyd, R. A. (2020). *Compendium of Rose Diseases and Pests (2nd ed.)* American Phytopathological Society Press.
- ISDI. (2024). Redes Neuronales Convolucionales. <https://www.isdi.education/es/blog/redes-neuronales-convolucionales>
- James Le. (2018, abril). *The 5 Computer Vision Techniques That Will Change How You See the World*. <https://heartbeat.comet.ml/the-5-computer-vision-techniques-that-will-change-how-you-see-the-world-1ee19334354b>
- Masso Garden. (2024). *Plagas y enfermedades*. <https://www.massogarden.com/es/plagas-y-enfermedades/enfermedades/roya-2>
- Metroflor. (2024). *Evolución y desafíos de la floricultura ecuatoriana en el futuro próximo*. <https://www.metroflorcolombia.com/evolucion-y-desafios-de-la-floricultura-ecuatoriana-en-el-futuro-proximo/>
- Ministerio de Agricultura Ganadería y Alimentación de Guatemala. (2024, octubre). *Aprenda aspectos básicos del cultivo de rosas*. <https://www.maga.gob.gt/aprenda-aspectos-basicos-del-cultivo-de-rosas/>
- Ministerio de Agricultura, Pesca y Alimentación de España. (2025). *Phragmidium mucronatum (Pers.) Schlecht*. [https://www.mapa.gob.es/ministerio/pags/plataforma\\_conocimiento/fichas/pdf/fd\\_190.pdf](https://www.mapa.gob.es/ministerio/pags/plataforma_conocimiento/fichas/pdf/fd_190.pdf)

- Noguer, A., & Antiga, J. (2025). Conocimientos básicos sobre visión artificial. [https://www.infoplc.net/files/documentacion/vision\\_artificial/infoPLC\\_net\\_Conocimientos\\_vISIONARTIFICIAL.pdf](https://www.infoplc.net/files/documentacion/vision_artificial/infoPLC_net_Conocimientos_vISIONARTIFICIAL.pdf)
- Pardo, C. J. (2021). *Introducción a la visión artificial con Python*. Universidad Surcolombiana. <https://www.usco.edu.co/archivosUsuarios/19/publicacion/facultad-de-ingenieria/CURSO%20INTRODUCCION%20A%20LA%20VISION%20ARTIFICIAL%20CON%20PYTHON%20-%20NIVEL%20I.pdf>
- Pataky, N. R. (2021). Common rust of roses. *University of Illinois*. <https://ipm.illinois.edu/diseases/rpds/630.pdf>
- Pino-Vargas, E. (2019). Los drones una herramienta para una agricultura eficiente: un futuro de alta tecnología. *Idesia (Arica)*, 37. [https://www.researchgate.net/publication/333912495\\_Los\\_drones\\_una\\_herramienta\\_para\\_una\\_agricultura\\_eficiente\\_un\\_futuro\\_de\\_alta\\_tecnologia](https://www.researchgate.net/publication/333912495_Los_drones_una_herramienta_para_una_agricultura_eficiente_un_futuro_de_alta_tecnologia)
- Purwono, I., Ma'arif, A., Rahmani, W., Imam, H., Fathurrahman, H. I. K., Frisky, A., & Haq, Q. M. U. (2023). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control Systems*, 2, 739-748. [https://www.researchgate.net/publication/367157330\\_Understanding\\_of\\_Convolutional\\_Neural\\_Network\\_CNN\\_A\\_Review](https://www.researchgate.net/publication/367157330_Understanding_of_Convolutional_Neural_Network_CNN_A_Review)
- Qamar, R., & Zardari, B. (2023). Artificial Neural Networks: An Overview. *Mesopotamian Journal of Computer Science*, 2023, 130-139. <https://doi.org/10.58496/MJCSC/2023/015>
- ResearchGate. (2023). *Improving Flask Implementation Using Hardware Assisted In-VM Isolation: The Flask architecture* [Consultado el 27 de noviembre de 2025]. ResearchGate. [https://www.researchgate.net/figure/The-Flask-architecture\\_fig1\\_290715751](https://www.researchgate.net/figure/The-Flask-architecture_fig1_290715751)
- ResearchGate. (2024). *Cleo: Smart Glasses to Monitor Consumption of Alcohol and Cigarettes: YOLOv5 architecture and workflow* [Consultado el 27 de noviembre de 2025]. ResearchGate. [https://www.researchgate.net/figure/YOLOv5-architecture-and-workflow\\_fig4\\_365345891](https://www.researchgate.net/figure/YOLOv5-architecture-and-workflow_fig4_365345891)
- ResearchGate. (2025). *Towards understanding bugs in Python interpreters: The General Architecture of Python* [Consultado el 27 de noviembre de 2025]. ResearchGate. [https://www.researchgate.net/figure/The-General-Architecture-of-Python\\_fig1\\_366233366](https://www.researchgate.net/figure/The-General-Architecture-of-Python_fig1_366233366)
- Rodolfo Ríos Hernández. (2021). *La Agricultura de Precisión. Una necesidad actual*. <https://www.redalyc.org/journal/5862/586269368010/html/>
- Rubén Wainschenker, P. T., José María Massa. (2011). *Procesamiento Digital de Imágenes - Clase Teórico Práctica N° 1*. <https://users.exa.unicen.edu.ar/catedras/pdi/FILES/TE/CP1.pdf?>
- Santana Mayorga, R. C. (2020). *Evaluación de métodos de extracción y dosis de aplicación de cola de caballo (Equisetum arvense) para el control ecológico de roya (Puccinia sp.) en el cultivo de cebolla blanca (Allium fistulosum)*. Universidad Técnica de Ambato. Facultad de Ciencias Agropecuarias. <https://repositorio.uta.edu.ec/bitstreams/825e831e-43e3-4e70-bc9f-a5e54412784c/download>
- Stehr, N. (2015). Drones: The Newest Technology for Precision Agriculture. *Natural Sciences Education*, 44, 89. [https://www.researchgate.net/publication/282447601\\_Drones\\_The\\_Newest\\_Technology\\_for\\_Precision\\_Agriculture](https://www.researchgate.net/publication/282447601_Drones_The_Newest_Technology_for_Precision_Agriculture)

- Steve Ramsay. (2021). *Rose Production Guidelines* [Consultado en línea]. West Grove, PA, USA. [https://starrosesandplants.com/app/uploads/2021/03/2021\\_RoseGrowersManual\\_English.pdf](https://starrosesandplants.com/app/uploads/2021/03/2021_RoseGrowersManual_English.pdf)
- Technologies, R. (2025). *MySQL*. Rays Technologies. <https://www.raystec.com/courses/database/mysql.html>
- Tello Marulanda, T. (2023). *Técnicas de visión artificial y aprendizaje de máquinas para la detección y clasificación de grietas*. Revista Ingeniería y Competitividad. [https://revistaingenieria.univalle.edu.co/index.php/ingenieria\\_y\\_competitividad/article/view/12143/15899](https://revistaingenieria.univalle.edu.co/index.php/ingenieria_y_competitividad/article/view/12143/15899)
- UNIR Mexico. (2024). *Redes Neuronales Artificiales*. <https://mexico.unir.net/noticias/ingenieria/redes-neuronales-artificiales/>
- Vásquez Tubón, V. (2013). *Control de trips (Frankliniella occidentales) mediante la aplicación de tres extractos botánicos en el cultivo de rosas (Rosa sp.) variedad Mohana*. Universidad Central del Ecuador. <http://www.dspace.uce.edu.ec/handle/25000/1104>
- Vives Garnique, L. A., Mejía Cabrera, H. I., Vilcherrez Chavarry, K. L., & Vassallo Barco, M. J. (2014). Visión artificial: Aplicación de filtros y segmentación en imágenes de hojas de café. *Revista Ingeniería: Ciencia, Tecnología e Innovación*. <https://revistas.uss.edu.pe/index.php/ING/article/download/198/209/778>
- Windham, A. (2021). *Rose Diseases: Identification and Management* (UT Extension Publication W833). University of Tennessee Institute of Agriculture. <https://plantsciences.tennessee.edu/wp-content/uploads/sites/25/2021/11/UT-Extension-Rose-diseases-Identification-and-management-W833.pdf>
- Yong, A. (2004). El cultivo del rosal y su propagación. *Cultivos Tropicales*, 25(2), 53-67. <https://www.redalyc.org/pdf/1932/193217832008.pdf>

## ANEXOS

### Anexo 1. Entrevista dirigida al propietario empresa florícola DAFRE

Sr. Freddy Castillo Propietario y encargado, empresa florícola DAFRE – Cultivo de rosas

“En los cultivos de rosas, la presencia de la roya es un problema constante, especialmente debido a la humedad y a las condiciones climáticas propias del invernadero. Esta enfermedad aparece principalmente en las hojas, generando manchas anaranjadas que, si no se detectan a tiempo, provocan la disminución de la producción. Actualmente, la detección se realiza de forma manual mediante inspecciones visuales, lo cual requiere bastante tiempo y personal, además de que no siempre se logra identificar la enfermedad en sus primeras etapas.

Para nosotros como productores, contar con un sistema tecnológico que permita detectar la roya mediante un dron resulta totalmente viable y necesario, ya que permitiría identificar las zonas afectadas con mayor precisión. Este tipo de sistema contribuiría significativamente a mejorar la productividad del cultivo.”

### Anexo 2. Encuesta dirigida a los trabajadores de la empresa de DAFRE

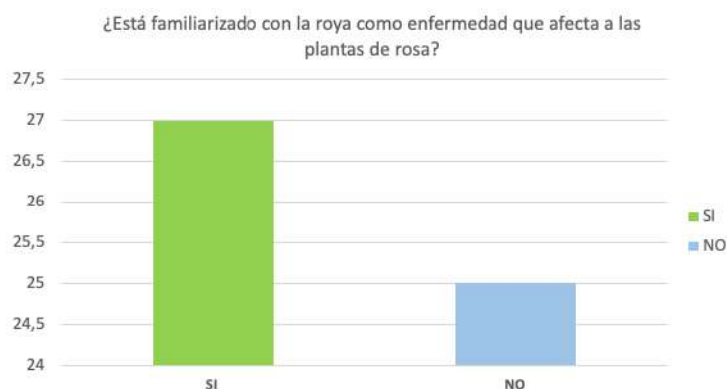
La roya es una enfermedad que afecta a las plantas de rosa, causando manchas en las hojas y debilitando las plantas. Detectar esta enfermedad a tiempo es crucial para evitar que se propague y cause grandes daños en los cultivos. Este prototipo está diseñado para utilizar un dron para detectar la roya en los cultivos de rosales, mejorando la eficiencia y precisión del monitoreo. Anexo 1.

1. ¿Está familiarizado con la roya como enfermedad que afecta a las plantas de rosa?  
 Sí       No
2. ¿Considera importante detectar la roya para evitar que se propague y dañe el cultivo?  
 Muy importante       Algo importante       No importante
3. ¿Cree que el uso de un dron podría ser útil para detectar la roya de manera más rápida y precisa?  
 Sí       No
4. ¿Considera que el uso de un dron puede facilitar el monitoreo de grandes áreas sin recorrer todo el terreno manualmente?  
 Sí       No
5. ¿Le parece importante que el prototipo del dron sea pequeño y resistente para usarse al aire libre o en invernaderos?

- Muy importante     Algo importante     No importante
6. ¿Cree que el dron debería monitorear el cultivo de forma continua, sin necesidad de ajustes constantes?  
 Sí             No
7. ¿Considera que el prototipo debe ser fácil de operar, incluso para personas sin conocimientos técnicos?  
 Sí             No
8. ¿Es importante poder ver en tiempo real las imágenes del dron y los resultados obtenidos?  
 Muy importante     Algo importante     No importante
9. ¿Cree que el sistema debería enviar alertas automáticas al detectar roya?  
 Sí             No
10. ¿Cree que el dron debería activarse automáticamente al detectar roya?  
 Sí             No
11. ¿Sería útil que el sistema guarde los resultados en una base de datos para futuras consultas?  
 Muy útil     Algo útil     No útil
12. ¿Piensa que debería existir un botón para reiniciar el prototipo en caso de fallos?  
 Sí             No
13. ¿Cree que el prototipo debe ser compatible con otros dispositivos como cámaras o pantallas, y tener una fuente de energía confiable?  
 Sí             No
14. ¿Debería el sistema permitir consultar un historial de imágenes registradas de las plantas afectadas por roya?  
 Sí             No
15. ¿Le gustaría que el sistema ofrezca recomendaciones automáticas sobre el tratamiento de las plantas afectadas?  
 Sí             No

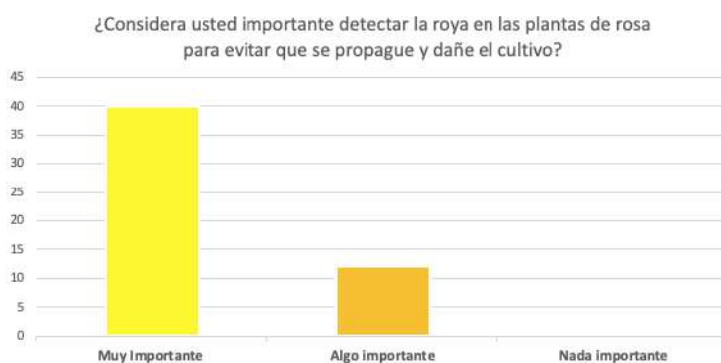
### **Anexo 3. Resultados de la encuesta realizada a los trabajadores de DAFRE**

**Pregunta 1:** En la primera pregunta se observó que el 52 % indicó que sí está familiarizado con la roya como enfermedad que afecta a las plantas de rosa, lo que evidencia que más de la mitad tiene cierto conocimiento sobre este problema presente en los cultivos. Sin embargo, el 48 % manifestó no conocerla, lo que refleja que casi la mitad de los encuestados aún desconoce esta enfermedad y sus posibles consecuencias.

**Figura 100.***Tabulación de la pregunta 1*

**Nota.** Resultados sobre la familiaridad con la roya como enfermedad. Elaborado por el autor.

**Pregunta 2:** Los resultados para la segunda pregunta muestran que la gran mayoría de los encuestados reconoce la importancia de detectar a tiempo la roya. El 77 % considera que esta acción es muy importante, ya que entienden que identificar la enfermedad de forma temprana ayuda a evitar su propagación y a proteger el cultivo. Mientras tanto, el 23 % opina que es algo importante, lo que indica que, aunque valoran su detección, no todos perciben el riesgo con la misma intensidad. Es importante resaltar que ninguno de los participantes considera que esta acción sea irrelevante, lo que demuestra una preocupación general por el cuidado y la salud del rosal.

**Figura 101.***Tabulación de la pregunta 2*

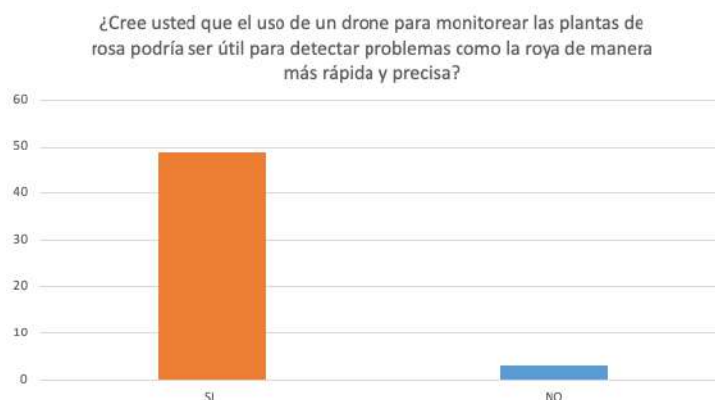
**Nota.** Resultados sobre la importancia de detectar la roya para evitar su propagación en el cultivo. Elaborado por el autor.

**Pregunta 3:** Para la siguiente pregunta se analizó que existe una percepción muy positiva sobre el uso de drones en el monitoreo de los cultivos de rosas con un 94 % de aprobación. La gran mayoría de los encuestados considera que esta tecnología permitiría identificar problemas como la roya de manera más rápida y precisa,

lo que refleja una clara apertura hacia soluciones innovadoras que faciliten el cuidado del cultivo y mejoren su control. Solo un pequeño grupo con el 6% manifestó que no lo considera útil, lo cual representa una minoría frente a la opinión general.

### Figura 102.

Tabulación de la pregunta 3

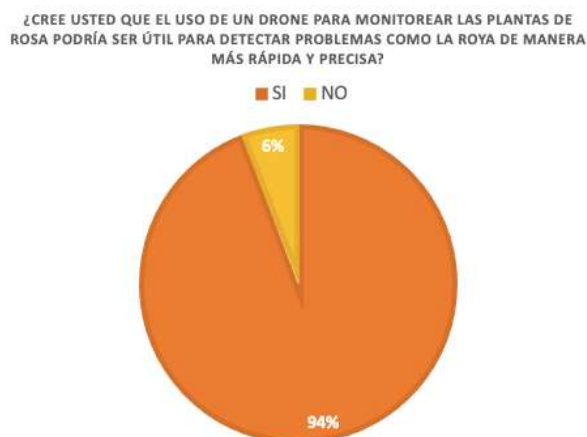


**Nota.** Resultados sobre la utilidad del dron para detectar problemas como la roya. Elaborado por el autor.

**Pregunta 4:** Para esta pregunta el 94% de los encuestados manifestó que esta tecnología sí facilita la supervisión de las plantas de rosa, destacando su utilidad para reducir el esfuerzo físico y el tiempo requerido en las inspecciones tradicionales realizadas de forma manual. Este alto porcentaje refleja que los productores reconocen el valor del dron como un recurso que optimiza las labores de control, permitiendo una cobertura más amplia y eficiente del terreno. Por otra parte, únicamente el 6% considera que su uso no representa una mejora significativa, lo que corresponde a una parte mínima dentro del total de participantes.

### Figura 103.

Tabulación de la pregunta 4



**Nota.** Resultados sobre la utilidad del dron para detectar roya de forma rápida y precisa. Elaborado por el autor.

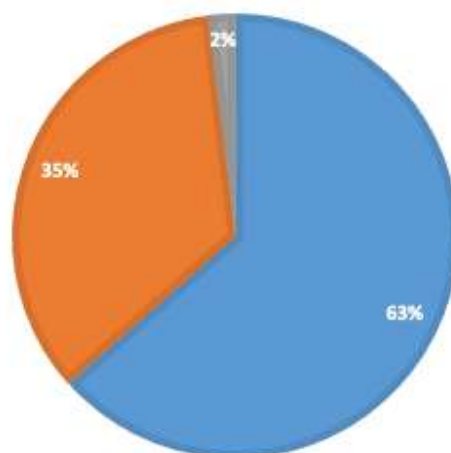
**Pregunta 5:** Con el fin de garantizar un monitoreo eficiente sin comprometer la integridad del cultivo de rosas, los encuestados manifestaron la importancia de que el prototipo del dron sea pequeño y resistente para su uso tanto en espacios abiertos como en invernaderos. En este sentido, el 63 % de la muestra considera que esta característica es muy importante, ya que permitiría un manejo más seguro y preciso dentro del área de cultivo. Por su parte, el 35 % indicó que es algo importante, reconociendo su utilidad aunque con menor nivel de prioridad. Finalmente, solo el 2 % señaló que esta condición no es relevante, lo que evidencia que la gran mayoría de los participantes valora el diseño compacto y resistente.

**Figura 104.**

*Tabulación de la pregunta 5*

¿LE PARECE IMPORTANTE QUE EL PROTOTIPO DEL DRONE SEA PEQUEÑO Y RESISTENTE PARA PODER USARSE AL AIRE LIBRE O EN INVERNADEROS?

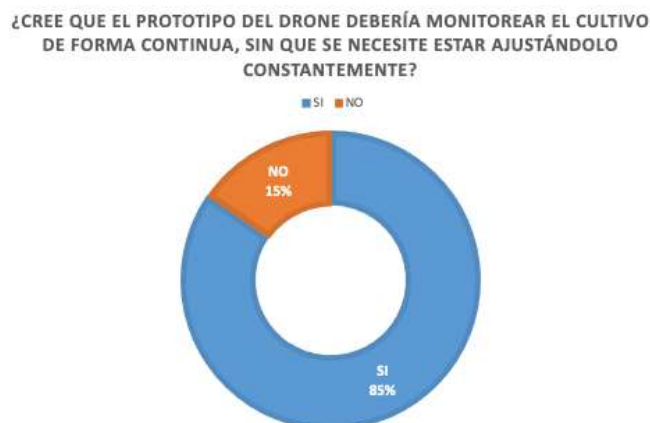
■ Muy importante ■ Algo importante ■ Nada importante



**Nota.** Resultados sobre el tamaño y resistencia del dron para uso en exterior e invernaderos. Elaborado por el autor.

**Pregunta 6:** La mayoría de los encuestados, que representa el 85 %, opina que el dron debería trabajar de forma continua sin tener que estar ajustándolo a cada momento. Esto demuestra que prefieren una herramienta que sea práctica y que les facilite el trabajo diario, permitiéndoles ahorrar tiempo y esfuerzo al momento de monitorear el cultivo.

Solo el 15 % considera que no es tan necesario que funcione de esta manera, pero es una parte pequeña frente a quienes sí ven ventaja en que el dron sea más autónomo. En general, se nota una buena aceptación hacia un sistema que haga el trabajo más sencillo y eficiente.

**Figura 105.***Tabulación de la pregunta 6*

**Nota.** Resultados sobre el monitoreo continuo del cultivo mediante el dron. Elaborado por el autor.

**Pregunta 7:** Al analizar las respuestas, se observa que una gran parte de los participantes cree que el dron debería ser fácil de manejar, incluso para personas que no cuentan con conocimientos técnicos. El 83 % considera que la simplicidad en su uso es un aspecto importante, ya que permitiría que cualquier trabajador del cultivo pueda operarlo sin complicaciones ni necesidad de capacitación avanzada. En cambio, el 17 % no lo considera tan necesario, aunque esta opinión representa un grupo reducido.

**Figura 106.***Tabulación de la pregunta 7*

**Nota.** Resultados sobre la facilidad de uso del prototipo. Elaborado por el autor.

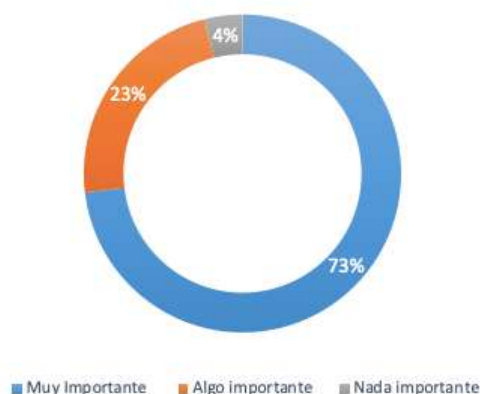
**Pregunta 8:** Los datos obtenidos muestran que la visualización en tiempo real de las imágenes captadas por el dron es una función ampliamente valorada por los productores. Para el 73 % de los encuestados, esta característica resulta fundamental, ya que les permitiría observar de manera inmediata el estado del cultivo. Un 23 %

considera que esta función tiene una importancia moderada, mientras que apenas el 4 % no la percibe como necesaria. Esto evidencia que la transmisión en tiempo real aporta un mayor control, seguridad y eficiencia en el monitoreo de los rosales.

**Figura 107.**

*Tabulación de la pregunta 8*

¿Es importante para usted poder ver en tiempo real las imágenes del dron?



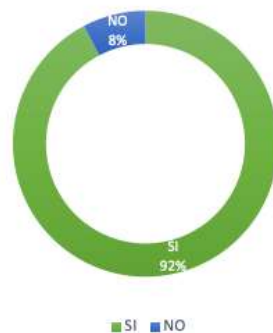
**Nota.** Resultados de sobre la importancia de visualizar en tiempo real las imágenes del dron. Elaborado por el autor.

**Pregunta 9:** Se consultó si el prototipo del dron debía mostrar las zonas con presencia de roya, obteniendo que el 92 % considera esta función necesaria para una intervención rápida y precisa, mientras que el 8 % no la considera imprescindible.

**Figura 108.**

*Tabulación de la pregunta 9*

¿Considera que el prototipo debería mostrar en su interfaz las zonas con presencia de roya detectadas al procesar las imágenes del cultivo?



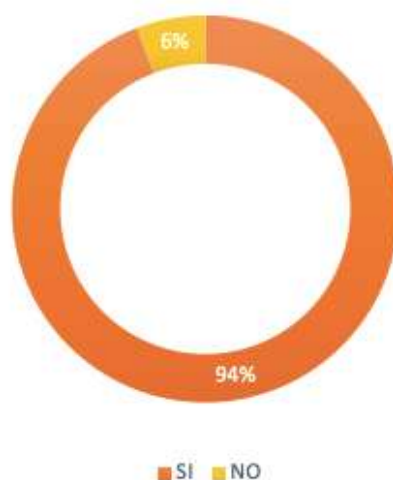
**Nota.** Resultados sobre la visualización de zonas con presencia de roya en la interfaz. Elaborado por el autor.

**Pregunta 10:** Para los productores, resulta mucho más cómodo que el prototipo analice las imágenes de forma automática apenas sean cargadas, sin tener que realizar pasos adicionales. El 94 % de los encuestados considera que esta función facilitaría su trabajo y haría el proceso más rápido y sencillo, permitiéndoles enfocarse en otras tareas del cultivo sin complicaciones técnicas. Por otro lado, solo el 6 % no lo ve como algo necesario, aunque representa una parte mínima frente a la opinión general. En conjunto, se evidencia que los participantes prefieren un sistema práctico que trabaje de forma automática y reduzca el esfuerzo del usuario durante el monitoreo del cultivo.

**Figura 109.**

*Tabulación de la pregunta 10*

**¿Le gustaría que el prototipo analice las imágenes automáticamente apenas sean cargadas, sin requerir pasos adicionales?**



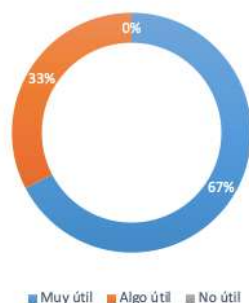
**Nota.** Resultados sobre el análisis automático de las imágenes al cargarse. Elaborado por el autor.

**Pregunta 11:** En el desarrollo de la encuesta se solicitó al propietario y trabajadores indicar si consideran útil que el prototipo guarde los resultados de las detecciones de roya en una base de datos para que puedan ser revisados en cualquier momento. Los resultados muestran que el 67 % de la muestra considera que esta función sería muy útil, ya que permitiría llevar un mejor control y seguimiento del estado del cultivo, mientras que el 33 % de la muestra señala que sería algo útil.

Cabe destacar que ningún encuestado considera que esta función sea innecesaria. Como se observa en el siguiente gráfico estadístico.

**Figura 110.***Tabulación de la pregunta 11*

¿Sería útil que el prototipo guarde los resultados de las detecciones de roya en una base de datos, para que puedan ser revisados en cualquier momento?

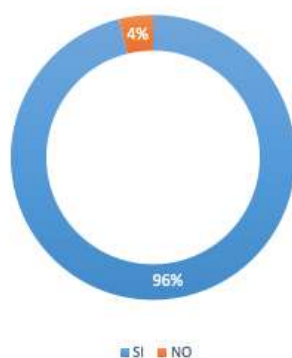


**Nota.** Resultados sobre la utilidad de almacenar las detecciones en una base de datos. Elaborado por el autor.

**Pregunta 12:** En cuanto a si el prototipo debería contar con una opción para reiniciarlo en caso de que algo no funcione correctamente, los resultados muestran que el 96 % de la muestra considera necesaria esta función, ya que permitiría solucionar posibles fallos de manera rápida y continuar con el monitoreo sin mayores inconvenientes, mientras que únicamente el 4 % de la muestra indica que no lo considera imprescindible.

**Figura 111.***Tabulación de la pregunta 12*

¿Piensa que debería haber una opción en el prototipo para reiniciarlo en caso de que algo no funcione bien?



**Nota.** Resultados sobre la opción de reinicio del prototipo. Elaborado por el autor.

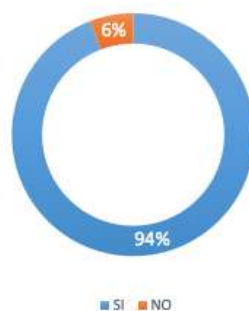
**Pregunta 13:** Los encuestados afirman en su mayoría que la aplicación web del prototipo debe ser compatible con distintos dispositivos como teléfonos móviles, tablets, laptops y computadoras de escritorio. Los resultados evidencian que el 94 % de la muestra considera necesaria esta compatibilidad, ya que permitiría acceder al

sistema desde diferentes equipos de forma práctica y flexible, mientras que únicamente el 6% de la muestra no lo considera importante.

**Figura 112.**

*Tabulación de la pregunta 13*

¿Cree que la aplicación web del prototipo debería ser compatible con distintos dispositivos, como teléfonos móviles, tablets, laptops y computadoras de escritorio?



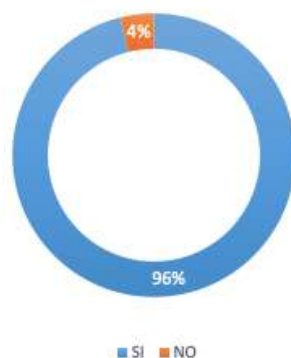
**Nota.** Resultados sobre compatibilidad multiplataforma de la aplicación web. Elaborado por el autor.

**Pregunta 14:** La siguiente pregunta de la encuesta propone la posibilidad de consultar un historial de imágenes registradas de las plantas afectadas por roya, esto indica que la gran mayoría está de acuerdo con esta función. Los resultados reflejan que el 96% de la muestra considera necesario que el sistema permita visualizar este historial, ya que facilitaría el seguimiento y control de la enfermedad a lo largo del tiempo, mientras que únicamente el 4% no lo considera imprescindible. Como se aprecia en el gráfico, existe una aceptación casi total hacia la inclusión de esta característica dentro del prototipo.

**Figura 113.**

*Tabulación de la pregunta 14*

¿Cree que el sistema debería permitir consultar un historial de imágenes registradas de las plantas afectadas por roya?



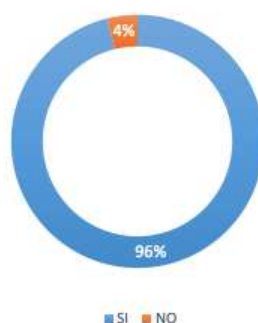
**Nota.** Resultados sobre el historial de imágenes del sistema. Elaborado por el autor.

**Pregunta 15:** Finalmente se preguntó sobre la inclusión de recomendaciones automáticas por parte del prototipo para el tratamiento o cuidado de las plantas afectadas con roya, en el que se obtuvo que la gran mayoría de los encuestados considera esta función como necesaria. Los resultados muestran que el 96 % de la muestra está de acuerdo con que el sistema ofrezca este tipo de sugerencias, mientras que únicamente el 4 % no lo considera indispensable. Como vemos en el gráfico a continuación, existe una clara preferencia por un prototipo que no solo detecte la enfermedad, sino que también brinde orientación para su adecuado manejo. 114

**Figura 114.**

*Tabulación de la pregunta 15*

¿Le gustaría que el prototipo ofrezca recomendaciones automáticas sobre el tratamiento o cuidado de las plantas detectadas con roya?

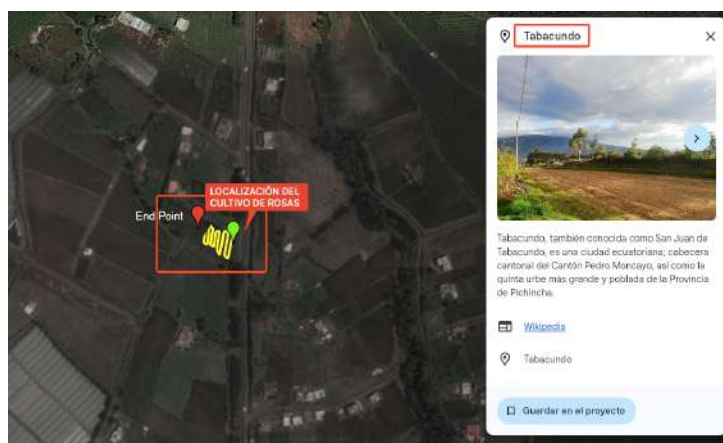


**Nota.** Resultados sobre recomendaciones automáticas del prototipo. Elaborado por el autor.

#### **Anexo 4. Localización del campo de cultivo de rosas de la empresa DAFRE**

**Figura 115.**

*Localización del campo del cultivo de rosas*



**Nota.** Ubicación geográfica del área donde se encuentra el invernadero de rosas en Tabacundo. Elaborado por el autor.

## Anexo 5. Tabla de verificación de los requerimientos

**Tabla 38.**

Tabla de verificación de los requerimientos de Stakeholders

#	Requerimiento	Sí	No	N/A	Observación
<b>Requerimientos Operacionales</b>					
STSR1	El sistema de detección debe ser implementado dentro de un invernadero.	X	<input type="checkbox"/>	<input type="checkbox"/>	
STSR2	El dron debe ser compacto y resistente para ambientes de exterior.	X	<input type="checkbox"/>	<input type="checkbox"/>	
STSR3	El sistema debe disponer de una base de datos estructurada para almacenar y gestionar las imágenes capturadas de las rosas durante el proceso de detección.	X	<input type="checkbox"/>	<input type="checkbox"/>	
STSR4	El plan de vuelo debe cubrir toda el área del cultivo.	X	<input type="checkbox"/>	<input type="checkbox"/>	
STSR5	El sistema debe ser fácil de integrar en el entorno de cultivo.	X	<input type="checkbox"/>	<input type="checkbox"/>	
STSR6	El sistema debe asegurar la integridad y respaldo de los datos recolectados.	X	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Requerimientos de Usuario</b>					
STSR7	El sistema debe ser fácil de usar por el agricultor o técnico.	X	<input type="checkbox"/>	<input type="checkbox"/>	
STSR8	Visualización de los resultados obtenidos durante la detección de imágenes de las rosas y la identificación de roya.	X	<input type="checkbox"/>	<input type="checkbox"/>	
STSR9	Reporte del estado del cultivo que pueda visualizar el agricultor.	X	<input type="checkbox"/>	<input type="checkbox"/>	
STSR10	El sistema debe ofrecer recomendaciones sobre el tratamiento de las plantas.	X	<input type="checkbox"/>	<input type="checkbox"/>	
STSR11	El sistema debe permitir el acceso a la información de los cultivos en cualquier momento.	X	<input type="checkbox"/>	<input type="checkbox"/>	
STSR12	La plataforma debe ser accesible desde dispositivos móviles.	X	<input type="checkbox"/>	<input type="checkbox"/>	

**Nota.** La tabla muestra la verificación de los requerimientos definidos por los stakeholders, organizados según su categoría operacional y de usuario. Elaborado por el autor.

**Tabla 39.**

Tabla de verificación de los requerimientos del sistema (SySR)

#	Requerimiento	Sí	No	N/A	Observación
<b>Requerimientos de uso</b>					
SySR1	El dispositivo debe estar encendido para llevar a cabo el proceso de detección.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR2	El algoritmo de detección debe ejecutarse al iniciar el plan de vuelo en el cultivo.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR3	El sistema guarda automáticamente los resultados de la detección en la base de datos.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR4	El sistema debe contar con una interfaz gráfica para iniciar la detección.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR5	El usuario podrá reiniciar el proceso de detección desde la aplicación web.	X	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Requerimientos de performance</b>					
SySR6	El sistema debe soportar la detección simultánea.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR7	El tiempo de respuesta del sistema de detección debe ser menor a 10 segundos.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR8	El dron debe tener un punto de inicio de fácil acceso.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR9	El sistema debe garantizar una velocidad de procesamiento estable.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR10	El sistema debe ser capaz de funcionar sin interrupciones durante al menos 8 horas.	X	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Requerimientos de interfaces</b>					
SySR11	La tarjeta de almacenamiento debe tener una capacidad mínima de 64 GB.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR12	El sistema debe ser compatible con múltiples tipos de cámaras.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR13	La base de datos debe recopilar y almacenar los datos en un formato adecuado.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR14	Conectividad a tecnología de comunicación inalámbrica.	X	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Requerimientos de estados</b>					
SySR15	La placa de procesamiento debe estar encendida para iniciar el sistema.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR16	La cámara del dron debe capturar imágenes con la resolución y calidad necesarias para identificar signos de roya en el cultivo.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR17	El sistema es capaz de indicar cuando la detección haya finalizado.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR18	El sistema debe indicar el estado de procesamiento de las imágenes durante cada fase de detección.	X	<input type="checkbox"/>	<input type="checkbox"/>	
<b>Requerimientos físicos</b>					
SySR19	El dispositivo debe ser portable para su uso en distintas zonas del cultivo.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR20	Se debe contar con un área de cultivo accesible para realizar la detección.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR21	El dron debe operar bajo condiciones ambientales adecuadas, debido a su susceptibilidad al agua.	X	<input type="checkbox"/>	<input type="checkbox"/>	
SySR22	El dispositivo debe tener protección contra golpes.	X	<input type="checkbox"/>	<input type="checkbox"/>	

**Nota.** La tabla presenta la verificación de los requerimientos del sistema, evaluados según su cumplimiento durante la fase de validación del prototipo. Elaborado por el autor.

**Tabla 40.**

Tabla de verificación de los requerimientos de arquitectura (SRSH)

#	Requerimiento	Sí	No	N/A	Observación
<b>Requerimientos lógicos</b>					
SRSH1	El sistema de monitoreo debe ser compatible con diferentes sistemas operativos.	X			
SRSH2	El sistema debe ser capaz de procesar las imágenes y videos.	X			
SRSH3	El sistema debe procesar los datos sin necesidad de conexión directa con el dron.	X			
SRSH4	Los datos deben ser subidos de forma manual a la aplicación web.	X			
SRSH5	El sistema debe permitir la exportación de datos en PDF para análisis posterior.	X			
<b>Requerimientos de hardware</b>					
SRSH6	El dron debe tener una cámara con alta resolución para la captura de imágenes.	X			
SRSH7	El sistema debe ser capaz de manejar y procesar grandes volúmenes de datos.	X			
SRSH8	El dron debe tener soporte GPS.	X			
SRSH9	Se debe incluir una batería eficiente para el dron.	X			
SRSH10	El dispositivo debe contar con una tarjeta gráfica dedicada NVIDIA compatible con el sistema de procesamiento y visión artificial.	X			
<b>Requerimientos de software</b>					
SRSH11	El sistema debe ser compatible con sistemas operativos basados en GNU/Linux.	X			
SRSH12	El software debe ser capaz de procesar imágenes capturadas en tiempo real.	X			
SRSH13	Uso eficiente de la memoria RAM para evitar sobrecargar el sistema.	X			
SRSH14	Capacidad de generar reportes para el tratamiento del cultivo.	X			
SRSH15	El software del sistema debe permitir actualizaciones seguras.	X			
<b>Requerimientos eléctricos</b>					
SRSH16	El sistema debe tener una fuente de energía para alimentar el sistema de detección.	X			
SRSH17	El sistema debe tener una fuente de energía de respaldo.	X			
SRSH18	El consumo energético del sistema debe ser eficiente.	X			
SRSH19	El sistema debe contar con un sistema de protección contra sobrecargas eléctricas.	X			
SRSH20	El sistema de monitoreo debe tener un consumo energético bajo.	X			
<b>Requerimientos de diseño</b>					
SRSH21	La estructura del dron debe ser compacta y ligera, con un peso no superior a 2 kg.	X			
SRSH22	El dron debe ser resistente al agua y al polvo para operar en condiciones agrícolas.	X			
SRSH23	El dron debe contar con una estructura robusta que soporte impactos y caídas menores.	X			
SRSH24	El sistema debe permitir un fácil acceso a la interfaz principal.	X			
SRSH26	El sistema de monitoreo debe permitir visualizar la presencia de roya.	X			



**Nota.** La tabla presenta la verificación de los requerimientos de arquitectura del sistema, confirmando su cumplimiento en los aspectos lógicos, de hardware, software, eléctricos y de diseño. Elaborado por el autor.

## Anexo 6. Placa de desarrollo / Dell G15 - Ryzen 7

El Dell G15 con procesador Ryzen 7 es un equipo portátil que ofrece el rendimiento necesario para trabajar con el prototipo de detección de roya, permitiendo procesar imágenes y ejecutar modelos de visión artificial de forma fluida. Su buen desempeño facilita el desarrollo, análisis y gestión del sistema, haciendo que las tareas se realicen de manera más rápida y estable. La Figura 116 muestra las especificaciones técnicas del hardware del equipo.

**Figura 116.**

*Especificaciones técnicas del hardware del equipo de procesamiento (CPU y GPU)*

CPU	Mainboard	Memory	SPD	Graphics	Bench	About
<b>Processor</b>						
Name	AMD Ryzen 7 5800H					
Code Name	Cezanne	Max TDP	45.0 W			
Package	Socket FP6					
Technology	7 nm	Core VID	1.006 V			
<b>Specification</b>						
AMD Ryzen 7 5800H with Radeon Graphics						
Family	F	Model	0	Stepping	0	
Ext. Family	19	Ext. Model	50	Revision	C2N-A0	
Instructions	MMX(+), SSE (1, 2, 3, 4.1, 4.2, 4A), SSSE3, x86-64, AES, AVX, AVX2, FMA3, SHA					
<b>Clocks (Core #0)</b>						
Core Speed	3232.69 MHz					
Multiplier	x 32.4					
Bus Speed	99.77 MHz					
Rated FSB						
<b>Cache</b>						
L1 Data	8 x 32 KBytes		8-way			
L1 Inst.	8 x 32 KBytes		8-way			
Level 2	8 x 512 KBytes		8-way			
Level 3	16 MBytes		16-way			
<b>Clocks</b>						
GFX Core	1695.0 MHz					
Shader / SoC						
Memory	6001.0 MHz					
<b>Display Device Selection</b>						
NVIDIA GeForce RTX 3050 Ti Laptop				Perf Level: Current		
<b>GPU</b>						
Name	NVIDIA GeForce RTX 3050 Ti Laptop GPU					
Board Manuf.	Dell					
Code Name	GA 107	Revision	A1			
Technology	8 nm	TDP	80.0 W			
ROPs / TMUs	32 / 80	Cores	2560			
<b>Memory</b>						
Size	4 GBytes					
Type	GDDR6					
Vendor	Samsung					
Bus Width	128 bits					
<b>Selection</b>						
Socket #1	Cores: 8			Threads: 16		

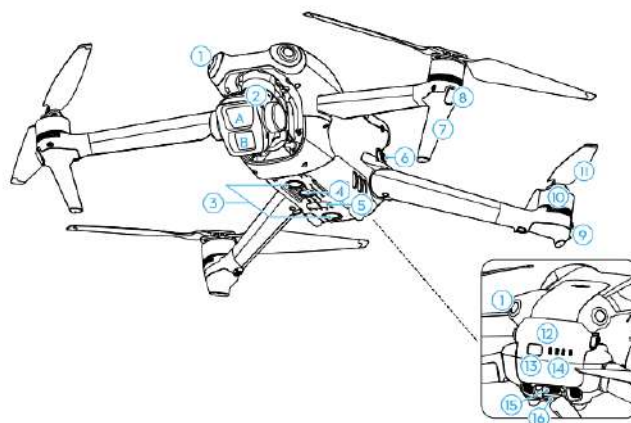
*Nota.* Elaborado por el autor a partir de información obtenida del sistema del equipo

## Anexo 7. Dron DJI Air 3

El dron DJI Air 3 permite realizar el monitoreo del cultivo, ofreciendo un vuelo estable y seguro durante su operación. Su estructura integra diversos sistemas que permiten captar imágenes nítidas y detalladas del estado de las plantas. Gracias a sus sensores y mecanismos de control, el equipo logra desplazarse con precisión sobre el terreno, contribuyendo a un seguimiento más eficiente y ordenado del cultivo sin necesidad de una supervisión directa constante. La Figura 117 presenta el diagrama esquemático del dron, donde se ilustran cada una de sus partes y funcionalidades principales.

**Figura 117.**

*Diagrama esquemático del dron DJI Air 3 y sus componentes principales*



*Nota.* Adaptado del manual técnico oficial del DJI Air 3 (DJI, 2023.)

### **Características**

1. Sistema de visión omnidireccional
2. Estabilizador y cámara
  - a) Telecámara intermedia
  - b) Cámara con gran angular
3. Sistema de visión inferior
4. Luz auxiliar
5. Sistema de detección por infrarrojos tridimensional
6. Bandas de sujeción de la batería
7. Tren de aterrizaje (antenas integradas)
8. LEDs delanteros
9. Indicadores de estado de la aeronave
10. Motores
11. Hélices
12. Batería de vuelo inteligente
13. Botón de encendido
14. LEDs de nivel de batería
15. Puerto USB-C
16. Ranura para tarjeta microSD

### **Anexo 8. Sistema operativo / Ubuntu**

Ubuntu se presenta como el sistema operativo que sostiene todo el funcionamiento del prototipo, brindando un entorno confiable y fácil de manejar. Gracias a su estabilidad, permite que las herramientas de procesamiento y análisis de imágenes trabajen de forma continua sin interrupciones. Además, su compatibilidad con Python y YOLOv5 facilita que el sistema ejecute las tareas de detección de roya de manera fluida, contribuyendo a que el monitoreo del cultivo se realice de forma ordenada, eficiente y segura.

#### **Características:**

- Sistema operativo libre y de código abierto.
- Alta estabilidad y rendimiento en tareas de procesamiento.
- Interfaz gráfica intuitiva y fácil de usar.
- Compatible con múltiples lenguajes de programación y frameworks.
- Soporte para herramientas de visión artificial y machine learning.
- Actualizaciones constantes y alto nivel de seguridad.
- Funciona en equipos de bajo y alto rendimiento.
- Amplia comunidad de soporte y documentación técnica.

## **Anexo 9. Entorno de programación / Visual Code**

Como entorno de programación, Visual Studio Code se emplea como la plataforma principal para el desarrollo y gestión del prototipo de detección de roya. Este editor permite organizar y estructurar el código de forma clara, facilitando la implementación de los algoritmos de visión artificial, la integración del modelo YOLOv5 y la conexión con la base de datos y la aplicación web. Su entorno intuitivo contribuye a una programación más ordenada, permitiendo realizar pruebas, depuración y ajustes del sistema de manera eficiente durante todo el proceso de desarrollo.

### **Características**

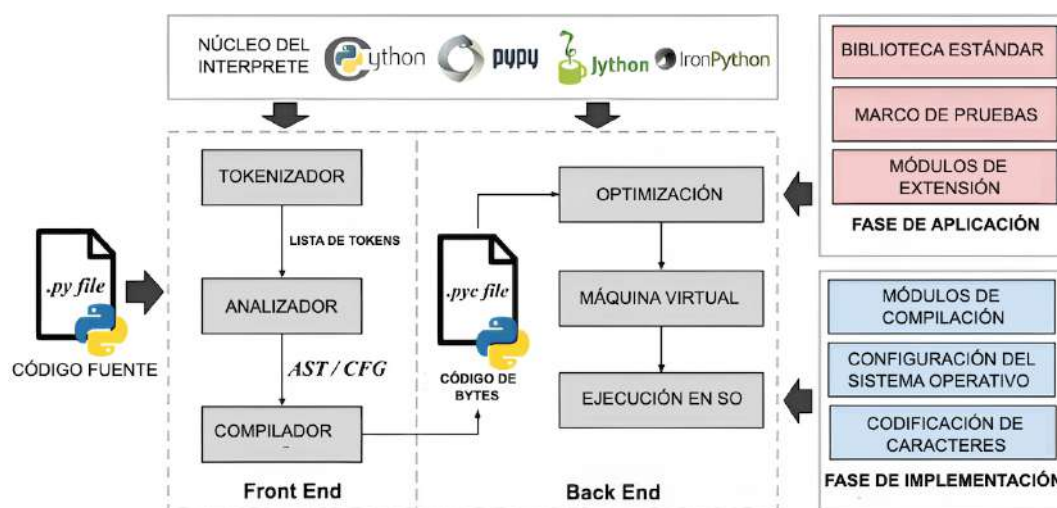
- Editor de código ligero, rápido y de alto rendimiento.
- Compatible con múltiples lenguajes de programación, incluyendo Python.
- Integración directa con entornos virtuales y gestores de paquetes.
- Soporte para extensiones orientadas a inteligencia artificial y desarrollo web.
- Herramientas de depuración y control de errores en tiempo real.
- Interfaz intuitiva que facilita la organización del proyecto.
- Compatible con sistemas operativos como Windows y Ubuntu.
- Permite la gestión de proyectos complejos de forma estructurada.

## Anexo 10. Lenguaje de programación / Python

Python es un lenguaje de programación de alto nivel que permite el desarrollo de soluciones eficientes mediante una estructura sintáctica simple y legible. En el presente prototipo, este lenguaje actúa como eje central del sistema de procesamiento, integrando módulos de visión artificial, análisis automatizado de imágenes y control de procesos. Su arquitectura facilita la conexión entre el modelo de detección YOLOv5 y el entorno operativo, optimizando la ejecución de tareas relacionadas con la identificación de zonas afectadas por roya y el manejo de los datos obtenidos durante el monitoreo del cultivo. Como se muestra en la Figura 118, se evidencia el flujo de procesamiento del intérprete y sus principales componentes funcionales.

**Figura 118.**

*Arquitectura del núcleo del intérprete de Python y sus fases de procesamiento*



*Nota.* Adaptado del esquema de funcionamiento interno del intérprete Python y sus componentes principales (ResearchGate, 2025.)

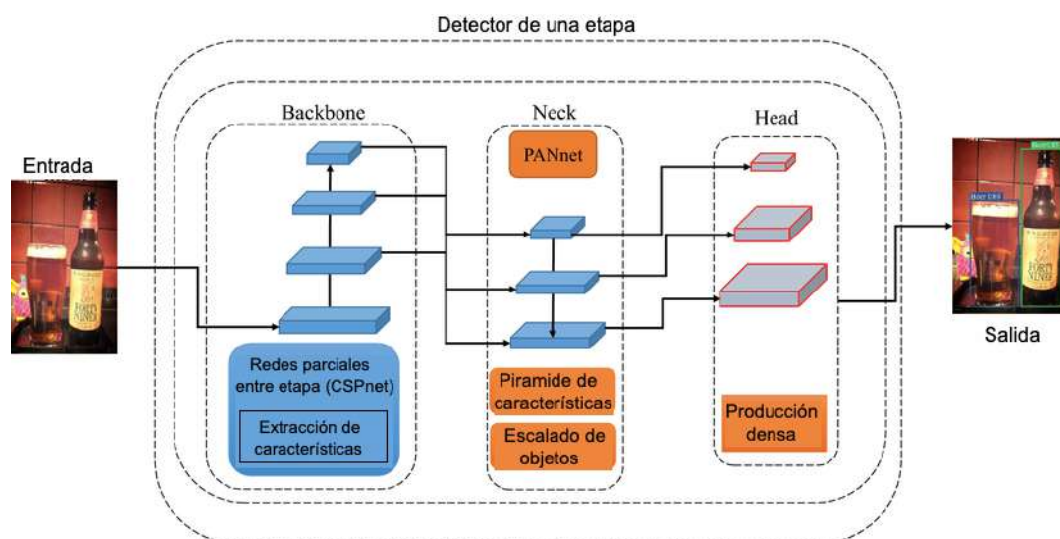
## Anexo 11. Algoritmo de visión artificial / Yolov5

YOLOv5 predice cuadros delimitadores en múltiples escalas, lo que permite una detección más precisa de objetos de diferentes tamaños dentro de una misma imagen. El sistema extrae características relevantes a partir de estas escalas mediante una arquitectura basada en redes piramidales, optimizando la representación de información visual en distintas resoluciones. Desde su extractor principal de características (Backbone), se incorporan capas convolucionales adicionales que fortalecen la identificación de patrones, mientras que el módulo Neck combina estas características mediante PANet para mejorar la localización de los objetos. Finalmente, el Head

genera predicciones que incluyen las coordenadas del cuadro delimitador, el nivel de confianza y la clasificación correspondiente, permitiendo una detección rápida y eficiente de la roya en las hojas de rosa. Este proceso posibilita una detección rápida y precisa, como se aprecia en la Figura 119, evidenciando su eficacia en el análisis automatizado.

**Figura 119.**

*Arquitectura general del modelo YOLOv5 para detección de objetos en una sola etapa*



*Nota.* Adaptado del esquema arquitectónico del modelo YOLOv5 y su flujo de procesamiento (ResearchGate, 2024).

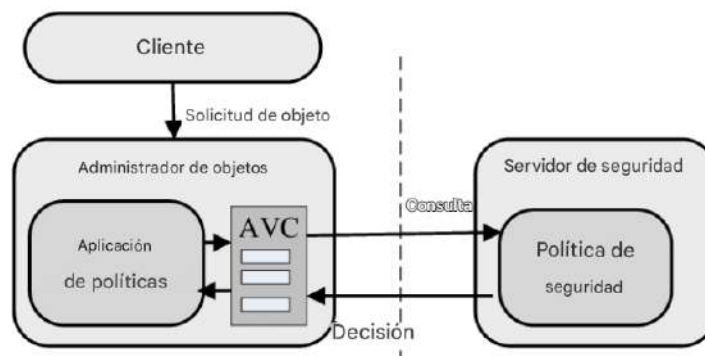
## Anexo 12. Aplicacion Web / Flask

La aplicación web desarrollada con Flask funciona como la interfaz principal mediante la cual el usuario interactúa con el sistema de monitoreo del cultivo. A través de esta plataforma se visualizan los resultados del análisis de roya y el historial de registros almacenados en la base de datos. Su diseño permite una navegación sencilla, facilitando el acceso a la información sin requerir conocimientos técnicos avanzados.

Esta aplicación se encarga además de gestionar las solicitudes del usuario, coordinar el procesamiento de imágenes con el modelo YOLOv5 y presentar los resultados de forma clara y comprensible. El flujo de interacción cliente-servidor y los componentes involucrados se detallan en la Figura 120, evidenciando la organización lógica del sistema y su funcionamiento interno.

**Figura 120.**

*Arquitectura general de la aplicación web desarrollada en Flask*



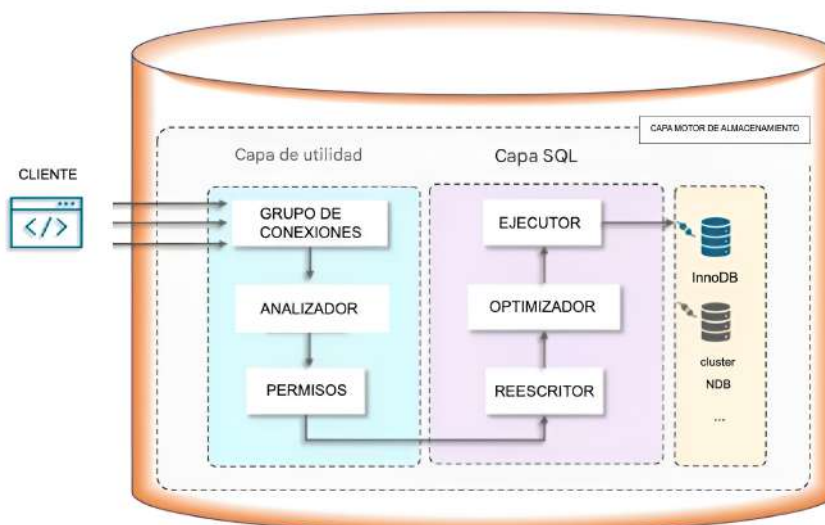
*Nota.* Adaptado del esquema de arquitectura de Flask y su flujo de interacción cliente-servidor ResearchGate, 2023

### Anexo 13. Base de datos / Mysql

MySQL gestiona el almacenamiento y organización de los datos generados por el prototipo, como imágenes procesadas, resultados de detección de roya y registros históricos del cultivo. Su arquitectura interna permite realizar consultas rápidas y seguras, garantizando integridad, orden y disponibilidad de la información, como se observa en la Figura 121.

**Figura 121.**

*Arquitectura general del motor de base de datos MySQL y sus componentes internos*



*Nota.* Adaptado del esquema de funcionamiento del sistema gestor de base de datos MySQL (Technologies, 2025).

## Anexo 14. Aplicación para la planificación de vuelo / Litchi

Litchi es la aplicación seleccionada para utilizarlo como herramienta principal para la planificación y supervisión de los vuelos del dron, permitiendo configurar rutas mediante waypoints y gestionar los distintos modos de vuelo, así como el control de la cámara y la visualización en tiempo real sobre un mapa satelital. La interfaz integra información de telemetría, punto de despegue y posición de la aeronave, lo que facilita la ejecución segura y repetible de misiones de monitoreo del cultivo orientadas a la detección de roya, tal como se muestra en la Figura 122.

**Figura 122.**

*Interfaz principal de la aplicación Litchi y sus elementos de control de vuelo*



*Nota.* Adaptado de la interfaz de planificación de misiones de la aplicación Litchi para drones DJI .

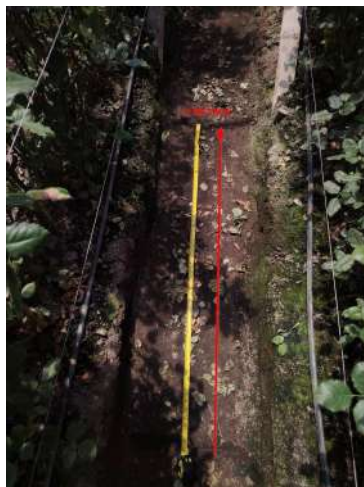
## Anexo 15. Procedimiento para la planificación de vuelo

Para garantizar una planificación de vuelo correcta y precisa, es fundamental mantener una distancia de 2 metros entre cada waypoint. Esto permite evitar el solapamiento en la captura de imágenes y asegura una cobertura uniforme del cultivo.

En la Figura 123 se evidencia el procedimiento realizado para medir y verificar la separación adecuada entre los waypoints dentro del área de cultivo.

**Figura 123.**

*Medición de la distancia entre waypoints en el cultivo*



**Nota.** Medición realizada en campo para verificar la separación de 2 metros entre waypoints. Elaborado por el autor.

Para obtener las coordenadas del lugar se utilizó la aplicación GPS Logger, la cual permite registrar con precisión los valores de latitud y longitud y almacenarlos en un archivo para su posterior uso. La Figura 124 muestra, a detalle, la interfaz de la aplicación y los elementos principales empleados para la captura de la información geográfica.

**Figura 124.**

*Captura de coordenadas con la aplicación GPS Logger*



**Nota.** Registro de coordenadas geográficas mediante la aplicación GPS Logger. Elaborado por el autor.

Una vez realizada la captura de las coordenadas de cada waypoint, se accede a la sección Archivo de la aplicación para descargar el documento que contiene todos los datos registrados durante el recorrido.

La Figura 125 muestra el archivo generado con la información completa de los puntos capturados.

**Figura 125.**

*Archivo generado por la aplicación GPS Logger*



**Nota.** Archivo de coordenadas exportado desde GPS Logger. Elaborado por el autor.

Al abrir el archivo de coordenadas generado por la aplicación GPS Logger, es posible visualizar la estructura y el contenido de cada punto registrado. En la Figura 126 se muestran los nombres asignados a cada waypoint, así como las coordenadas correspondientes a su ubicación exacta dentro del recorrido.

**Figura 126.**

*Visualización de coordenadas exportadas desde GPS Logger*

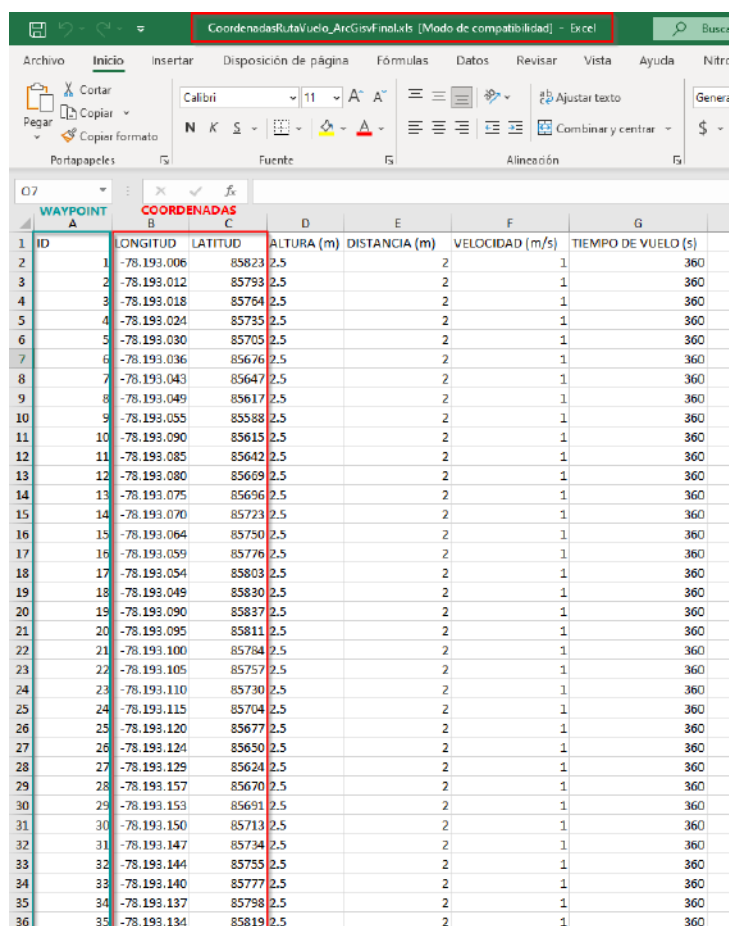


**Nota.** Estructura del archivo KML con los waypoints y sus respectivas coordenadas. Elaborado por el autor.

Las coordenadas recolectadas se organizan en una hoja de cálculo en Excel, asignando un ID a cada waypoint y complementando la información con datos de altura, velocidad, distancia y tiempo de vuelo. Esta estructuración, mostrada en la Figura 127, permite preparar adecuadamente el archivo para su posterior exportación y procesamiento en ArcGIS.

**Figura 127.**

*Organización de coordenadas y parámetros de vuelo en Excel*



WAYPOINT	COORDENADAS					
ID	LONGITUD	LATITUD	ALTURA (m)	DISTANCIA (m)	VELOCIDAD (m/s)	TIEMPO DE VUELO (s)
1	-78.193.006	85823	2.5	2	1	360
2	-78.193.012	85793	2.5	2	1	360
3	-78.193.018	85764	2.5	2	1	360
4	-78.193.024	85735	2.5	2	1	360
5	-78.193.030	85705	2.5	2	1	360
6	-78.193.036	85676	2.5	2	1	360
7	-78.193.043	85647	2.5	2	1	360
8	-78.193.049	85617	2.5	2	1	360
9	-78.193.055	85588	2.5	2	1	360
10	-78.193.090	85615	2.5	2	1	360
11	-78.193.085	85642	2.5	2	1	360
12	-78.193.080	85669	2.5	2	1	360
13	-78.193.075	85696	2.5	2	1	360
14	-78.193.070	85723	2.5	2	1	360
15	-78.193.064	85750	2.5	2	1	360
16	-78.193.059	85776	2.5	2	1	360
17	-78.193.054	85803	2.5	2	1	360
18	-78.193.049	85830	2.5	2	1	360
19	-78.193.090	85837	2.5	2	1	360
20	-78.193.095	85811	2.5	2	1	360
21	-78.193.100	85784	2.5	2	1	360
22	-78.193.105	85757	2.5	2	1	360
23	-78.193.110	85730	2.5	2	1	360
24	-78.193.115	85704	2.5	2	1	360
25	-78.193.120	85677	2.5	2	1	360
26	-78.193.124	85650	2.5	2	1	360
27	-78.193.129	85624	2.5	2	1	360
28	-78.193.157	85670	2.5	2	1	360
29	-78.193.153	85691	2.5	2	1	360
30	-78.193.150	85713	2.5	2	1	360
31	-78.193.147	85734	2.5	2	1	360
32	-78.193.144	85755	2.5	2	1	360
33	-78.193.140	85777	2.5	2	1	360
34	-78.193.137	85798	2.5	2	1	360
35	-78.193.134	85819	2.5	2	1	360

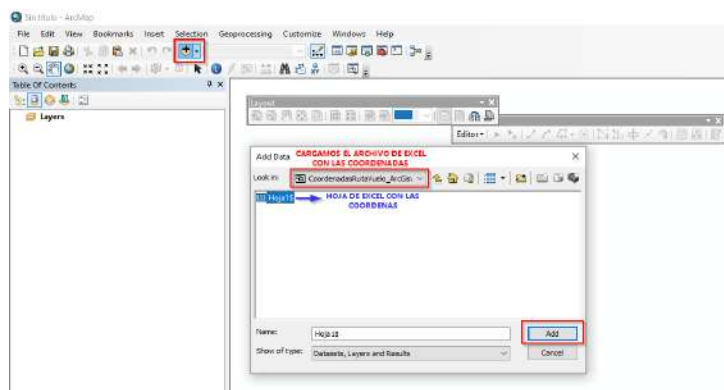
**Nota.** Hoja de cálculo con los waypoints y parámetros de vuelo listos para su exportación a ArcGIS. Elaborado por el autor.

Para realizar el mapeo de los waypoints se empleó el software ArcGIS, específicamente la versión ArcMap 10.8, que permite gestionar y representar de forma precisa las coordenadas geográficas del plan de vuelo, como se muestra en la Figura 128.

**Figura 128.***ArcMap 10.8*

**Nota.** ArcMap 10.8 utilizada para el mapeo de los waypoints. Elaborado por el autor.

Una vez en la interfaz principal de ArcMap, se procede a cargar el archivo de coordenadas elaborado previamente en Excel, utilizando la opción Add Data, tal como se muestra en la Figura 129.

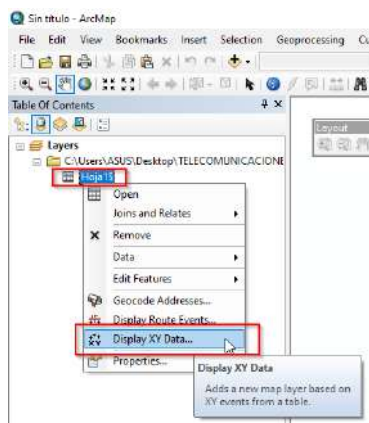
**Figura 129.***Carga del archivo de coordenadas en ArcMap*

**Nota.** Ventana de ArcMap para añadir la hoja de Excel con las coordenadas de los waypoints. Elaborado por el autor.

Al cargar el archivo de coordenadas, la hoja de cálculo se añade como una capa en ArcMap. A continuación, se debe seleccionar dicha hoja y utilizar la opción Display XY Data para representar los puntos en el mapa, como se muestra en la Figura 130.

**Figura 130.**

*Visualización de coordenadas en ArcMap mediante Display XY Data*

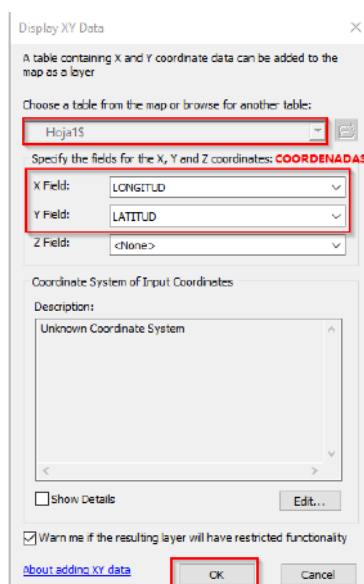


**Nota.** Selección de la hoja de Excel y de la opción Display XY Data para graficar las coordenadas de los waypoints en el mapa. Elaborado por el autor.

En la ventana Display XY Data de ArcMap se reconocen automáticamente los campos de LONGITUD y LATITUD de la hoja de Excel. En este apartado se asignan estos parámetros como coordenadas X e Y para cargar los puntos y generar el mapa correspondiente, como se muestra en la Figura 131.

**Figura 131.**

*Configuración de campos de coordenadas en ArcMap*

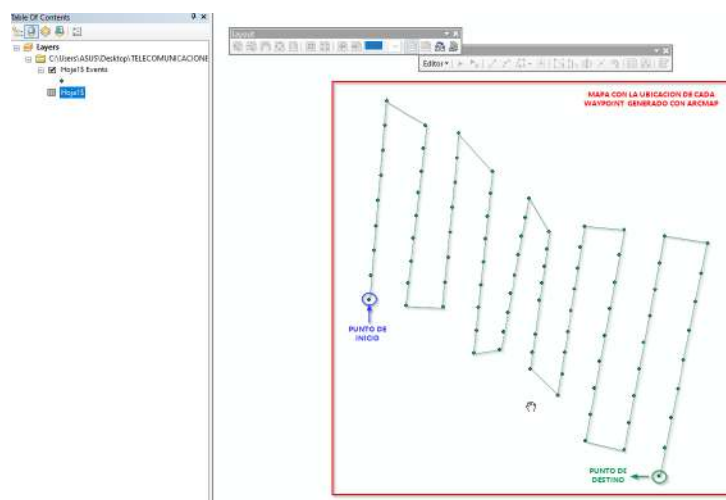


**Nota.** Asignación de los campos de longitud y latitud en la opción Display XY Data para generar el mapa de waypoints. Elaborado por el autor.

Finalmente, ArcMap genera la capa de puntos con las coordenadas de cada waypoint, lo que permite visualizar el trazado completo de la ruta de vuelo y exportar el mapa en formato de archivo o PDF, como se muestra en la Figura 132.

**Figura 132.**

*Ruta de vuelo generada en ArcMap a partir de los waypoints*



**Nota.** Mapa con la ubicación de cada waypoint del plan de vuelo generado en ArcMap. Elaborado por el autor.

## **Anexo 16. Procedimiento de etiquetado con Roboflow**

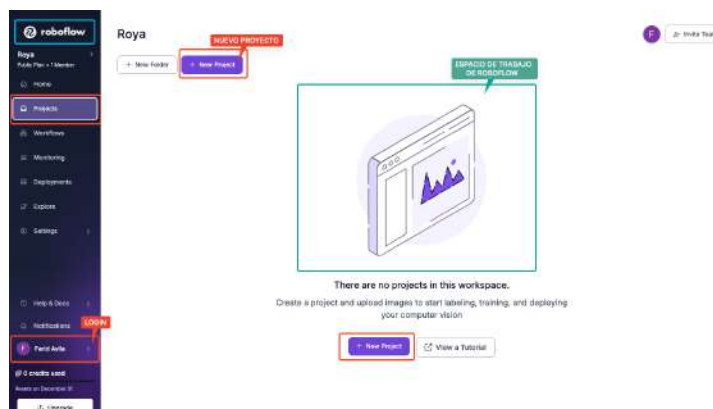
Roboflow es una plataforma en línea diseñada para gestionar datasets de visión por computador, que integra en un mismo entorno la organización de imágenes, el etiquetado, el aumento de datos y la exportación en múltiples formatos compatibles con diferentes modelos de deep learning.

Al estar basada en la nube, permite trabajar de forma centralizada sobre el conjunto de datos, mantener el control de versiones y colaborar con otros usuarios en el mismo proyecto. Gracias a estas funciones, se convierte en una herramienta adecuada para la detección y segmentación de objetos, como la identificación de roya en los cultivos de rosas.

Para iniciar el proceso de etiquetado del dataset es necesario iniciar sesión y, en la sección Projects, crear un nuevo proyecto, tal como se muestra en la Figura 133.

**Figura 133.**

Creación de un nuevo proyecto en Roboflow

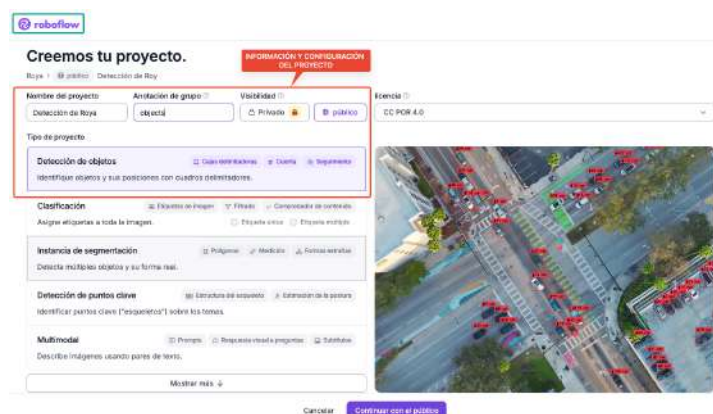


**Nota.** Interfaz de Roboflow para crear un nuevo proyecto. Elaborado por el autor.

Roboflow brinda una alta flexibilidad en la creación de proyectos. Al iniciar uno nuevo es posible asignar el nombre del proyecto, el grupo de trabajo y el nivel de visibilidad (privado o público), lo que facilita su colaboración con otros usuarios. Asimismo, la plataforma permite elegir distintos tipos de aplicación; en este caso, el proyecto se configuró como detección de objetos, con el fin de identificar y localizar la roya en las imágenes, tal como se muestra en la Figura 134.

**Figura 134.**

Configuración del proyecto de detección de roya en Roboflow

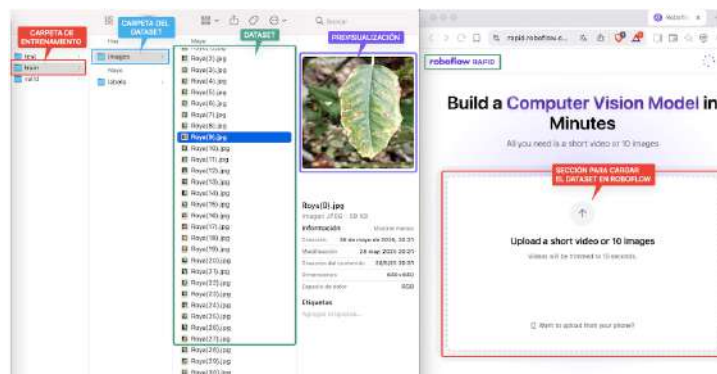


**Nota.** Interfaz de configuración del proyecto en Roboflow. Elaborado por el autor.

Una vez completada la configuración inicial del proyecto, es necesario cargar el dataset en la plataforma Roboflow. La Figura 135 muestra este proceso, donde las imágenes del cultivo afectado por roya se organizan en una carpeta específica destinada al conjunto de entrenamiento, lo que facilita su gestión y posterior etiquetado.

Figura 135.

Organización y carga del dataset de roya en Roboflow



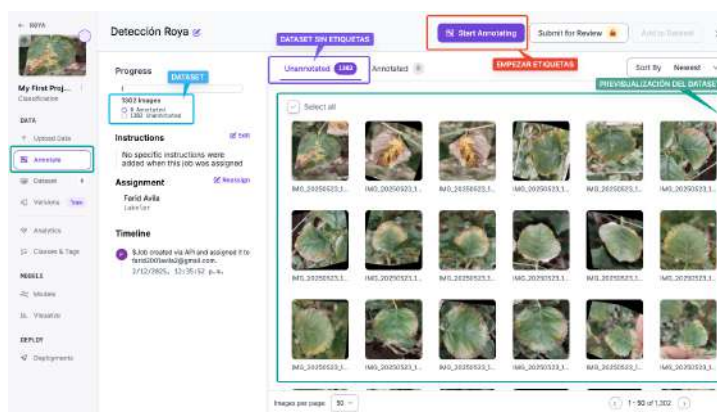
**Nota.** Estructura de carpetas del dataset y sección de carga en Roboflow para el conjunto de entrenamiento.

Elaborado por el autor.

Una vez cargado el dataset en la plataforma de Roboflow, se puede iniciar el proceso de etiquetado. Desde la sección Annotations se dispone de una barra de progreso que permite visualizar el avance del etiquetado, indicando cuántas imágenes han sido etiquetadas y cuántas aún están pendientes. Adicionalmente, se muestra una previsualización del dataset previamente cargado, tal como se observa en la Figura 136.

Figura 136.

Panel de anotaciones y progreso de etiquetado en Roboflow

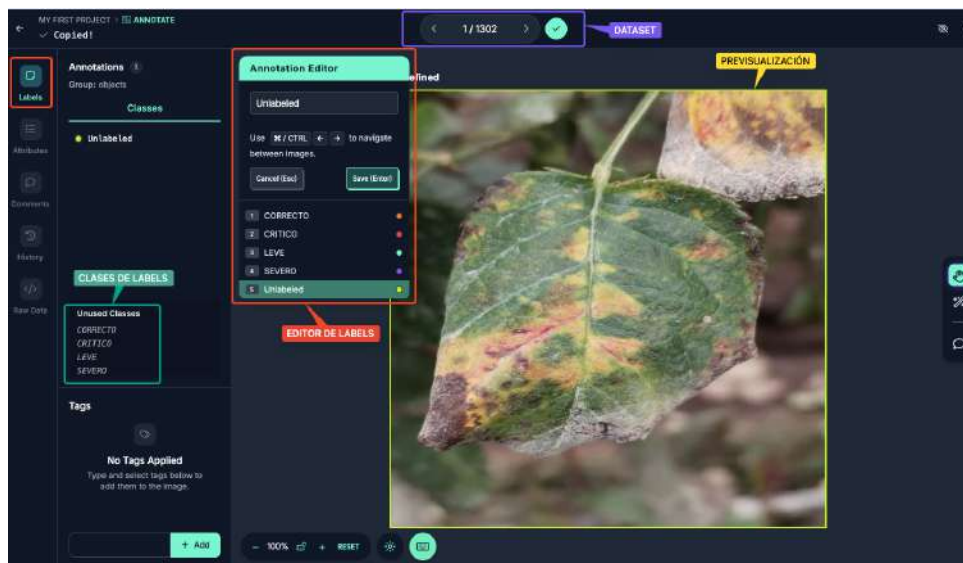


**Nota.** Barra de progreso y vista previa del dataset durante el proceso de etiquetado en Roboflow. Elaborado por el autor.

En el editor de etiquetas de Roboflow se definen las clases asociadas a las anotaciones para identificar la roya según su nivel de gravedad. Para este proyecto se establecieron cuatro categorías: CORRECTO, LEVE, SEVERO y CRÍTICO, tal como se muestra en la Figura 137.

**Figura 137.**

*Definición de clases de severidad de roya en Roboflow*

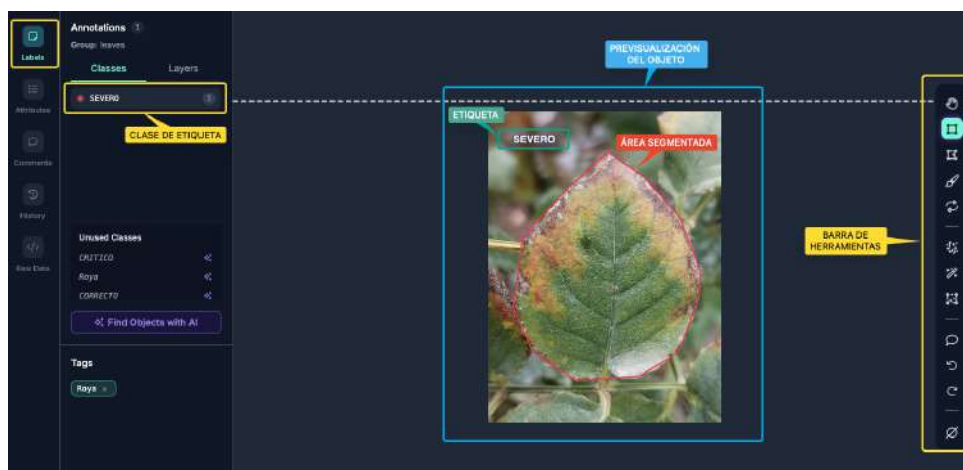


**Nota.** Editor de etiquetas con las clases de labels para la clasificación de la roya. Elaborado por el autor.

Una vez establecidas las clases, se procede a segmentar el área de interés dentro del objeto. Como se muestra en la Figura 138, se visualiza una hoja del cultivo de rosas con síntomas de roya en nivel severo, donde la región afectada se delimita mediante la herramienta de segmentación y se asocia a su etiqueta correspondiente.

**Figura 138.**

*Segmentación del área afectada por roya en Roboflow*



**Nota.** Segmentación de la zona afectada en una hoja de rosa etiquetada como nivel severo dentro del editor de Roboflow. Elaborado por el autor.

Una vez concluido el proceso de etiquetado del dataset, se accede a la sección Versiones. En este apartado se

presenta información detallada del conjunto etiquetado y, además, Roboflow permite dividir el dataset en sub-conjuntos de entrenamiento, validación y prueba para su posterior uso en el modelo de entrenamiento con yolov5. También es posible aplicar ajustes adicionales que mejoran la segmentación, como filtros, rotaciones y otras técnicas de aumento de datos, para finalmente generar el archivo de entrenamiento con las etiquetas definidas, tal como se muestra en la Figura 139.

**Figura 139.**

*Parámetros del preprocesamiento en Roboflow*

**Versiones**

Versiones

Aún no se han creado versiones.

Nombre de la versión: Detección de Roya

**1. INFORMACIÓN DEL DATASET**

- ✓ **Imágenes fuente**: Imágenes: 930, Clases: 5, Sin anotar: 0

**2. DIVISIÓN DEL DATASET**

- ✓ **Prueba de tren/división**: Conjunto de entrenamiento: 651 imágenes, Conjunto de validación: 186 imágenes, Conjunto de prueba: 93 imágenes

**3. AJUSTES DEL PREPROCESAMIENTO**

- ✓ **Preprocesamiento**: Auto-Este: Aplicado, Ajustor: Estirar a 640x640, Ajuste automático del contraste: Uso de ecualización adaptativa

**4. AJUSTES ADICIONALES**

- ✓ **Aumentar**: Voltear: Horizontal, Vertical; 90° Girar: En el sentido de las agujas del reloj, en el sentido contrario a las agujas del reloj, al revés; Rotación: Entre -23° y +23°; Escala: +23% Horizontal, +24% Vertical

**5. CREAR ARCHIVO PARA EL ENTRENAMIENTO**

**5** **Crear**

Revise sus selecciones y seleccione un tamaño de versión para crear una instantánea momento a momento de su conjunto de datos con las transformaciones aplicadas.

Las versiones más grandes tardan más en entrenarse, pero a menudo dan como resultado Mejor rendimiento del modelo. [Ver cómo se calcula esto](#)

Versión máxima tamaño

1,581 imágenes (2x)

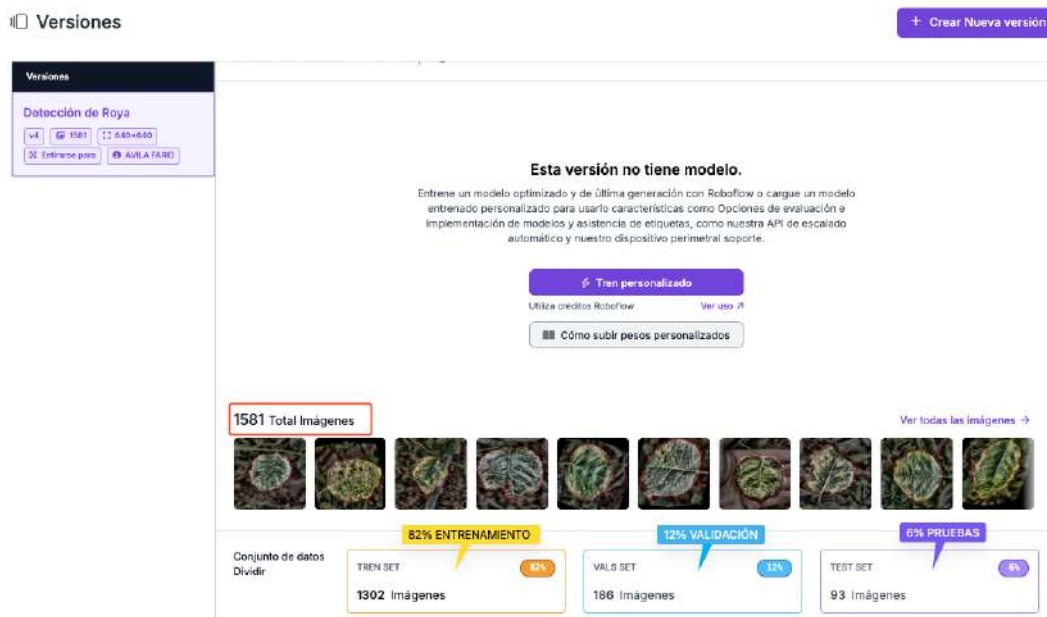
**ENTRENAMIENTO**

**Nota.** Etapas realizadas en el proceso de etiquetado. Elaborado por el autor.

Para generar el archivo .zip con las imágenes y sus etiquetas, Roboflow muestra un resumen del conjunto de datos compuesto por 1581 imágenes, de las cuales el 82 % se destina al entrenamiento, el 12 % a la validación y el 6 % a las pruebas, distribución que se utilizará para el modelo de entrenamiento de YOLOv5, como se observa en la Figura 140.

Figura 140.

Distribución del dataset etiquetado para el entrenamiento del modelo

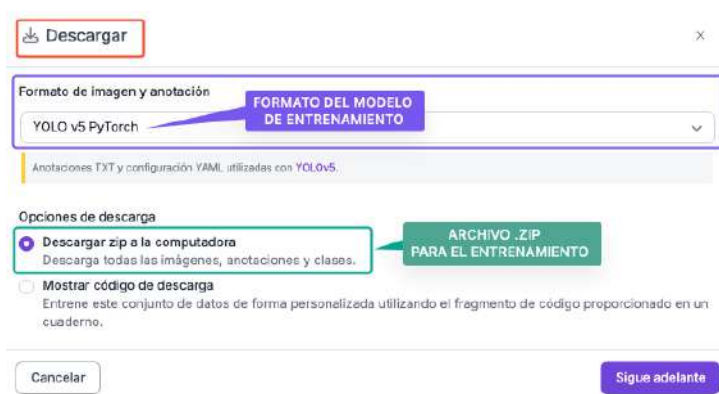


**Nota.** Previsualización de la distribución del dataset con sus respectivas etiquetas. Elaborado por el autor.

Finalmente, para descargar el archivo que contiene el conjunto de datos de entrenamiento, se debe seleccionar correctamente el formato del modelo con el que se trabajará, en este caso se selecciono YOLOv5 PyTorch, para asegurar su compatibilidad con el modelo de entrenamiento. Roboflow permite obtener el dataset etiquetado en un archivo comprimido .zip o mediante un fragmento de código que puede integrarse directamente en el script de entrenamiento. Para este proyecto se eligió la opción de descargar el conjunto de datos en formato .zip, como se muestra en la Figura 141.

Figura 141.

Descarga del dataset etiquetado en formato compatible con YOLOv5



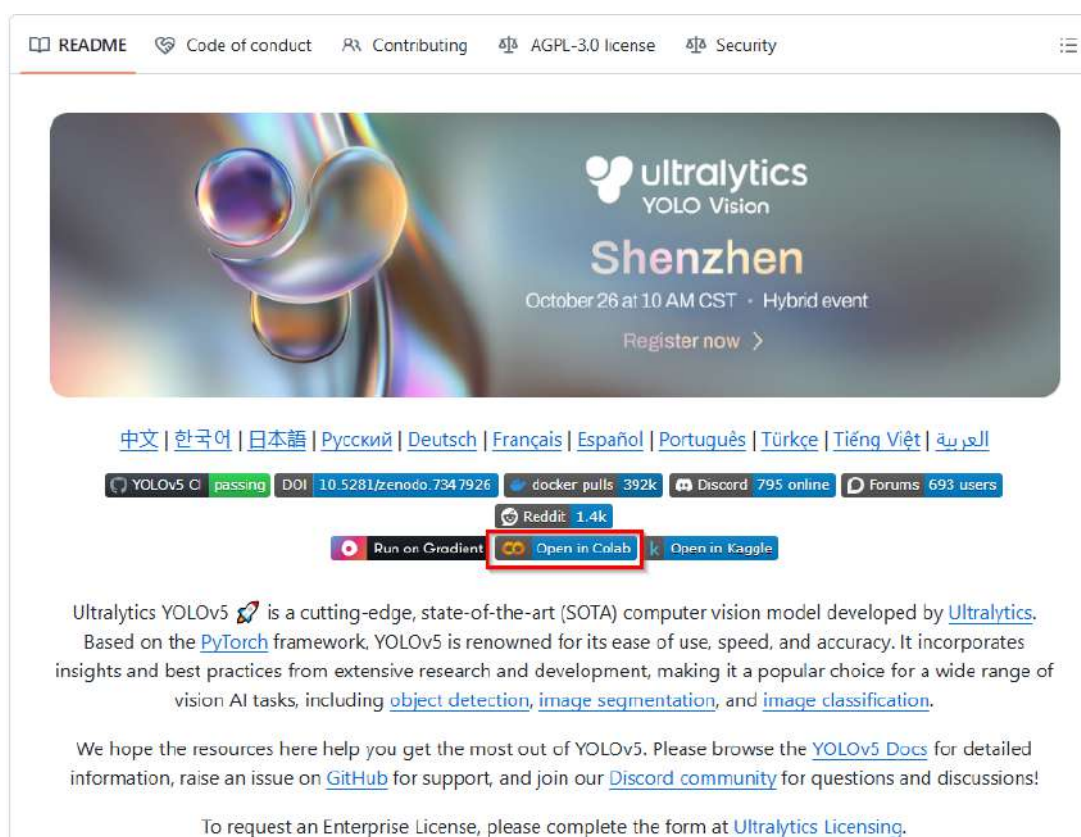
**Nota.** Ventana de descarga en Roboflow compatible con YOLOv5 PyTorch. Elaborado por el autor.

## Anexo 17. Procedimiento de entrenamiento con Google Colab

Para iniciar el proceso de entrenamiento se accede a la documentación oficial de YOLOv5, disponible en su repositorio de GitHub. En este recurso se encuentran las instrucciones, scripts y archivos de configuración necesarios para implementar el modelo en Google Colab. La Figura 142 muestra la vista general del repositorio, donde se organizan las principales carpetas y archivos utilizados durante el entrenamiento.

**Figura 142.**

*Repositorio oficial de YOLOv5 en GitHub*

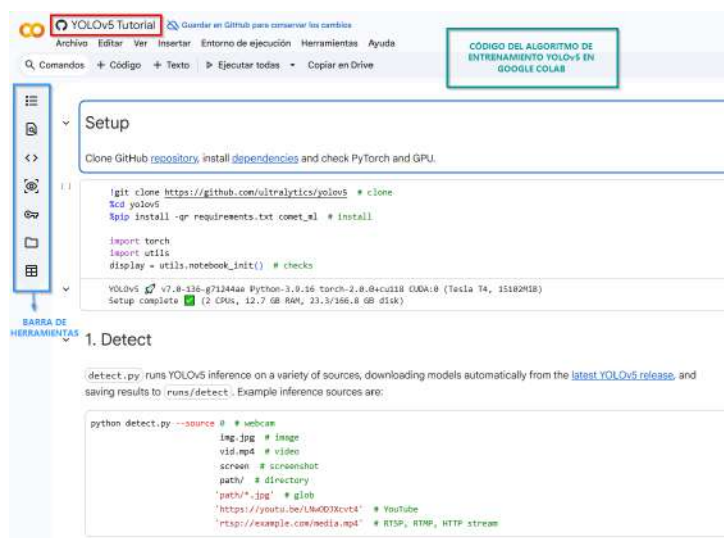


**Nota.** Vista del repositorio oficial de YOLOv5 en GitHub utilizado para el entrenamiento del modelo. Elaborado por el autor.

Una vez abierto el notebook de ejemplo de YOLOv5 en Google Colab desde el repositorio, se visualiza la interfaz de trabajo que incluye la barra de herramientas, el entorno de ejecución y las celdas de código. Esta vista permite revisar, modificar y ejecutar cada instrucción del entrenamiento de manera ordenada y dinámica, tal como se muestra en la Figura 143.

Figura 143.

Interfaz de trabajo de Google Colab con el notebook de YOLOv5

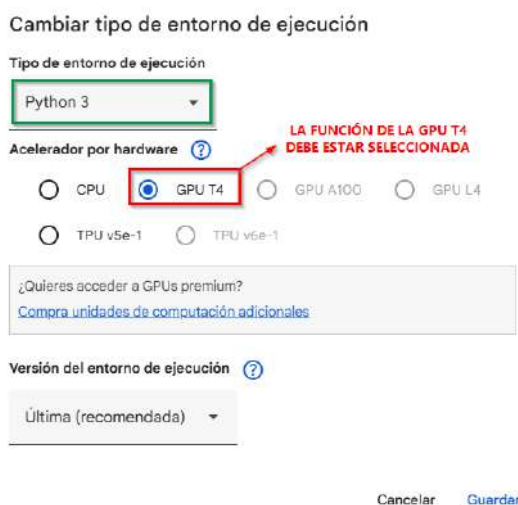


**Nota.** Vista general del entorno de Google Colab utilizado para ejecutar el proceso de entrenamiento de YOLOv5. Elaborado por el autor.

Google Colab permite modificar el entorno de ejecución para optimizar el rendimiento durante el entrenamiento del modelo. Por ello, es necesario verificar en la configuración que la GPU T4 se encuentre activada, ya que este recurso acelera significativamente el procesamiento. La Figura 144 muestra la sección donde se selecciona la GPU para el entrenamiento.

Figura 144.

Selección de GPU en Google Colab



**Nota.** Selección de la GPU T4 en los ajustes de Google Colab para optimizar el entrenamiento del modelo. Elaborado por el autor.

Una vez verificado que la GPU se encuentra activada, se procede a clonar el repositorio de YOLOv5 desde GitHub. Este paso es necesario para descargar el código fuente, las dependencias y los scripts de entrenamiento que permitirán utilizar el algoritmo dentro de Google Colab, tal como se muestra en la Figura 145.

**Figura 145.**

*Clonación del repositorio de YOLOv5 en Google Colab*



```

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

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

*** Cloning into 'yolov5'...
remote: Enumerating objects: 17757, done.
remote: Counting objects: 100% (122/122), done.
remote: Compressing objects: 100% (83/83), done.
remote: Total 17757 (delta 73), reused 39 (delta 39), pack-reused 17635 (from 3)
Receiving objects: 100% (17757/17757), 17.15 MiB | 20.93 MiB/s, done.
Resolving deltas: 100% (12055/12055), done.
/content/yolov5
_____ 780.9/780.9 kB 15.6 MB/s eta 0:00:00
_____ 1.1/1.1 MB 43.4 MB/s eta 0:00:00
_____ 1.3/1.3 MB 70.2 MB/s eta 0:00:00

```

**CLONACIÓN DEL DIRECTORIO DE YOLOv5**

**Nota.** Clonación del repositorio de YOLOv5 desde GitHub para el entrenamiento en Google Colab. Elaborado por el autor. Elaborado por el autor.

También es necesario cargar la carpeta del dataset en Google Colab para iniciar el proceso de entrenamiento. Para ello, se importa el archivo comprimido almacenado previamente en Google Drive y se realiza su descompresión dentro del entorno de trabajo. La Figura 146 muestra el código utilizado para la importación y extracción del dataset en Google Colab.

**Figura 146.**

*Importación del dataset desde Google Drive hacia Google Colab*



```

from google.colab import drive

drive.mount('/content/drive')

Mounted at /content/drive

!unzip -q '/content/drive/MyDrive/Colab Notebooks/data.zip' -d /content/

```

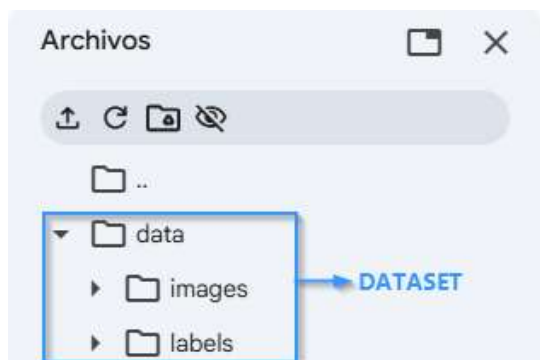
**IMPORTACIÓN DEL DATASET DESDE GOOGLE DRIVE**

**Nota.** Importación y descompresión del dataset almacenado en Google Drive dentro de Google Colab. Elaborado por el autor.

En la Figura 147 se observa que, una vez descomprimido el dataset desde Google Drive, Google Colab cargó correctamente todos los archivos. Esto confirma que la importación y extracción del conjunto de datos se realizó de manera adecuada, permitiendo continuar con el proceso de entrenamiento sin inconvenientes.

**Figura 147.**

*Verificación de la carga del dataset en Google Colab*



**Nota.** Carga de los archivos del dataset tras su descompresión en Google Colab. Elaborado por el autor.

Una vez configurados los parámetros de entrenamiento, Google Colab ejecuta el proceso y muestra en pantalla el avance de cada época (epoch). En la Figura 148 se aprecia el comando empleado, donde se definen el tamaño de imagen (640), el tamaño de lote (batch 16), el número de iteraciones (150 epochs), la ruta del archivo .yaml y la versión de YOLOv5 utilizada para el ajuste del modelo. Esta ejecución permite seguir en tiempo real métricas como la pérdida y el rendimiento del modelo a lo largo de todo el entrenamiento.

**Figura 148.**

*Ejecución del entrenamiento y parámetros configurados en YOLOv5*

```

# Train YOLOv5s on COCO128 for 3 epochs
!python train.py --img 640 --batch 16 --epochs 150 --data /content/yolov5/data/CustomData.yaml --weights yolov5s.pt --cache

PARÁMETROS DEL ENTRENAMIENTO
-----
57/140 15.26 0.02576 0.01638 0.0003843 82 640: 64% 43/67 [00:40<00:27, 1.16s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
with torch.cuda.amp.autocast(enabled=True): 0.01641 0.0003844 82 640: 66% 44/67 [00:50<00:20, 1.17s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
57/140 15.26 0.0258 0.01637 0.0003819 49 640: 67% 45/67 [00:51<00:25, 1.16s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
with torch.cuda.amp.autocast(enabled=True): 0.01635 0.0003849 50 640: 69% 46/67 [00:52<00:24, 1.15s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
57/140 15.26 0.02587 0.01629 0.0003809 61 640: 70% 47/67 [00:53<00:23, 1.10s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
with torch.cuda.amp.autocast(enabled=True): 0.01629 0.0003809 61 640: 72% 48/67 [00:54<00:21, 1.14s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
57/140 15.26 0.02586 0.01631 0.0003809 70 640: 73% 49/67 [00:55<00:20, 1.14s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
with torch.cuda.amp.autocast(enabled=True): 0.01635 0.0003857 73 640: 75% 50/67 [00:57<00:19, 1.13s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
57/140 15.26 0.02582 0.01648 0.0003827 85 640: 76% 51/67 [00:58<00:18, 1.14s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
with torch.cuda.amp.autocast(enabled=True): 0.01633 0.0003800 95 640: 78% 52/67 [00:59<00:17, 1.13s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
57/140 15.26 0.02583 0.01651 0.0003774 05 640: 79% 53/67 [01:00<00:15, 1.13s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
with torch.cuda.amp.autocast(enabled=True): 0.01654 0.0003751 72 640: 81% 54/67 [01:01<00:14, 1.12s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
57/140 15.26 0.0259 0.01646 0.0003714 00 640: 82% 55/67 [01:02<00:13, 1.15s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
with torch.cuda.amp.autocast(enabled=True): 0.01645 0.0003680 60 640: 84% 56/67 [01:04<00:12, 1.14s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
57/140 15.26 0.02591 0.01654 0.0003663 80 640: 85% 57/67 [01:05<00:11, 1.14s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
with torch.cuda.amp.autocast(enabled=True): 0.01649 0.0003643 53 640: 87% 58/67 [01:06<00:10, 1.13s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
57/140 15.26 0.02593 0.01652 0.0004039 04 640: 88% 59/67 [01:07<00:09, 1.14s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
with torch.cuda.amp.autocast(enabled=True): 0.01640 0.0004009 58 640: 90% 60/67 [01:08<00:08, 1.13s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
57/140 15.26 0.02590 0.01652 0.0003989 88 640: 91% 61/67 [01:09<00:06, 1.13s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
with torch.cuda.amp.autocast(enabled=True): 0.01647 0.0003964 58 640: 93% 62/67 [01:10<00:05, 1.12s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
57/140 15.26 0.02598 0.01644 0.0003904 43 640: 94% 63/67 [01:11<00:04, 1.13s/it]/content/yolov5/train.py:413: FutureWarning: `torch.cuda.amp.autocast(args...
with torch.cuda.amp.autocast(enabled=True): 0.01644 0.0003904 43

```

**Nota.** Proceso de entrenamiento mostrando la evolución de las épocas y los parámetros utilizados. Elaborado por el autor.

Finalmente, una vez completado el entrenamiento, Google Colab permite descargar el archivo .pt, que corresponde al modelo entrenado y listo para ser integrado en el sistema de detección desarrollado en Python, utilizando las librerías de PyTorch.

### Figura 149.

*Descarga del modelo entrenado en Google Colab*



**Nota.** Descarga del archivo .pt generado tras el proceso de entrenamiento del modelo en Google Colab.

Elaborado por el autor.

## Anexo 18. Código

```
34
35
36 # =====
37 # IMPORTACIONES Y CONFIGURACIONES
38 # =====
39
40 from flask import Flask, render_template, Response, session, redirect,
    url_for, flash, request, jsonify, send_from_directory
41 from flask_sqlalchemy import SQLAlchemy
42 from xhtml2pdf import pisa
43 from io import BytesIO
44 import os
45 from datetime import datetime, timedelta
46 from clima import obtener_clima
47 from werkzeug.utils import secure_filename
48 import cv2
```

```
49 import numpy as np
50 import torch
51 import re
52 import subprocess
53 import time
54 import threading
55 from sqlalchemy.dialects.mysql import dialect # Parche para evitar error
    de isolation_level
56
57 # =====
58 # CONFIGURACIÓN FLASK Y BASE DE DATOS
59 # =====
60
61 app = Flask(__name__)
62 app.secret_key = 'clave_secreta_supersegura'
63 app.permanent_session_lifetime = timedelta(minutes=30)
64
65 # Parchemos el error de isolation_level de PyMySQL
66 dialect.get_isolation_level = lambda *args, **kwargs: "READ_COMMITTED"
67
68 # Configuración conexión MySQL con SQLAlchemy
69 app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql+pymysql://admin:1234
    @localhost/DBRoya'
70 app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
71 db = SQLAlchemy(app)
72
73 # =====
74 # MODELOS DE BASE DE DATOS (ORM)
75 # =====
76
77 class Usuario(db.Model):
78     __tablename__ = 'usuarios'
79     id = db.Column(db.Integer, primary_key=True)
80     username = db.Column(db.String(50), unique=True, nullable=False)
81     password = db.Column(db.String(255), nullable=False)
```

```

82
83 class Registro(db.Model):
84     __tablename__ = 'registros'
85     id_registro = db.Column(db.String(32), primary_key=True)
86     fecha_hora = db.Column(db.DateTime, nullable=False)
87     nombre_proyecto = db.Column(db.String(100))
88     numero_invernadero = db.Column(db.String(32))
89     tipo_operacion = db.Column(db.String(64))
90     zona = db.Column(db.String(64))
91     temperatura = db.Column(db.Float)
92     humedad = db.Column(db.Float)
93     descripcion_clima = db.Column(db.String(100))
94     viento = db.Column(db.Float)
95     usuario_id = db.Column(db.Integer, db.ForeignKey('usuarios.id'))
96
97 class Archivo(db.Model):
98     __tablename__ = 'archivos'
99     archivo_id = db.Column(db.Integer, primary_key=True, autoincrement=True
100     )
101     usuario_id = db.Column(db.Integer, db.ForeignKey('usuarios.id'))
102     tipo = db.Column(db.Enum('imagen', 'video'), nullable=False)
103     archivo = db.Column(db.String(255), nullable=False)
104     nombre_personalizado = db.Column(db.String(255), default='')
105     fecha_subida = db.Column(db.DateTime, default=datetime.utcnow)
106 # =====
107 # CREACIÓN DE CARPETAS NECESARIAS
108 # =====
109
110 os.makedirs('Dataset', exist_ok=True) # Carpeta para archivos originales
111 os.makedirs('Resultados', exist_ok=True) # Carpeta para resultados
112     procesados
113 os.makedirs('PDFs', exist_ok=True) # Carpeta para PDFs generados
114
114 # Lock para sincronizar acceso al frame del detector

```

```

115 detector_lock = threading.Lock()
116
117 # =====
118 # CLASE: DetectorRoya (YOLOv5)
119 # =====
120
121 class DetectorRoya:
122     def __init__(self, model_path='Yolov5sRoya.pt'):
123         # Carga el modelo YOLOv5 personalizado
124         try:
125             self.model = torch.hub.load('ultralytics/yolov5', 'custom',
126                                         path=model_path, force_reload=False)
127             print("Modelo YOLOv5 cargado correctamente.")
128         except Exception as e:
129             print(f"[DetectorRoya] Error cargando modelo: {e}")
130             self.model = None
131
132         self.cap = None # Captura de video (cámara)
133         self.total_frames = 0 # Contador frames procesados
134         self.total_detections = 0 # Contador total detecciones
135         self.start_time = time.time() # Tiempo inicio detección
136         self._open_camera() # Inicializa la cámara
137
138     def _open_camera(self):
139         # Intenta abrir la cámara; si está abierta la cierra primero
140         self.stop()
141         try:
142             self.cap = cv2.VideoCapture(0)
143             if not self.cap.isOpened():
144                 print(f"[DetectorRoya] No se pudo acceder a la cámara. ¿Está
145                     conectada/ocupada?")
146                 self.cap.release()
147                 self.cap = None
148         except Exception as e:
149             print(f"[DetectorRoya] Error abriendo cámara: {e}")

```

```
148         self.cap = None
149
150     def _update_frames(self):
151         # (No se usa en el código actual, pero puede ser hilo para
152           actualización continua)
153         while self.running:
154             if self.cap is None or not self.cap.isOpened():
155                 time.sleep(0.1)
156                 continue
157             ret, frame = self.cap.read()
158             if not ret or frame is None:
159                 time.sleep(0.05)
160                 continue
161
162             results = self.model(frame)
163             detecciones = results.xyxy[0].shape[0]
164             self.total_frames += 1
165             self.total_detecciones += detecciones
166
167             if hasattr(results, 'render'):
168                 rendered = results.render()
169                 frame_rendered = rendered[0] if isinstance(rendered, list)
170                   else frame
171             else:
172                 frame_rendered = frame
173
174             _, jpeg = cv2.imencode('.jpg', frame_rendered)
175             with detector_lock:
176                 self.frame = jpeg.tobytes()
177
178     def get_frame(self):
179         # Obtiene frame actual procesado (imagen JPEG)
180         with detector_lock:
```

```

181 def get_stats(self):
182     # Retorna estadísticas de FPS y detecciones promedio
183     elapsed_time = time.time() - self.start_time
184     fps = self.total_frames / elapsed_time if elapsed_time > 0 else 0
185     avg_detections = self.total_detections / self.total_frames if self.
        total_frames > 0 else 0
186     return {
187         "total_frames": self.total_frames,
188         "total_detections": self.total_detections,
189         "fps": round(fps, 2),
190         "avg_detections_per_frame": round(avg_detections, 2)
191     }
192
193 def get_frame(self):
194     # Método principal para obtener frame desde cámara y modelo,
195     # con reconexión automática si falla algo
196     try:
197         if self.model is None:
198             print("[DetectorRoya] El modelo YOLO no está cargado.")
199             return None
200         if self.cap is None or not self.cap.isOpened():
201             print("[DetectorRoya] Intentando reconectar cámara...")
202             self._open_camera()
203             if self.cap is None or not self.cap.isOpened():
204                 return None
205         ret, frame = self.cap.read()
206         if not ret or frame is None:
207             print("[DetectorRoya] Frame no válido, intentando siguiente
                frame...")
208             return None
209         results = self.model(frame)
210         detecciones = results.xyxy[0].shape[0]
211         self.total_frames += 1
212         self.total_detections += detecciones
213         if hasattr(results, 'render'):

```

```
214         rendered = results.render()
215         frame_rendered = rendered[0] if isinstance(rendered, list)
                else frame
216     else:
217         frame_rendered = frame
218     _, buffer = cv2.imencode('.jpg', frame_rendered)
219     return buffer.tobytes()
220 except Exception as e:
221     print(f"[DetectorRoya]_Error_al_capturar_frame:_{e}")
222     return None
223
224 def stop(self):
225     # Libera la cámara si está abierta
226     try:
227         if self.cap is not None and self.cap.isOpened():
228             self.cap.release()
229             self.cap = None
230     except Exception as e:
231         print(f"[DetectorRoya]_Error_al_liberar_cámara:_{e}")
232
233 def restart(self):
234     # Reinicia la cámara y detector
235     self.stop()
236     self._open_camera()
237
238 def __del__(self):
239     # Destructor para asegurar liberar recursos
240     self.stop()
241
242 # Instancia global del detector
243 detector = None
244
245 # =====
246 # RUTAS RELACIONADAS AL DETECTOR Y STREAMING
247 # =====
```

```
248
249 @app.route('/restart', methods=['POST'])
250 def restart_detector():
251     # Reinicia el detector si está activo
252     global detector
253     if detector:
254         detector.restart()
255         return jsonify({"status": "Detector_reiniciado_correctamente"})
256     else:
257         return jsonify({"error": "Detector_no_iniciado"}), 400
258
259 # =====
260 # RUTAS DE AUTENTICACIÓN Y DASHBOARD
261 # =====
262
263 @app.route('/')
264 def home():
265     # Redirige a login
266     return redirect(url_for('login'))
267
268 @app.route('/login', methods=['GET', 'POST'])
269 def login():
270     # Manejo de login con verificación simple (ojo que la contraseña está
271     # en texto plano)
272     if request.method == 'POST':
273         user = request.form['username']
274         password = request.form['password']
275         usuario = Usuario.query.filter_by(username=user).first()
276         if usuario and usuario.password == password:
277             session['username'] = user
278             session['user_id'] = usuario.id
279             flash('Inicio_de_sesión_exitoso', 'success')
280             return redirect(url_for('dashboard'))
281         else:
282             flash('Usuario_o_contraseña_incorrectos', 'error')
```

```
282         return redirect(url_for('login'))
283     return render_template('login.html')
284
285 @app.route('/dashboard', methods=['GET', 'POST'])
286 def dashboard():
287     # Página principal post login donde se registra info y se obtiene clima
288     if 'username' not in session:
289         flash('Primero inicia sesión', 'warning')
290         return redirect(url_for('login'))
291
292     ciudad_por_defecto = 'Tabacundo'
293
294     if request.method == 'POST':
295         # Procesa el formulario de registro y guarda en BD
296         ciudad = request.form.get('ciudad', ciudad_por_defecto)
297         try:
298             clima = obtener_clima(ciudad)
299             if not clima:
300                 flash(f"No se encontró clima para '{ciudad}'. Usando valores por defecto.", 'warning')
301         except Exception as e:
302             print(f"[ERROR] [obtener_clima] {e}")
303             clima = None
304             flash(f"Ocurrió un error al obtener el clima para '{ciudad}'. Se usan valores por defecto.", 'danger')
305
306     id_registro = request.form['registro_id']
307     registro = Registro(
308         id_registro=id_registro,
309         fecha_hora=datetime.strptime(request.form['fecha_hora'], '%Y-%m-%dT%H:%M'),
310         nombre_proyecto=request.form.get('nombre_proyecto', ''),
311         numero_invernadero=request.form.get('numero_invernadero', ''),
312         tipo_operacion=request.form['tipo_operacion'],
313         zona=ciudad,
```

```
314     usuario_id=session['user_id'],
315     temperatura=clima['temperatura'] if clima else 20.0,
316     humedad=clima['humedad'] if clima else 50.0,
317     descripcion_clima=clima['descripcion'] if clima else "N/A",
318     viento=clima['viento'] if clima else 0.0
319 )
320 db.session.add(registro)
321 db.session.commit()
322
323 # Guarda en sesión para usar en streaming
324 session['registro_actual'] = {
325     "id_registro": id_registro,
326     "fecha_hora": request.form['fecha_hora'],
327     "nombre_proyecto": request.form.get('nombre_proyecto', ''),
328     "numero_invernadero": request.form.get('numero_invernadero', ''
329     ),
330     "tipo_operacion": request.form['tipo_operacion'],
331     "zona": ciudad,
332     "usuario": session['username'],
333     "temperatura": clima['temperatura'] if clima else 20.0,
334     "humedad": clima['humedad'] if clima else 50.0,
335     "descripcion_clima": clima['descripcion'] if clima else "N/A",
336     "viento": clima['viento'] if clima else 0.0
337 }
338
339 return redirect(url_for('start_stream'))
340
341 # GET: muestra dashboard con clima por defecto
342 try:
343     clima = obtener_clima(ciudad_por_defecto)
344 except Exception as e:
345     print(f"[ERROR] [obtener_clima]_{e}")
346     clima = None
347
348 now = datetime.utcnow() - timedelta(hours=5) # Ajuste horario Ecuador
349 fecha_ecuador = now.strftime('%Y-%m-%dT%H:%M')
```

```
348
349     return render_template('dashboard.html',
350                            fecha_ecuador=fecha_ecuador,
351                            temperatura=clima['temperatura'] if clima else
352                                20.0,
353                            humedad=clima['humedad'] if clima else 50.0,
354                            ciudad=ciudad_por_defecto,
355                            descripcion=clima['descripcion'] if clima else "
356                                N/A",
357                            viento=clima['viento'] if clima else 0.0)
358
359 # =====
360 # STREAMING Y TRANSMISIÓN EN VIVO
361 # =====
362
363 @app.route('/start_stream')
364 def start_stream():
365     # Renderiza página con transmisión en vivo, requiere sesión activa
366     if 'username' not in session:
367         flash('Inicia sesión para acceder al sistema', 'warning')
368         return redirect(url_for('login'))
369     return render_template('stream.html', registro=session.get('
370         registro_actual'))
371
372 @app.route('/iniciar_detector', methods=['POST'])
373 def iniciar_detector():
374     # Inicializa el detector YOLO globalmente para streaming
375     global detector
376     try:
377         detector = DetectorRoya()
378         return jsonify({"status": "ok"})
379     except Exception as e:
380         return jsonify({"status": "error", "mensaje": str(e)})
381
382 def gen_frames():
```

```
380 # Generador de frames JPEG para streaming via Flask Response
381 global detector
382 while True:
383     if detector is None:
384         break
385     frame = detector.get_frame()
386     if frame is None:
387         break
388     yield (b'--frame\r\n'
389           b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
390
391 @app.route('/video_feed')
392 def video_feed():
393     # Ruta que envía la transmisión de video en vivo
394     if 'username' not in session:
395         return redirect(url_for('login'))
396     return Response(gen_frames(), mimetype='multipart/x-mixed-replace;
397                   boundary=frame')
398
399 @app.route('/stats')
400 def stats():
401     # Devuelve estadísticas del detector en JSON
402     global detector
403     if detector:
404         return jsonify(detector.get_stats())
405     return jsonify({"error": "Detector no iniciado"}), 400
406
407 @app.route('/finalizar_stream', methods=['POST'])
408 def finalizar_stream():
409     # Detiene el detector y libera recursos
410     global detector
411     if detector:
412         detector.stop()
413         detector = None
414     return ('', 204)
```

```
414
415 @app.route('/limpiar_registro', methods=['POST'])
416 def limpiar_registro():
417     # Limpia el registro actual de la sesión
418     session.pop('registro_actual', None)
419     return ('', 204)
420
421 @app.route('/logout')
422 def logout():
423     # Cierra sesión y libera recursos
424     session.pop('username', None)
425     session.pop('user_id', None)
426     session.pop('registro_actual', None)
427     global detector
428     if detector:
429         detector.stop()
430         detector = None
431     flash('Sesión_cerrada_correctamente', 'info')
432     return redirect(url_for('home'))
433
434 # =====
435 # GESTIÓN DE SUBIDA Y VISUALIZACIÓN DE ARCHIVOS UPLOAD
436 # =====
437
438 @app.route('/upload')
439 def upload_index():
440     # Muestra archivos subidos del usuario agrupados por fecha y tipo
441     if 'username' not in session:
442         flash('Inicia_sesión_para_ver_tus_archivos', 'warning')
443         return redirect(url_for('login'))
444
445     user_id = session['user_id']
446     archivos = Archivo.query.filter_by(usuario_id=user_id).order_by(Archivo
447         .fecha_subida.desc()).all()
```

```
448     archivos_por_fecha = {}
449     for archivo in archivos:
450         fecha_str = archivo.fecha_subida.strftime("%Y-%m-%d")
451         if fecha_str not in archivos_por_fecha:
452             archivos_por_fecha[fecha_str] = {"videos": [], "imagenes": []}
453         archivo_data = {
454             "filename": archivo.archivo,
455             "nombre_personalizado": archivo.nombre_personalizado or ""
456         }
457         if archivo.tipo == 'video':
458             archivos_por_fecha[fecha_str]["videos"].append(archivo_data)
459         else:
460             archivos_por_fecha[fecha_str]["imagenes"].append(archivo_data)
461
462     return render_template('upload.html', archivos_por_fecha=
463                           archivos_por_fecha)
464 @app.route('/cargar_archivo', methods=['POST'])
465 def cargar_archivo():
466     # Procesa la carga de un archivo (imagen o video)
467     if 'username' not in session:
468         flash('Inicia sesión para cargar archivos', 'warning')
469         return redirect(url_for('login'))
470
471     if 'archivo' not in request.files:
472         flash('No se subió ningún archivo', 'warning')
473         return redirect(url_for('upload_index'))
474
475     archivo = request.files['archivo']
476     if archivo.filename == '':
477         flash('Archivo inválido', 'error')
478         return redirect(url_for('upload_index'))
479
480     nombre_personalizado = request.form.get('nombrePersonalizado', '')
481
```

```

482 # Guarda archivo con timestamp para evitar colisiones
483 timestamp = datetime.now().strftime('%Y%m%d_%H%M%S')
484 filename = f"{timestamp}_{secure_filename(archivo.filename)}"
485 filepath = os.path.join('Dataset', filename)
486 archivo.save(filepath)
487
488 # Define tipo por extensión
489 extension = os.path.splitext(filename)[1].lower()
490 tipo = 'video' if extension in ['.mp4', '.avi', '.mov'] else 'imagen'
491
492 flash('Archivo␣cargado␣correctamente.␣Iniciando␣detección...', 'success
      ')
493
494 output_filename = procesar_archivo_individual(filepath)
495
496 # Guarda, consulta y compara el resultado en BD
497 nuevo_archivo = Archivo(
498     usuario_id=session['user_id'],
499     tipo=tipo,
500     archivo=output_filename,
501     nombre_personalizado=nombre_personalizado
502 )
503 db.session.add(nuevo_archivo)
504 db.session.commit()
505
506 return redirect(url_for('upload_index'))
507
508 def procesar_archivo_individual(filepath):
509     # Procesa el archivo cargado con YOLOv5 para detección y guarda
510     resultado
511     try:
512         print("Cargando␣modelo␣YOLOv5...")
513         model1 = torch.hub.load('ultralytics/yolov5', 'custom', path='./
          Yolov5sRoya.pt', force_reload=False)
514         print("␣Modelo␣cargado.")

```

```
514
515     extension = os.path.splitext(filepath)[1].lower()
516     output_path = ""
517     base_filename = os.path.basename(filepath).split('.')[0]
518
519     # Procesamiento para imagen
520     if extension in ['.jpg', '.jpeg', '.png']:
521         imagen = cv2.imread(filepath)
522         if imagen is None:
523             print("Imagen no válida.")
524             return ""
525         resized = cv2.resize(imagen, (416, 416))
526         results = model1(resized)
527         output = np.squeeze(results.render())
528         output_path = os.path.join('Resultados', f"det_{os.path.
529             basename(filepath)}")
530         cv2.imwrite(output_path, output)
531         return os.path.basename(output_path)
532
533     # Procesamiento para video
534     elif extension in ['.mp4', '.avi', '.mov']:
535         avi_output_path = os.path.join('Resultados', f"{base_filename}
536             _detectado.avi")
537         cap = cv2.VideoCapture(filepath)
538         if not cap.isOpened():
539             print("No se pudo abrir el video.")
540             return ""
541         ret, frame = cap.read()
542         if not ret:
543             print("El video no tiene frames válidos.")
544             cap.release()
545             return ""
546         fps = int(cap.get(cv2.CAP_PROP_FPS)) or 20
```

```
547 ancho = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)) or 640
548 alto = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)) or 480
549
550 # Ajuste para que ancho y alto sean pares (requerimiento de
      códecs)
551 ancho = int(ancho // 2 * 2)
552 alto = int(alto // 2 * 2)
553
554 fourcc = cv2.VideoWriter_fourcc(*'XVID')
555 out = cv2.VideoWriter(avi_output_path, fourcc, fps, (ancho,
      alto))
556
557 cap.set(cv2.CAP_PROP_POS_FRAMES, 0)
558 while True:
559     ret, frame = cap.read()
560     if not ret:
561         break
562     results = model1(frame)
563     frame_render = np.squeeze(results.render())
564     frame_render = cv2.resize(frame_render, (ancho, alto))
565     # Asegura que el frame tenga 3 canales para video
566     if len(frame_render.shape) == 2 or frame_render.shape[2] !=
        3:
567         frame_render = cv2.cvtColor(frame_render, cv2.
            COLOR_GRAY2BGR)
568     out.write(frame_render)
569 cap.release()
570 out.release()
571
572 # Convierte AVI a MP4 H.264 para mejor compatibilidad
573 mp4_output_path = os.path.join('Resultados', f"{base_filename}
      _detectado.mp4")
574 comando = [
575     'ffmpeg', '-y', '-i', avi_output_path,
576     '-vcodec', 'libx264', '-acodec', 'aac', mp4_output_path
```

```
577     ]
578     print("Convirtiendo AVI a MP4 H.264 con ffmpeg...")
579     resultado = subprocess.run(comando, stdout=subprocess.PIPE,
580                               stderr=subprocess.PIPE)
581     if os.path.exists(mp4_output_path) and os.path.getsize(
582         mp4_output_path) > 0:
583         print(f"Video MP4 compatible guardado: {mp4_output_path}")
584         output_path = mp4_output_path
585         try:
586             os.remove(avi_output_path) # Borra AVI temporal
587         except Exception as e:
588             print(f"No se pudo borrar el AVI temporal: {e}")
589         else:
590             print("Error al convertir a MP4. Revisa ffmpeg.")
591             output_path = avi_output_path
592         return os.path.basename(output_path)
593     else:
594         print("Tipo de archivo no soportado")
595         return ""
596 except Exception as e:
597     print("Error procesando archivo:", e)
598     return ""
599
600
601 # Según el número de detecciones, se clasifica el nivel de severidad de la
602     roya en el cultivo.
603
604 def clasificar_severidad(cantidad_detecciones):
605
606     # Si se detectan 2 o menos casos de roya, se considera una afectación
607     leve.
608
609     if cantidad_detecciones <= 2:
610         return "leve"
```



```

638         zona=datos.get("zona"),
639         tipo_operacion=datos.get("tipo_operacion"),
640         nombre_proyecto=datos.get("nombre_proyecto"),
641         numero_invernadero=datos.get("numero_invernadero
        "),
642         temperatura=float(datos.get("temperatura", 0)),
643         humedad=float(datos.get("humedad", 0)),
644         descripcion_clima=datos.get("descripcion_clima")
        ,
645         viento=datos.get("viento"),
646         estado=datos.get("estado"),
647         avg=datos.get("avg"),
648         frames=datos.get("frames"),
649         detecciones=datos.get("detecciones"),
650         ultimos_registros=ultimos_registros,
651         tendencia=tendencia)
652     result = BytesIO()
653     pisa.CreatePDF(html, dest=result)
654     response = Response(result.getvalue(), content_type='application/pdf')
655     response.headers['Content-Disposition'] = f"inline; filename=reporte_{
        datos.get('id')}.pdf"
656     return response
657
658 # =====
659 # RUTAS PARA ACCEDER A ARCHIVOS Y PDFS
660 # =====
661
662 @app.route('/resultados/<path:filename>')
663 def obtener_resultado(filename):
664     # Devuelve archivo procesado (imagen/video) para descarga o
        visualización
665     return send_from_directory('Resultados', filename)
666
667 @app.route('/eliminar_resultado/<filename>', methods=['POST'])
668 def eliminar_resultado(filename):

```

```
669 # Elimina archivo procesado y su registro en BD
670 archivo = Archivo.query.filter_by(archivo=filename, usuario_id=session.
        get('user_id')).first()
671 ruta = os.path.join('Resultados', filename)
672 if archivo:
673     db.session.delete(archivo)
674     db.session.commit()
675 if os.path.exists(ruta):
676     os.remove(ruta)
677     flash(f'Archivo eliminado: {filename}', 'info')
678 else:
679     flash(f'No se encontró el archivo: {filename}', 'error')
680 return redirect(url_for('upload_index'))
681
682 @app.route('/pdfs/<path:filename>')
683 def descargar_pdf(filename):
684     # Descarga archivo PDF generado
685     return send_from_directory('PDFs', filename)
686
687 # =====
688 # HISTORIAL DE RESULTADOS
689 # =====
690
691 @app.route('/historial')
692 def ver_historial():
693     # Muestra historial de archivos procesados por el usuario
694     if 'username' not in session:
695         flash('Inicia sesión para ver tu historial', 'warning')
696         return redirect(url_for('login'))
697
698     user_id = session['user_id']
699     archivos = Archivo.query.filter_by(usuario_id=user_id).order_by(Archivo
        .fecha_subida.desc()).all()
700
701     registros = []
```

```
702     for archivo in archivos:
703         registros.append({
704             'fecha_hora': archivo.fecha_subida.strftime('%Y-%m-%d_%H:%M:%S'
705                 ),
706             'usuario': session['username'],
707             'archivo_original': archivo.archivo,
708             'resultado': archivo.archivo,
709             'nombre_personalizado': getattr(archivo, 'nombre_personalizado'
710                 , ''), # Si existe campo
711             'tipo': archivo.tipo
712         })
713
714     return render_template('historial.html', registros=registros)
715
716 # =====
717 # API DE CLIMA
718 # =====
719
720 @app.route('/api/clima')
721 def api_clima():
722     # Retorna datos climáticos en JSON para la ciudad solicitada
723     ciudad = request.args.get('ciudad', 'Tabacundo')
724     datos = obtener_clima(ciudad)
725     if datos:
726         return jsonify(datos)
727     return jsonify({"error": "Ciudad no encontrada"}), 404
728
729 # =====
730 # EJECUCIÓN DEL SERVIDOR
731 # =====
732
733 if __name__ == '__main__':
734     with app.app_context():
735         db.create_all() # Crea tablas en la base de datos si no existen
736     app.run(host='0.0.0.0', port=5000, debug=True)
```

Listing 1: Fragmento del archivo app.py

```
34
35 import requests
36
37 def obtener_clima(ciudad):
38     try:
39         api_key = "dbd10be6005e0ad92123756eb2ce273e"
40         url = f"https://api.openweathermap.org/data/2.5/weather?q={ciudad}&
           units=metric&appid={api_key}&lang=es"
41
42         response = requests.get(url)
43         if response.status_code == 200:
44             data = response.json()
45             return {
46                 "temperatura": data["main"]["temp"],
47                 "humedad": data["main"]["humidity"],
48                 "descripcion": data["weather"][0]["description"],
49                 "viento": data["wind"]["speed"]
50             }
51         else:
52             return None
53     except Exception as e:
54         print(f"Error al obtener clima: {e}")
55         return None
56 # =====
57 # MÓDULO PARA OBTENER DATOS CLIMÁTICOS
58 # =====
59
60 import requests
61
62 def obtener_clima(ciudad):
63     """
64     Consulta el clima actual para una ciudad usando la API de
           OpenWeatherMap.
```



```

94         # En caso de respuesta inválida, retorna None
95         return None
96
97     except Exception as e:
98         # Captura cualquier excepción y la imprime para debugging
99         print(f"Error al obtener clima: {e}")
100        return None

```

Listing 2: Fragmento del archivo clima.py

## DASHBOARD

```

34
35
36 <!DOCTYPE html>
37 <html lang="es">
38 <head>
39     <meta charset="UTF-8" />
40     <title>Dashboard - Sistema de Detección de Roya</title>
41     <!-- Fuente Montserrat desde Google Fonts -->
42     <link href="https://fonts.googleapis.com/css2?family=Montserrat:
         wght@400;700&display=swap" rel="stylesheet" />
43     <!-- Estilos personalizados -->
44     <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
45     <style>
46         /* Estilos generales y de layout */
47         body {
48             font-family: 'Montserrat', sans-serif;
49             background: linear-gradient(120deg, #f0fdf4 60%, #d2f2ea 100%);
50             padding: 0;
51             margin: 0;
52             min-height: 100vh;
53         }
54

```

```
55     /* Contenedor header UTN principal */
56     .header-portada {
57         display: flex;
58         align-items: center;
59         justify-content: space-between;
60         gap: 28px;
61         margin-bottom: 32px;
62         margin-top: 0;
63         padding: 20px 0 14px 0;
64         background: rgba(255,255,255,0.97);
65         border-radius: 18px;
66         box-shadow: 0 4px 22px rgba(44, 62, 80, 0.09);
67     }
68     /* Logo UTN y telecom */
69     .header-logo-portada {
70         flex: 1;
71         display: flex;
72         align-items: center;
73         justify-content: center;
74     }
75     /* Estilo imagen logo */
76     .header-logo-portada img {
77         width: 120px;
78         height: 120px;
79         object-fit: contain;
80         background: #f8f8f8;
81         border-radius: 15px;
82         border: 1.5px solid #e0e0e0;
83         box-shadow: 0 1px 8px rgba(0,0,0,0.06);
84     }
85     /* Contenedor títulos en header */
86     .header-titulos-portada {
87         flex: 3;
88         text-align: center;
89         display: flex;
```

```
90         flex-direction: column;
91         align-items: center;
92         gap: 8px;
93     }
94     /* Título principal UTN */
95     .titulo-utn {
96         font-size: 1.9rem;
97         font-weight: 900;
98         color: #d3371b;
99         letter-spacing: 2px;
100        text-transform: uppercase;
101        margin-bottom: 0;
102        margin-top: 0;
103    }
104    /* Subtítulo carrera */
105    .subtitulo-carrera {
106        font-size: 1.25em;
107        color: #18467a;
108        font-weight: 700;
109        letter-spacing: 1.5px;
110        text-transform: uppercase;
111        margin-bottom: 0;
112    }
113    /* Texto autor y director */
114    .header-estudiante-portada {
115        font-size: 1.08rem;
116        font-weight: 500;
117        color: #363636;
118        letter-spacing: 0.3px;
119        margin-top: 7px;
120        margin-bottom: 0;
121        line-height: 1.8;
122    }
123    .header-estudiante-portada strong {
124        font-weight: 900;
```

```

125         color: #363636;
126     }
127     /* Responsive para pantallas pequeñas */
128     @media (max-width: 900px) {
129         .header-portada {
130             flex-direction: column;
131             gap: 14px;
132             padding: 14px 4px 10px 4px;
133         }
134         .header-logo-portada img {
135             width: 85px;
136             height: 85px;
137         }
138         .titulo-utn {
139             font-size: 1.15rem;
140         }
141         .subtitulo-carrera {
142             font-size: 1rem;
143         }
144     }
145 </style>
146 </head>
147 <body>
148     <div class="container">
149         <!-- ===== HEADER UTN ===== -->
150         <div class="header-portada">
151             <!-- Logo izquierdo UTN -->
152             <div class="header-logo-portada">
153                 
155             </div>
156             <!-- Títulos centrales -->
157             <div class="header-titulos-portada">
158                 <div class="titulo-utn">UNIVERSIDAD TÉCNICA DEL NORTE</div>

```

```

159         <div class="subtitulo-carrera">CARRERA DE INGENIERÍA EN
          TELECOMUNICACIONES</div>
160     <div class="header-estudiante-portada">
161         <strong>Autor:</strong> <span style="color:#353434;">
          Farid Avila</span><br>
162         <strong>Director:</strong> <span style="color:#353434;">
          >Msc. Carlos Vásquez</span><br>
163     </div>
164 </div>
165
166 <!-- Logo derecho Telecom -->
167 <div class="header-logo-portada">
168     
169 </div>
170 </div>
171 <!-- ===== FIN HEADER ===== -->
172
173 <!-- Título principal de la página -->
174 <h2 class="titulo-centrado">
175     SISTEMA INTELIGENTE PARA LA DETECCIÓN AUTOMATIZADA DE ROYA CON
          IA
176 </h2>
177
178 <!-- Formulario para crear nuevo registro de detección -->
179 <h2>CREAR NUEVO REGISTRO DE DETECCIÓN</h2>
180 <form method="POST" action="{[_url_for('dashboard')_]}">
181     <div class="form-clima-wrapper">
182         <!-- Columna izquierda del formulario -->
183         <div class="form-col">
184             <label for="registro_id"> ID de Registro</label>
185             <input type="text1" id="registro_id" name="registro_id"
          readonly autocomplete="off"/>
186

```

```
187     <label for="nombre_proyecto">Nombre del proyecto</label
      >
188     <input type="text" id="nombre_proyecto" name="
      nombre_proyecto" required autocomplete="off"/>
189
190     <label for="tipo_operacion"> Tipo de operación</label>
191     <select id="tipo_operacion" name="tipo_operacion"
      required>
192         <option value="registro"> Crear nuevo registro</
      option>
193     </select>
194 </div>
195
196 <!-- Columna central del formulario -->
197 <div class="form-col">
198     <label for="fecha_hora"> Fecha y hora</label>
199     <input type="datetime-local" id="fecha_hora" name="
      fecha_hora" readonly value="{{_fecha_ecuador_}}"/>
200
201     <label for="numero_invernadero"> Número de Invernadero
      </label>
202     <select id="numero_invernadero" name="
      numero_invernadero" required>
203         <option value="" disabled selected>Selecciona
      Invernadero</option>
204         <option value="Invernadero_1">Invernadero 1</option
      >
205         <option value="Invernadero_2">Invernadero 2</option
      >
206         <option value="Invernadero_3">Invernadero 3</option
      >
207         <option value="Invernadero_4">Invernadero 4</option
      >
208         <option value="Invernadero_5">Invernadero 5</option
      >
```

```

209         </select>
210
211         <label for="ciudad"> Ciudad / Zona monitoreada</label>
212         <input type="text1" id="ciudad" name="ciudad" value="{
                _ciudad_}" required autocomplete="off"/>
213     </div>
214
215     <!-- Columna derecha con datos climáticos -->
216     <div class="clima-box">
217         <h3> Datos climáticos</h3>
218         <div class="clima-item"> Temperatura: <span id="tempVal
                ">{{ temperatura }} °C</span></div>
219         <div class="clima-item"> Humedad: <span id="humVal">{{
                humedad }} %</span></div>
220         <div class="clima-item"> Descripción: <span id="descVal
                ">{{ descripcion }}</span></div>
221         <div class="clima-item"> Viento: <span id="vientoVal"
                >{{ viento }} m/s</span></div>
222     </div>
223 </div>
224
225     <!-- Botones para enviar o navegar -->
226     <div class="button-group">
227         <button class="submit-btn" type="submit"> Detección en
                Tiempo Real</button>
228         <a href="{_url_for('logout')_}" class="logout-btn">Cerrar
                sesión </a>
229         <a href="{_url_for('upload_index')_}" class="upload-btn">
                Cargar archivos</a>
230     </div>
231 </form>
232 </div>
233
234 <!-- SCRIPT PARA GENERAR ID Y ACTUALIZAR CLIMA DINÁMICAMENTE -->
235 <script>

```

```
236 window.onload = () => {
237     // Inputs del formulario a manipular
238     const idInput = document.getElementById('registro_id');
239     const ciudadInput = document.getElementById('ciudad');
240
241     // Función para generar un ID único tipo REG-timestamp-
242     aleatorio
243     const generarID = () => {
244         const timestamp = Date.now().toString(36);
245         const aleatorio = Math.floor(Math.random() * 1000).toString
246         (36);
247         return 'REG-' + timestamp + '-' + aleatorio;
248     };
249
250     // Setea el ID generado en el input
251     idInput.value = generarID();
252
253     // Función para actualizar datos climáticos desde API vía fetch
254     const actualizarClima = (ciudad) => {
255         fetch(`/api/clima?ciudad=${encodeURIComponent(ciudad)}`)
256         .then(res => res.json())
257         .then(data => {
258             if (!data.error) {
259                 document.getElementById('tempVal').innerText =
260                 `${data.temperatura.toFixed(1)} °C`;
261                 document.getElementById('humVal').innerText = `
262                 ${data.humedad.toFixed(1)} %`;
263                 document.getElementById('descVal').innerText =
264                 data.descripcion;
265                 document.getElementById('vientoVal').innerText
266                 = `${data.viento.toFixed(1)} m/s`;
267             }
268         })
269         .catch(err => console.error("Error al obtener clima:",
270             err));
271     };
272 }
```

```

264         };
265
266         // Actualiza clima al cargar la página y al cambiar la ciudad
267         actualizarClima(ciudadInput.value);
268         ciudadInput.addEventListener('change', e => actualizarClima(e.
                target.value));
269     };
270 </script>
271 </body>
272 </html>

```

Listing 3: Fragmento del archivo clima.py

## HISTORIAL

```

34
35 <!DOCTYPE html>
36 <html lang="es">
37 <head>
38     <meta charset="UTF-8" />
39     <title>Historial de Resultados</title>
40
41     <!-- Fuente Montserrat para estilo moderno y limpio -->
42     <link href="https://fonts.googleapis.com/css2?family=Montserrat:
         wght@400;700&display=swap" rel="stylesheet" />
43
44     <style>
45         /* ===== VARIABLES CSS ===== */
46         :root {
47             --primary: #27ae60;
48             --primary-dark: #219150;
49             --secondary: #fff;
50             --accent: #1abc9c;
51             --danger: #e74c3c;
52             --info: #2980b9;

```

```
53     --bg: #f9f9f9;
54     --border-radius: 22px;
55 }
56
57 /* ===== HEADER BANNER ===== */
58 .header-banner {
59     display: flex;
60     align-items: center;
61     justify-content: center;
62     gap: 36px;
63     background: linear-gradient(90deg, #27ae60 70%, #1abc9c 100%);
64     color: white;
65     border-radius: 0 0 30px 30px;
66     padding: 32px 0 22px 0;
67     box-shadow: 0 4px 18px rgba(44,62,80,0.11);
68 }
69 /* Logos con efectos hover */
70 .logo-img {
71     width: 70px;
72     height: 70px;
73     border-radius: 16px;
74     background: #fff;
75     box-shadow: 0 3px 18px rgba(0,0,0,0.13);
76     object-fit: contain;
77     padding: 8px;
78     transition: transform 0.2s, box-shadow 0.2s;
79     border: 2px solid #f0f0f0;
80 }
81 .logo-img:hover {
82     transform: scale(1.08) rotate(-3deg);
83     box-shadow: 0 8px 32px rgba(44,62,80,0.18);
84 }
85 /* Título principal */
86 .header-banner h1 {
87     margin: 0 18px;
```

```
88     font-size: 2.3em;
89     font-weight: 700;
90     letter-spacing: 2px;
91     text-shadow: 1px 2px 8px rgba(0,0,0,0.09);
92 }
93
94 /* Responsive para móviles */
95 @media (max-width: 700px) {
96     .header-banner {
97         flex-direction: column;
98         gap: 16px;
99         padding: 22px 0 12px 0;
100    }
101    .header-banner h1 {
102        font-size: 1.3em;
103    }
104    .logo-img {
105        width: 56px;
106        height: 56px;
107    }
108 }
109
110 /* ===== ESTILOS GENERALES ===== */
111 body {
112     font-family: 'Montserrat', sans-serif;
113     background: var(--bg);
114     padding: 0;
115     margin: 0;
116 }
117 .container {
118     max-width: 1600px;
119     margin: 38px auto 0 auto;
120     background: white;
121     padding: 36px 32px 32px 32px;
122     border-radius: var(--border-radius);
```

```
123         box-shadow: 0 0 32px rgba(44, 62, 80, 0.11);
124     }
125
126     /* Mini dashboard con estadísticas rápidas */
127     .mini-dashboard {
128         display: flex;
129         justify-content: space-between;
130         align-items: center;
131         margin-bottom: 24px;
132         gap: 10px;
133         flex-wrap: wrap;
134     }
135     .mini-dashboard .summary-box {
136         background: linear-gradient(90deg, #eafaf1, #d2f2ea 70%);
137         color: #26a65b;
138         border-radius: 18px;
139         padding: 18px 36px;
140         font-weight: bold;
141         font-size: 1.06em;
142         text-align: center;
143         box-shadow: 0 2px 8px rgba(44, 62, 80, 0.07);
144         min-width: 140px;
145     }
146
147     /* Títulos y filtros */
148     h2 {
149         text-align: center;
150         color: var(--primary);
151         font-size: 2em;
152         letter-spacing: 1px;
153     }
154     .filtros {
155         display: flex;
156         justify-content: space-between;
157         flex-wrap: wrap;
```

```
158         gap: 18px;
159         margin: 30px 0 18px 0;
160         position: sticky;
161         top: 0;
162         background: white;
163         z-index: 3;
164         padding: 14px 0 8px 0;
165     }
166     .filtros input,
167     .filtros select {
168         padding: 12px;
169         border-radius: 12px;
170         border: 1px solid #b4e2c9;
171         font-size: 1em;
172         min-width: 185px;
173         transition: border 0.2s;
174         outline: none;
175     }
176     .filtros input:focus,
177     .filtros select:focus {
178         border: 2px solid var(--primary);
179     }
180     /* Botón limpiar filtros */
181     .clear-btn {
182         background: var(--danger);
183         color: white;
184         border: none;
185         border-radius: 10px;
186         font-weight: 600;
187         padding: 9px 18px;
188         cursor: pointer;
189         margin-left: 12px;
190         transition: background 0.18s;
191     }
192     .clear-btn:hover {
```

```
193         background: #c0392b;
194     }
195     /* Texto cantidad resultados */
196     .count-results {
197         margin-left: auto;
198         color: #2980b9;
199         font-weight: 700;
200         font-size: 1.02em;
201         margin-top: 10px;
202     }
203
204     /* Tabla scrollable */
205     .tabla-scroll {
206         overflow-x: auto;
207         border-radius: 12px;
208         box-shadow: 0 2px 12px rgba(44,62,80,0.09);
209         margin-top: 16px;
210     }
211     table {
212         width: 100%;
213         border-collapse: separate;
214         border-spacing: 0;
215         min-width: 900px;
216         background: white;
217         margin: 0;
218     }
219     th, td {
220         padding: 13px 10px;
221         text-align: left;
222         vertical-align: middle;
223         border-bottom: 1px solid #e9e9e9;
224         white-space: nowrap;
225         font-size: 1em;
226     }
227     th {
```

```
228     position: sticky;
229     top: 0;
230     background: #27ae60;
231     color: #fff;
232     font-size: 1.07em;
233     z-index: 2;
234     border-bottom: 3px solid #1abc9c;
235     letter-spacing: 1px;
236 }
237 tr:nth-child(even) {
238     background-color: #f8fafb;
239 }
240
241 /* Estilos especiales para celdas */
242 td.resultado {
243     max-width: 240px;
244     overflow: hidden;
245     text-overflow: ellipsis;
246     white-space: nowrap;
247     color: #fff;
248     background: #e74c3c;
249     border-radius: 7px;
250     font-weight: 600;
251     font-size: 0.97em;
252     padding: 11px 8px;
253 }
254 td.resultado a {
255     color: #fff;
256     text-decoration: underline dotted;
257     cursor: pointer;
258 }
259 td.archivo-original {
260     max-width: 340px;
261     overflow: hidden;
262     text-overflow: ellipsis;
```

```
263         white-space: nowrap;
264         font-size: 0.98em;
265     }
266     td.usuario {
267         font-weight: bold;
268         color: #27ae60;
269         font-size: 1em;
270     }
271     td.nombre-personalizado {
272         font-style: italic;
273         color: #04866c;
274         font-size: 1.18em;
275         font-weight: bold;
276         max-width: 220px;
277         white-space: nowrap;
278         overflow: hidden;
279         text-overflow: ellipsis;
280     }
281
282     /* Etiquetas tipo chip */
283     .chip {
284         display: inline-block;
285         padding: 3px 15px;
286         border-radius: 50px;
287         background: #d2f2ea;
288         color: #27ae60;
289         font-size: 0.95em;
290         font-weight: 600;
291         margin: 0 2px;
292     }
293     .chip.tipo-video {
294         background: #e3edfa;
295         color: #2980b9;
296     }
297     .chip.tipo-imagen {
```

```
298     background: #ffe5e5;
299     color: #e74c3c;
300 }
301
302 /* Botón volver al menú */
303 .volver {
304     text-align: center;
305     margin-top: 32px;
306 }
307 .submit-btn {
308     margin-top: 15px;
309     padding: 12px 32px;
310     font-weight: bold;
311     border: none;
312     border-radius: 12px;
313     background-color: var(--primary);
314     color: white;
315     cursor: pointer;
316     text-decoration: none;
317     font-size: 1.08em;
318     box-shadow: 0 2px 12px rgba(44, 62, 80, 0.10);
319     transition: background 0.21s;
320 }
321 .submit-btn:hover {
322     background-color: var(--primary-dark);
323 }
324
325 /* Responsive */
326 @media (max-width: 900px) {
327     .container {
328         padding: 16px 5vw 20px 5vw;
329     }
330     .mini-dashboard {
331         flex-direction: column;
332     }
```

```
333     .header-banner {
334         flex-direction: column;
335         gap: 12px;
336         padding: 16px 0;
337     }
338 }
339 @media (max-width: 650px) {
340     .filtros {
341         flex-direction: column;
342     }
343     table, th, td {
344         font-size: 0.96em;
345     }
346     .mini-dashboard .summary-box {
347         padding: 8px 12px;
348         font-size: 0.98em;
349     }
350 }
351 </style>
352 </head>
353 <body>
354
355 <!-- ===== HEADER BANNER ===== -->
356 <div class="header-banner">
357     
359     <h1>Historial de Detección de Roya</h1>
360     
362 </div>
363
364 <!-- ===== CONTENIDO PRINCIPAL ===== -->
365 <div class="container">
```

```

365 <!-- Mini dashboard con estadísticas generales -->
366 <div class="mini-dashboard">
367     <div class="summary-box">Archivos: <span id="countArchivos">0</span
          ></div>
368     <div class="summary-box">Imágenes: <span id="countImágenes">0</span
          ></div>
369     <div class="summary-box">Videos: <span id="countVideos">0</span></
          div>
370     <div class="summary-box">Usuarios: <span id="countUsuarios">0</span
          ></div>
371 </div>
372
373 <!-- Filtros para búsqueda -->
374 <div class="filtros">
375     <input type="text" id="filtroUsuario" placeholder="└ Filtrar└por└
          usuario...">
376     <input type="date" id="filtroFecha">
377     <select id="filtroTipo">
378         <option value=""> Filtrar por tipo...</option>
379         <option value="imagen"> Imagen</option>
380         <option value="video"> Video</option>
381     </select>
382     <button class="clear-btn" onclick="limpiarFiltros()">Limpiar</
          button>
383     <span class="count-results" id="countResultados"></span>
384 </div>
385
386 <!-- Tabla scrollable con registros -->
387 <div class="tabla-scroll">
388     {% if registros %}
389     <table id="tablaHistorial">
390         <thead>
391             <tr>
392                 <th style="width:└160px;"> Fecha</th>
393                 <th style="width:└110px;"> Usuario</th>

```

```

394         <th style="width: 340px;"> Archivo Original</th>
395         <th style="width: 240px;"> Resultado</th>
396         <th style="width: 180px;"> Nombre Personalizado</th>
397         <th style="width: 90px;"> Tipo</th>
398     </tr>
399 </thead>
400 <tbody>
401 {% for r in registros %}
402 <tr>
403     <td>{{ r.fecha_hora }}</td>
404     <td class="usuario">{{ r.usuario }}</td>
405     <td class="archivo-original" title="{{ ur.archivo_original_
406         }}">{{ r.archivo_original }}</td>
407     <td class="resultado" title="{{ ur.resultado_ }}">
408         <a href="{{ url_for('obtener_resultado', filename=r.
409             resultado_ }}" target="_blank">
410             {{ r.resultado }}
411         </a>
412     </td>
413     <td class="nombre-personalizado" title="{{ ur.
414         nombre_personalizado_ }}">{{ r.nombre_personalizado }}</
415         td>
416     <td>
417         <span class="chip_tipo-{{ ur.tipo_ }}">{{ r.tipo |
418             capitalize }}</span>
419     </td>
420 </tr>
421 {% endfor %}
422 </tbody>
423 </table>
424 {% else %}
425 <p style="text-align: center; font-size: 18px;"> No hay registros
426     disponibles aún.</p>
427 {% endif %}
428 </div>

```

```
423
424 <!-- Botón para volver al menú principal -->
425 <div class="volver">
426     <a href="{url_for('upload_index')}" class="submit-btn">Volver
427         al Menú Principal</a>
428 </div>
429
430 <!-- ===== SCRIPT DE FILTRADO Y ESTADÍSTICAS
431     ===== -->
432 <script>
433     // Referencias a inputs y tabla
434     const filtroUsuario = document.getElementById("filtroUsuario");
435     const filtroFecha = document.getElementById("filtroFecha");
436     const filtroTipo = document.getElementById("filtroTipo");
437     const tabla = document.getElementById("tablaHistorial")?.
438         getElementsByTagName("tbody")[0];
439     const countResultados = document.getElementById("countResultados");
440
441     // Limpia todos los filtros y aplica filtro sin restricciones
442     function limpiarFiltros() {
443         filtroUsuario.value = '';
444         filtroFecha.value = '';
445         filtroTipo.value = '';
446         aplicarFiltros();
447     }
448
449     // Aplica los filtros sobre las filas de la tabla
450     function aplicarFiltros() {
451         if (!tabla) return;
452         let count = 0;
453         for (const fila of tabla.rows) {
454             // Obtiene valores de usuario, fecha y tipo de la fila
455             const usuarioFila = fila.cells[1].textContent.toLowerCase();
456             const fechaFila = fila.cells[0].textContent;
```

```
455     const tipoFila = fila.cells[5].textContent.toLowerCase();
456
457     // Verifica coincidencia con filtros (case insensitive)
458     const coincideUsuario = filtroUsuario.value === "" ||
         usuarioFila.includes(filtroUsuario.value.toLowerCase());
459     const coincideFecha = filtroFecha.value === "" || fechaFila.
         startsWith(filtroFecha.value);
460     const coincideTipo = filtroTipo.value === "" || tipoFila.
         includes(filtroTipo.value.toLowerCase());
461
462     // Muestra u oculta fila según filtro
463     const visible = coincideUsuario && coincideFecha &&
         coincideTipo;
464     fila.style.display = visible ? "" : "none";
465
466     if (visible) count++;
467 }
468 // Actualiza contador de resultados visibles
469 countResultados.innerHTML = `Resultados mostrados: ${count}`;
470 }
471
472 // Actualiza mini-dashboard con estadísticas de la tabla
473 function actualizarMiniDashboard() {
474     if (!tabla) return;
475     let total = 0, imgs = 0, vids = 0;
476     let usuariosSet = new Set();
477
478     for (const fila of tabla.rows) {
479         total++;
480         let tipo = fila.cells[5].textContent.trim().toLowerCase();
481         if (tipo.includes('imagen')) imgs++;
482         if (tipo.includes('video')) vids++;
483         usuariosSet.add(fila.cells[1].textContent.trim());
484     }
485     document.getElementById("countArchivos").innerHTML = total;
```

```

486     document.getElementById("countImágenes").innerText = imgs;
487     document.getElementById("countVideos").innerText = vids;
488     document.getElementById("countUsuarios").innerText = usuariosSet.
         size;
489     countResultados.innerText = `Resultados mostrados: ${total}`;
490 }
491
492 // Inicializa contadores y filtros al cargar página
493 window.onload = function() {
494     actualizarMiniDashboard();
495     aplicarFiltros();
496 };
497
498 // Eventos para aplicar filtros dinámicamente
499 filtroUsuario?.addEventListener("input", aplicarFiltros);
500 filtroFecha?.addEventListener("change", aplicarFiltros);
501 filtroTipo?.addEventListener("change", aplicarFiltros);
502 </script>
503
504 </body>
505 </html>

```

Listing 4: Fragmento del archivo clima.py

**login**

```

34
35 <!DOCTYPE html>
36 <html lang="es">
37 <head>
38     <meta charset="UTF-8" />
39     <title>Login - Detección de Roya</title>
40     <style>
41         /* General */
42         body {

```

```
43     margin: 0;
44     padding: 0;
45     font-family: 'Montserrat', sans-serif;
46     background: linear-gradient(135deg, #8c5bc9 0%, #1b6ecc 100%);
47     height: 100vh;
48     display: flex;
49     justify-content: center;
50     align-items: center;
51 }
52
53 /* Contenedor principal */
54 .container1 {
55     background: white;
56     padding: 40px 35px 35px 35px;
57     border-radius: 25px;
58     box-shadow: 0 12px 25px rgba(52, 56, 59, 0.1);
59     width: 360px;
60     text-align: center;
61     transition: transform 0.3s ease;
62 }
63
64 .container1:hover {
65     transform: translateY(-5px);
66 }
67
68 /* Título */
69 .container1 h2 {
70     margin-bottom: 20px;
71     font-weight: 900;
72     color: #2f363d;
73     font-size: 1.8rem;
74     letter-spacing: 1.3px;
75     text-transform: uppercase;
76 }
77
```

```
78     /* Descripción */
79     .container1 p {
80         color: #696c70;
81         font-weight: 600;
82         margin-bottom: 30px;
83         font-size: 1.05rem;
84     }
85
86     /* Inputs */
87     input[type="text"], input[type="password"],
88     .campo-usuario, .campo-clave {
89         width: 100%;
90         padding: 14px 18px;
91         margin: 15px 0;
92         border: 2px solid #c4cee2;
93         border-radius: 18px;
94         font-size: 1.1rem;
95         font-weight: 600;
96         transition: border-color 0.3s ease, box-shadow 0.3s ease;
97         box-shadow: inset 0 0 8px #edd3ef;
98         color: #6e676a;
99         box-sizing: border-box;
100    }
101
102    input[type="text"]:focus, input[type="password"]:focus,
103    .campo-usuario:focus, .campo-clave:focus {
104        border-color: #cae6e1;
105        box-shadow: 0 0 12px #c6d1c688;
106        outline: none;
107    }
108
109    /* Botón ingresar */
110    input[type="submit"] {
111        width: 100%;
112        padding: 14px 0;
```

```
113     margin-top: 25px;
114     border: none;
115     border-radius: 22px;
116     font-weight: 900;
117     font-size: 1.15rem;
118     color: white;
119     background: linear-gradient(90deg, #0aceb4 0%, #4287d6 90%);
120     cursor: pointer;
121     box-shadow: 0 5px 15px #5d5be088;
122     transition: background 0.25s ease, transform 0.2s ease;
123 }
124
125 input[type="submit"]:hover {
126     background: linear-gradient(90deg, #2e8ac0 0%, #3789e7 90%);
127     transform: scale(1.03);
128 }
129
130 /* Flash messages */
131 .flash1 {
132     margin: 20px 0;
133     padding: 15px 20px;
134     border-radius: 18px;
135     font-weight: 700;
136     font-size: 1rem;
137     letter-spacing: 0.6px;
138     user-select: none;
139 }
140
141 .flash1.error {
142     background: #95a4e5;
143     color: #721c24;
144     border: 1.8px solid #aec5e7;
145     box-shadow: 0 2px 8px rgba(255, 0, 0, 0.15);
146 }
147
```

```
148     .flash1.success {
149         background: #f7f7f7;
150         color: #2f6ece;
151         border: 1.8px solid #8ba6f0;
152         box-shadow: 0 2px 8px rgba(52, 91, 136, 0.15);
153     }
154
155     /* Toggle password icon */
156     .password-wrapper {
157         position: relative;
158         width: 100%;
159     }
160
161     #togglePassword {
162         position: absolute;
163         right: 16px;
164         top: 50%;
165         transform: translateY(-50%);
166         cursor: pointer;
167         user-select: none;
168         font-size: 1.2rem;
169         color: #1b8796;
170     }
171
172     /* Responsive */
173     @media (max-width: 400px) {
174         .container1 {
175             width: 90vw;
176             padding: 30px 25px;
177         }
178         input[type="submit"] {
179             font-size: 1rem;
180             padding: 12px 0;
181         }
182     }
```

```
183     </style>
184 </head>
185 <body>
186
187 <div class="container1">
188     <h2> Detección de Roya con Drones</h2>
189     <p>Accede al sistema inteligente de monitoreo para cultivos de rosas.</
190     p>
191     {% with messages = get_flashed_messages(with_categories=true) %}
192     {% if messages %}
193         {% for category, message in messages %}
194             <div class="flash1_{{_category_}}">{{ message }}</div>
195         {% endfor %}
196     {% endif %}
197 {% endwith %}
198
199     <form method="POST" novalidate>
200         <input type="text" id="username" name="username" placeholder="
201         Usuario" class="campo-usuario" required>
202
203         <div class="password-wrapper">
204             <input type="password" id="password" name="password"
205             placeholder="Contraseña" class="campo-clave" required>
206
207             <span id="togglePassword"></span>
208         </div>
209
210         <input type="submit" value="Ingresar">
211     </form>
212 </div>
213
214 <script>
215     const togglePassword = document.getElementById('togglePassword');
216     const passwordInput = document.getElementById('password');
```

```
215 togglePassword.addEventListener('click', () => {
216     if(passwordInput.type === 'password'){
217         passwordInput.type = 'text';
218         togglePassword.textContent = '';
219     } else {
220         passwordInput.type = 'password';
221         togglePassword.textContent = '';
222     }
223 });
224 </script>
225
226 </body>
227 </html>
```

Listing 5: Fragmento del archivo clima.py

## plantilla pdf

```
34
35 <!DOCTYPE html>
36 <html lang="es">
37 <head>
38     <meta charset="UTF-8" />
39     <title>Reporte de Detección de Roya</title>
40     <style>
41         /* Estilos básicos y fuentes */
42         body {
43             font-family: 'Arial', sans-serif;
44             margin: 20px;
45             color: #2c3e50;
46             background-color: #fdfefe;
47         }
48         /* Títulos principales */
49         h1 {
50             text-align: center;
```

```
51     color: #2c3e50;
52     font-size: 26px;
53     margin-bottom: 5px;
54 }
55 h2 {
56     font-size: 18px;
57     color: #34495e;
58     border-bottom: 2px solid #7AD7F0;
59     padding-bottom: 4px;
60     margin-top: 20px;
61     display: flex;
62     align-items: center;
63     gap: 8px;
64 }
65 /* Secciones con fondo y borde */
66 .section {
67     background-color: #f4f9fb;
68     border: 1px solid #d5eaf3;
69     border-left: 5px solid #7AD7F0;
70     padding: 15px 20px;
71     margin-bottom: 25px;
72     border-radius: 6px;
73 }
74 /* Texto y listas */
75 p, li {
76     font-size: 13px;
77     line-height: 1.6;
78 }
79 ul {
80     padding-left: 20px;
81 }
82 /* Tablas con borde y colores suaves */
83 table {
84     width: 100%;
85     border-collapse: collapse;
```

```
86         margin-top: 10px;
87     }
88     th, td {
89         border: 1px solid #dbeef5;
90         padding: 6px;
91         font-size: 12px;
92         text-align: center;
93     }
94     th {
95         background-color: #eaf9fd;
96         color: #2c3e50;
97     }
98     /* Alertas y ok visuales */
99     .alerta {
100         color: #e74c3c;
101         font-weight: bold;
102     }
103     .ok {
104         color: #27ae60;
105         font-weight: bold;
106     }
107     /* Footer con texto pequeño */
108     .footer {
109         text-align: center;
110         margin-top: 40px;
111         font-size: 12px;
112         color: #777;
113     }
114     /* Contenedor gráfico */
115     .grafico {
116         text-align: center;
117         margin-top: 20px;
118     }
119     .grafico img {
120         width: 80%;
```

```
121         max-width: 600px;
122         border: 1px solid #ccc;
123         border-radius: 8px;
124     }
125     /* Íconos en títulos */
126     .icono-h2 {
127         width: 20px;
128         height: 20px;
129         display: inline-block;
130         vertical-align: middle;
131     }
132 </style>
133 </head>
134 <body>
135
136     <!-- Universidad y carrera arriba centrado -->
137     <p style="text-align:center; font-size: 14px; line-height:1.7;">
138         <span style="font-weight:bold;">UNIVERSIDAD TÉCNICA DEL NORTE</
139         span><br>
140         <span>Carrera de Ingeniería en Telecomunicaciones</span>
141     </p>
142     <!-- Título principal con ícono SVG -->
143     <h1>
144         <span style="vertical-align:middle;">
145             <svg class="icono-h2" viewBox="0 0 24 24" fill="#43b37c"><
146                 circle cx="12" cy="12" r="10"/><path d="M12 17v-6M12 7h.01"
147                 stroke="#fff" stroke-width="2" stroke-linecap="round"/></svg
148             >
149         </span>
150         REPORTE TÉCNICO DE LA DETECCIÓN DE ROYA
151     </h1>
152     <!-- Subtítulo centrado -->
```

```

151 <p style="text-align:center; font-size: 12px; line-height:1.7; margin-
152     bottom: 10px; margin-top: 0;">
153     <span style="font-weight: bold;">
154         SISTEMA INTELIGENTE PARA LA DETECCIÓN AUTOMATIZADA DE ROYA CON
155         IA
156     </span>
157 </p>
158 <!-- Sección Datos del Registro -->
159 <div class="section">
160     <h2>
161         <!-- Icono SVG -->
162         <svg class="icono-h2" viewBox="0 0 24 24" fill="#0ab9e7"><rect
163             x="4" y="4" width="16" height="16" rx="3" /><path d="M9 8
164             h6M9 12h6M9 16h2" stroke="#fff" stroke-width="2"/></svg>
165         Datos del Registro
166     </h2>
167     <p><strong>ID del Registro:</strong> {{ id }}</p>
168     <p><strong>Fecha y hora:</strong> {{ fecha_hora }}</p>
169     <p><strong>Zona monitoreada:</strong> {{ zona }}</p>
170     <p><strong>Tipo de operación:</strong> {{ tipo_operacion }}</p>
171     <p><strong>Temperatura:</strong> {{ temperatura }} °C</p>
172     <p><strong>Humedad relativa:</strong> {{ humedad }} %</p>
173 </div>
174 <!-- Sección Condiciones climáticas y riesgo -->
175 <div class="section">
176     <h2>
177         <svg class="icono-h2" viewBox="0 0 24 24" fill="#8ed6fb"><
178             ellipse cx="12" cy="12" rx="10" ry="8"/><path d="M8 16 12 7 4
179             7" stroke="#3498db" stroke-width="1.5"/></svg>
180         Condiciones Climáticas y Riesgo
181     </h2>
182     <p><strong>Temperatura actual:</strong> {{ temperatura }} °C -
183     <br>
184     <strong>{% if 15 <= temperatura <= 24 %}</strong>

```

```

180         <span style="color:␣#e67e22;"><strong>[!]</strong> Riesgo
           alto: condiciones propicias para la roya.</span>
181     {% else %}
182         <span class="ok" >[] Temperatura poco favorable para el
           hongo.</span>
183     {% endif %}
184 </p>
185 <p><strong>Humedad relativa:</strong> {{ humedad }} % -
186     {% if humedad >= 85 %}
187         <span style="color:␣#e67e22;"><strong>[!]</strong> Alta
           humedad: favorece el desarrollo de roya.</span>
188     {% else %}
189         <span class="ok" >[] Humedad controlada. Menor riesgo de
           infección.</span>
190     {% endif %}
191 </p>
192 {% if 15 <= temperatura <= 24 and humedad >= 85 %}
193     <p class="alerta">[!!!] Alerta máxima: condiciones muy
           favorables para la propagación de roya. Actúe inmediatamente
           .</p>
194 {% else %}
195     <p class="ok" >[] Las condiciones actuales no favorecen la
           propagación de roya. Se recomienda vigilancia preventiva.</p>
           >
196     {% endif %}
197 </div>
198
199 <!-- Sección Resumen histórico -->
200 <div class="section">
201     <h2>
202         <svg class="icono-h2" viewBox="0␣0␣24␣24" fill="#a067fa"><rect
           x="4" y="4" width="16" height="16" rx="3" /><path d="M7␣1413
           -3␣4␣4" stroke="#fff" stroke-width="2"/><circle cx="9" cy="9
           " r="1" fill="#fff"/></svg>
203     Resumen Histórico (últimos 3 registros)

```

```

204     </h2>
205     <table>
206         <tr><th>Fecha</th><th>Estado</th><th>Promedio detecciones/frame
                </th></tr>
207         {% for reg in ultimos_registros %}
208         <tr>
209             <td>{{ reg.fecha_hora }}</td>
210             <td>{{ reg.estado if reg.estado else 'N/A' }}</td>
211             <td>{{ reg.avg if reg.avg else 'N/A' }}</td>
212         </tr>
213         {% endfor %}
214     </table>
215     <p><strong>Tendencia general:</strong> {{ tendencia }}</p>
216 </div>
217
218 <!-- Sección gráfico -->
219 <div class="section_grafico">
220     <h2>
221         <svg class="icono-h2" viewBox="0 0 24 24" fill="#4d97fe"><rect
                x="3" y="3" width="18" height="18" rx="3"/><path d="M7 17
                V7M12 17V13M17 17V10" stroke="#fff" stroke-width="2"/></svg>
222         Gráfico de Evolución
223     </h2>
224     
225 </div>
226
227 <!-- Sección estado general -->
228 <div class="section">
229     <h2>
230         <svg class="icono-h2" viewBox="0 0 24 24" fill="#43b37c"><path
                d="M12 2C7 2 2 8 12 22C22 8 17 2 12 2Z"/><circle cx="12" cy=
                "8.5" r="3.5" fill="#fff"/></svg>
231         Estado General del Cultivo
232     </h2>

```

```

233     <p><strong>Estado actual:</strong> {{ estado }}</p>
234     <p><strong>Promedio de detecciones por frame:</strong> {{ avg }}</p>
      >
235 </div>
236
237 <!-- Sección análisis detección -->
238 <div class="section">
239     <h2>
240         <svg class="icono-h2" viewBox="0 0 24 24" fill="#2c3e50"><
              circle cx="12" cy="12" r="10"/><path d="M16 12a4 4 0 1 1-8 0
              4 4 0 0 1 8 0z" fill="#fff"/></svg>
241         Análisis de Detección
242     </h2>
243     <p><strong>Frames procesados:</strong> {{ frames }}</p>
244     <p><strong>Total de detecciones:</strong> {{ detecciones }}</p>
245 </div>
246
247 <!-- Sección recomendaciones -->
248 <div class="section">
249     <h2>
250         <svg class="icono-h2" viewBox="0 0 24 24" fill="#ffe066"><
              circle cx="12" cy="12" r="10"/><path d="M9 14l3-3 2 2"
              stroke="#4d97fe" stroke-width="2"/></svg>
251         Recomendaciones Técnicas
252     </h2>
253     {% if estado == "Grave" %}
254         <ul>
255             <li>Aplicar fungicida sistémico (2.5 L/ha) cada 7 días.</li>
              >
256             <li>Monitoreo diario del cultivo.</li>
257             <li>Eliminar hojas gravemente afectadas.</li>
258         </ul>
259     {% elif estado == "Crítico" %}
260         <ul>

```

```

261         <li>Aplicar fungicida de contacto (2 L/ha) cada 10 días.</
           li>
262         <li>Revisar zonas críticas cada 48 horas.</li>
263     </ul>
264     {% elif estado == "Leve" %}
265     <ul>
266         <li>Aplicar fungicida preventivo (1.5 L/ha) cada 15 días.</
           li>
267         <li>Monitoreo semanal.</li>
268     </ul>
269     {% else %}
270     <ul>
271         <li>No se ha detectado roya. Mantener monitoreo quincenal
           .</li>
272         <li>Evitar humedad alta persistente.</li>
273     </ul>
274     {% endif %}
275 </div>
276
277 <!-- Sección información sobre roya -->
278 <div class="section">
279     <h2>
280         <svg class="icono-h2" viewBox="0 0 24 24" fill="#7AD7F0"><rect
           x="4" y="4" width="16" height="16" rx="3"/><path d="M8 8
           h8v8H8z" fill="#fff"/></svg>
281         Información sobre la Roya
282     </h2>
283     <p>La <strong>roya</strong> (<em>Phragmidium mucronatum</em>) es
           una enfermedad fúngica que afecta a los rosales, causando
           pérdida de hojas, menor floración y debilitamiento general de la
           planta.</p>
284     <p><strong>Medidas clave:</strong></p>
285     <ul>
286         <li>Evitar exceso de humedad ambiental.</li>
287         <li>Eliminar restos vegetales infectados.</li>

```

```

288         <li>Usar fungicidas según estado de infección.</li>
289         <li>Proporcionar buena ventilación entre plantas.</li>
290     </ul>
291 </div>
292
293 <!-- Pie de página -->
294 <div class="footer">
295     Universidad Técnica del Norte | Carrera de Ingeniería en
296     Telecomunicaciones | Año 2025
297 </div>
298 </body>
299 </html>

```

Listing 6: Fragmento del archivo clima.py

## Stream

```

34
35 <!DOCTYPE html>
36 <html lang="es">
37 <head>
38     <meta charset="UTF-8" />
39     <title>Transmisión - Detección de Roya</title>
40     <!-- Enlace a CSS y fuentes -->
41     <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
42     <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&display=swap" rel="stylesheet">
43     <!-- Librería Chart.js para gráficos -->
44     <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
45
46     <style>
47         /* Estilos generales de la página */
48         body {

```

```
49     font-family: 'Montserrat', sans-serif;
50     background: #f0fdf4;
51     margin: 0;
52     padding: 0;
53 }
54 .container {
55     max-width: 900px;
56     margin: 30px auto;
57     padding: 30px;
58     background: white;
59     border-radius: 20px;
60     box-shadow: 0 6px 20px rgba(0,0,0,0.15);
61     text-align: center;
62 }
63 /* Estadísticas de frames, detecciones, etc. */
64 .stat-item {
65     background: #e8f5e9;
66     padding: 15px;
67     border-radius: 10px;
68     margin: 10px;
69     display: inline-block;
70     width: 42%;
71 }
72 /* Canvas para gráfico */
73 canvas {
74     margin-top: 20px;
75     border-radius: 10px;
76 }
77 /* Bloques de información */
78 .info-block {
79     background: #e0f5e9;
80     padding: 20px;
81     margin-top: 20px;
82     border-radius: 12px;
83     text-align: left;
```

```
84     }
85     /* Botones principales */
86     .button {
87         background-color: #1069b3;
88         color: white;
89         padding: 12px 20px;
90         margin: 10px 5px 0 5px;
91         border: none;
92         border-radius: 10px;
93         font-weight: bold;
94         cursor: pointer;
95         transition: background 0.2s, color 0.2s;
96     }
97     /* Botones deshabilitados */
98     .button:disabled {
99         background-color: #cccccc !important;
100        color: #666666 !important;
101        cursor: not-allowed;
102        opacity: 0.7;
103    }
104    /* Pantalla de carga */
105    #loadingScreen {
106        position: fixed;
107        top: 0; left: 0;
108        width: 100vw;
109        height: 100vh;
110        background: rgba(240, 253, 244, 0.98);
111        display: flex;
112        flex-direction: column;
113        justify-content: center;
114        align-items: center;
115        z-index: 9999;
116    }
117    /* Spinner animado */
118    .spinner {
```

```
119         width: 60px;
120         height: 60px;
121         border: 6px solid #e2f4e3;
122         border-top: 6px solid #2d572c;
123         border-radius: 50%;
124         animation: spin 1s linear infinite;
125         margin-bottom: 20px;
126     }
127     /* Animación giro */
128     @keyframes spin {
129         0% { transform: rotate(0deg); }
130         100% { transform: rotate(360deg); }
131     }
132     /* Texto de carga */
133     #loadingText {
134         font-weight: bold;
135         font-size: 1.2em;
136         color: #4c572c;
137     }
138 </style>
139 </head>
140 <body>
141
142 <!-- Pantalla de carga visible mientras inicializa -->
143 <div id="loadingScreen">
144     <div class="spinner"></div>
145     <p id="loadingText">Inicializando el detector de roya...</p>
146 </div>
147
148 <!-- Contenedor principal -->
149 <div class="container">
150     <h2> Transmisión en Vivo - Detector de Roya</h2>
151
152     <!-- Imagen con video feed (inicialmente oculta) -->
```

```

153 <img id="videoFeed" style="width:100%;border-radius:12px;display:
      none;" />
154
155 <!-- Estadísticas dinámicas -->
156 <div>
157     <div class="stat-item"><strong>Frames:</strong> <span id="
          total_frames">0</span></div>
158     <div class="stat-item"><strong>Detecciones:</strong> <span id="
          total_detections">0</span></div>
159     <div class="stat-item"><strong>FPS:</strong> <span id="fps">0</span
          ></div>
160     <div class="stat-item"><strong>Promedio:</strong> <span id="
          avg_detections">0</span></div>
161 </div>
162
163 <!-- Gráfico con Chart.js -->
164 <canvas id="statsChart" width="800" height="300"></canvas>
165
166 <!-- Botones de control -->
167 <div>
168     <button class="button" id="btnPause" onclick="pausarStreaming()">
          Pausar Streaming</button>
169     <button class="button" id="btnPlay" onclick="reanudarStreaming()"
          disabled>Reanudar Streaming</button>
170     <button class="button" id="btnRestart" onclick="reiniciarDetector()"
          "> Reiniciar Detector</button>
171     <!-- Botón snapshot eliminado -->
172     <button class="button" id="btnStop" onclick="detenerTransmision()">
          Detener Transmisión</button>
173     <button class="button" id="btnMostrar" onclick="mostrarResultados()"
          " disabled> Mostrar Resultados</button>
174     <button class="button" id="btnVolver" onclick="volverAlDashboard()"
          disabled > Volver al Dashboard</button>
175 </div>
176

```

```

177 <!-- Información y recomendaciones del cultivo -->
178 <div class="info-block" id="resultadoCultivo" style="display:none;">
179     <h3> Estado del Cultivo</h3>
180     <p id="estadoCultivo"></p>
181     <h4> Recomendaciones:</h4>
182     <ul id="recomendaciones"></ul>
183     <h4> Uso de fungicida:</h4>
184     <p id="usoFungicida"></p>
185 </div>
186
187 <!-- Información detallada del registro -->
188 <div class="info-block">
189     <h3> Información del Registro</h3>
190     <p><strong> ID:</strong> {{ registro.id }}</p>
191     <p><strong> Fecha:</strong> {{ registro.fecha_hora }}</p>
192     <p><strong> Proyecto:</strong> {{ registro.nombre_proyecto }}</p>
193     <p><strong> Invernadero:</strong> {{ registro.numero_invernadero
194         }}</p>
195     <p><strong> Zona:</strong> {{ registro.zona }}</p>
196     <p><strong> Tipo:</strong> {{ registro.tipo_operacion }}</p>
197     <hr style="border-top:1px solid #b1d9b2;margin:15px 0;">
198     <h4> Condiciones Climáticas</h4>
199     <p><strong> Temperatura:</strong> {{ registro.temperatura }} °C</p>
200     <p><strong> Humedad:</strong> {{ registro.humedad }} %</p>
201     <p><strong> Descripción:</strong> {{ registro.descripcion_clima
202         }}</p>
203     <p><strong> Viento:</strong> {{ registro.viento }} m/s</p>
204
205 <!-- Formulario para generar PDF -->
206 <form id="formGenerarPDF" method="POST" action="/generar_pdf"
207     target="_blank">
208     <input type="hidden" name="id" value="{{_registro.id}}">
209     <input type="hidden" name="fecha_hora" value="{{_registro.
210         fecha_hora}}">
211     <input type="hidden" name="zona" value="{{_registro.zona}}">

```

```
208     <input type="hidden" name="tipo_operacion" value="{{_registro.
        tipo_operacion_}}">
209     <input type="hidden" name="nombre_proyecto" value="{{_registro.
        nombre_proyecto_}}">
210     <input type="hidden" name="numero_invernadero" value="{{_
        registro.numero_invernadero_}}">
211     <input type="hidden" name="estado" id="estado_final_pdf">
212     <input type="hidden" name="frames" id="frames_final_pdf">
213     <input type="hidden" name="detecciones" id="
        detecciones_final_pdf">
214     <input type="hidden" name="avg" id="avg_final_pdf">
215     <input type="hidden" name="temperatura" value="{{_registro.
        temperatura_}}">
216     <input type="hidden" name="humedad" value="{{_registro.humedad_
        }}">
217     <input type="hidden" name="descripcion_clima" value="{{_
        registro.descripcion_clima_}}">
218     <input type="hidden" name="viento" value="{{_registro.viento_}}
        ">
219     <button type="submit" class="button"> Generar PDF</button>
220 </form>
221 </div>
222 </div>
223
224 <script>
225     // Variables globales para controlar estado streaming y estadísticas
226     let intervalID;
227     let statsFinales = {};
228     let streamingPaused = false;
229     let streamingStopped = false;
230
231     // Oculta la pantalla de carga y muestra el video
232     function ocultarLoading() {
233         document.getElementById('loadingScreen').style.display = 'none';
234         document.getElementById('videoFeed').style.display = 'block';
```



```
267
268         const now = new Date().toLocaleTimeString();
269         if (statsChart.data.labels.length > 20) {
270             statsChart.data.labels.shift();
271             statsChart.data.datasets.forEach(ds => ds.data.
                shift());
272         }
273
274         statsChart.data.labels.push(now);
275         statsChart.data.datasets[0].data.push(data.total_frames
            );
276         statsChart.data.datasets[1].data.push(data.
            total_detections);
277         statsChart.data.datasets[2].data.push(data.fps);
278         statsChart.data.datasets[3].data.push(data.
            avg_detections_per_frame);
279         statsChart.update();
280
281         actualizarRecomendaciones(data.avg_detections_per_frame
            );
282     }
283 });
284 }
285
286 // Actualiza las recomendaciones y estado del cultivo según promedio
    detecciones
287 function actualizarRecomendaciones(avg) {
288     let estado = '';
289     let recomendaciones = [];
290     let uso = '';
291
292     if (avg >= 5) {
293         estado = 'Grave';
294         recomendaciones = ['Aplicar fungicida sistémico', 'Monitorear
                diario', 'Eliminar hojas enfermas'];
```

```
295     uso = '2.5L/ha_cada_7_días';
296 } else if (avg >= 2) {
297     estado = 'Crítico';
298     recomendaciones = ['Fungicida_de_contacto', 'Monitoreo_
        frecuente'];
299     uso = '2L/ha_cada_10_días';
300 } else if (avg > 0) {
301     estado = 'Leve';
302     recomendaciones = ['Fungicida_preventivo', 'Monitoreo_semanal'
        ];
303     uso = '1.5L/ha_cada_15_días';
304 } else {
305     estado = 'Sin_roya_detectada';
306     recomendaciones = ['Mantener_monitoreo_regular'];
307     uso = 'No_aplicar_fungicida';
308 }
309
310 // Actualiza DOM con estado y recomendaciones
311 document.getElementById('estadoCultivo').textContent = estado;
312 const lista = document.getElementById('recomendaciones');
313 lista.innerHTML = '';
314 recomendaciones.forEach(r => {
315     const li = document.createElement('li');
316     li.textContent = r;
317     lista.appendChild(li);
318 });
319
320 document.getElementById('usoFungicida').textContent = uso;
321
322 // Actualiza campos ocultos para el PDF
323 document.getElementById('estado_final_pdf').value = estado;
324 document.getElementById('frames_final_pdf').value = statsFinales.
    total_frames;
325 document.getElementById('detecciones_final_pdf').value =
    statsFinales.total_detections;
```

```
326     document.getElementById('avg_final_pdf').value = statsFinales.  
        avg_detections_per_frame;  
327 }  
328  
329 // Pausa el streaming y limpia la fuente del video  
330 function pausarStreaming() {  
331     document.getElementById('videoFeed').src = '';  
332     streamingPaused = true;  
333     clearInterval(intervalID);  
334     actualizarBotones();  
335 }  
336  
337 // Reanuda el streaming y reinicia actualización de stats  
338 function reanudarStreaming() {  
339     document.getElementById('videoFeed').src = '{{_url_for("video_feed  
        ")_}}';  
340     streamingPaused = false;  
341     streamingStopped = false;  
342     intervalID = setInterval(actualizarStats, 2000);  
343     actualizarBotones();  
344 }  
345  
346 // Reinicia el detector en el backend y resetea la gráfica  
347 function reiniciarDetector() {  
348     fetch('/restart', { method: 'POST' })  
349         .then(res => res.json())  
350         .then(data => {  
351             alert(data.status || data.error);  
352             if (!streamingPaused && !streamingStopped) {  
353                 document.getElementById('videoFeed').src = '{{_url_for  
                    ("video_feed")_}}?' + new Date().getTime();  
354             }  
355             resetChart();  
356             // Habilita botones Mostrar Resultados y Volver  
357             document.getElementById('btnMostrar').disabled = false;
```

```
358         document.getElementById('btnVolver').disabled = false;
359         actualizarBotones();
360     })
361     .catch(() => alert('Error al reiniciar detector'));
362 }
363
364 // Detiene la transmisión, limpia y habilita botones
365 function detenerTransmision() {
366     fetch('/finalizar_stream', { method: 'POST' })
367     .then(res => {
368         if (res.ok) {
369             alert('Transmisión detenida correctamente');
370             document.getElementById('videoFeed').src = '';
371             streamingPaused = false;
372             streamingStopped = true;
373             clearInterval(intervalID);
374             // Habilita botones Mostrar Resultados y Volver
375             document.getElementById('btnMostrar').disabled = false;
376             document.getElementById('btnVolver').disabled = false;
377             actualizarBotones();
378         } else {
379             alert('Error al detener transmisión');
380         }
381     })
382     .catch(() => alert('Error al detener transmisión'));
383 }
384
385 // Muestra resultados del cultivo y actualiza botones
386 function mostrarResultados() {
387     document.getElementById('resultadoCultivo').style.display = 'block'
388     ;
389     document.getElementById('btnMostrar').disabled = true;
390     document.getElementById('btnVolver').disabled = false;
391 }
```

```
392 // Redirige al dashboard principal
393 function volverAlDashboard() {
394     window.location.href = "{{_url_for('dashboard')}}";
395 }
396
397 // Actualiza estados de botones según estado streaming
398 function actualizarBotones() {
399     document.getElementById('btnPause').disabled = streamingPaused ||
        streamingStopped;
400     document.getElementById('btnPlay').disabled = !streamingPaused ||
        streamingStopped;
401     document.getElementById('btnRestart').disabled = streamingStopped;
402     // Botón snapshot eliminado, no controlado aquí
403     document.getElementById('btnStop').disabled = streamingStopped;
404     // Mostrar y Volver se activan manualmente
405 }
406
407 // Resetea el gráfico de estadísticas a estado inicial
408 function resetChart() {
409     statsChart.data.labels = [];
410     statsChart.data.datasets.forEach(dataset => dataset.data = []);
411     statsChart.update();
412 }
413
414 // Inicialización al cargar la página
415 window.onload = () => {
416     // Inicia detector en backend
417     fetch('/iniciar_detector', { method: 'POST' })
418         .then(res => res.json())
419         .then(data => {
420             if (data.status === 'ok') {
421                 // Setea fuente de video para mostrar streaming
422                 const video = document.getElementById('videoFeed');
423                 video.src = "{{_url_for('video_feed')}}";
424             }
425         });
426 }
```

```
425         // Oculta loading cuando se carga el video
426         video.onload = () => {
427             ocultarLoading();
428         };
429
430         // Timeout de fallback por si el video tarda
431         setTimeout(() => {
432             ocultarLoading();
433         }, 5000);
434
435         // Actualiza stats periódicamente
436         intervalID = setInterval(actualizarStats, 2000);
437         actualizarBotones();
438     } else {
439         // Si falla, redirige al dashboard
440         window.location.href = "{{url_for('dashboard')}}";
441     }
442 })
443 .catch(() => {
444     // En error, redirige también al dashboard
445     window.location.href = "{{url_for('dashboard')}}";
446 });
447 };
448 </script>
449 </body>
450 </html>
```

Listing 7: Fragmento del archivo clima.py

## Upload

```
34
35
36 <!DOCTYPE html>
37 <html lang="es">
```

```
38 <head>
39   <meta charset="UTF-8" />
40   <title>Cargar y Ver Resultados - Sistema de Roya</title>
41   <!-- Google Fonts -->
42   <link href="https://fonts.googleapis.com/css2?family=Montserrat:
      wght@400;700&display=swap" rel="stylesheet" />
43   <meta name="viewport" content="width=device-width, initial-scale=1.0">
44   <style>
45     body {
46       font-family: 'Montserrat', sans-serif;
47       background: linear-gradient(120deg, #f0fdf4 60%, #d2f2ea 100%);
48       padding: 0;
49       margin: 0;
50       min-height: 100vh;
51     }
52     .header-portada {
53       display: flex;
54       align-items: center;
55       justify-content: space-between;
56       gap: 28px;
57       margin-bottom: 32px;
58       margin-top: 0;
59       padding: 20px 0 14px 0;
60       background: rgba(255,255,255,0.97);
61       border-radius: 18px;
62       box-shadow: 0 4px 22px rgba(44, 62, 80, 0.09);
63     }
64     .header-logo-portada {
65       flex: 1;
66       display: flex;
67       align-items: center;
68       justify-content: center;
69     }
70     .titulo-centrado {
71       text-align: center;
```

```
72     letter-spacing: 2px;
73     font-size: 2.3rem;
74     margin-top: 18px;
75     margin-bottom: 36px;
76     font-weight: 900;
77     background: linear-gradient(90deg, #1A77C8 20%, #0e0af0c9 80%);
78     -webkit-background-clip: text;
79     -webkit-text-fill-color: transparent;
80     background-clip: text;
81     color: #18467a;
82     text-transform: uppercase;
83     font-family: 'Montserrat', Arial, sans-serif;
84     text-shadow: 0 3px 14px #41b3f540, 0 1px 0 #227c4933;
85     line-height: 1.18;
86 }
87 .header-logo-portada img {
88     width: 120px;
89     height: 120px;
90     object-fit: contain;
91     background: #f8f8f8;
92     border-radius: 15px;
93     border: 1.5px solid #e0e0e0;
94     box-shadow: 0 1px 8px rgba(0,0,0,0.06);
95 }
96 .header-titulos-portada {
97     flex: 3;
98     text-align: center;
99     display: flex;
100    flex-direction: column;
101    align-items: center;
102    gap: 8px;
103 }
104 .titulo-utn {
105     font-size: 1.9rem;
106     font-weight: 900;
```

```
107         color: #d3371b;
108         letter-spacing: 2px;
109         text-transform: uppercase;
110         margin-bottom: 0;
111         margin-top: 0;
112     }
113     .subtitulo-carrera {
114         font-size: 1.25em;
115         color: #18467a;
116         font-weight: 700;
117         letter-spacing: 1.5px;
118         text-transform: uppercase;
119         margin-bottom: 0;
120     }
121     .header-estudiante-portada {
122         font-size: 1.08rem;
123         font-weight: 500;
124         color: #363636;
125         letter-spacing: 0.3px;
126         margin-top: 7px;
127         margin-bottom: 0;
128     }
129     @media (max-width: 900px) {
130         .header-portada {
131             flex-direction: column;
132             gap: 14px;
133             padding: 14px 4px 10px 4px;
134         }
135         .header-logo-portada img {
136             width: 85px;
137             height: 85px;
138         }
139         .titulo-utn {
140             font-size: 1.15rem;
141         }
```

```
142     .subtitulo-carrera {
143         font-size: 1rem;
144     }
145 }
146 .container {
147     max-width: 1100px;
148     margin: 40px auto;
149     background: #fff;
150     padding: 32px;
151     border-radius: 24px;
152     box-shadow: 0 8px 32px rgba(44, 62, 80, 0.12);
153     position: relative;
154     animation: fadeIn 1s;
155 }
156 @keyframes fadeIn { from { opacity: 0; transform: translateY(30px);
157     } to { opacity: 1; transform: none; } }
158 h2 {
159     margin-bottom: 22px;
160     color: #186cb1;
161     font-size: 2.1rem;
162     display: flex;
163     align-items: center;
164     gap: 8px;
165 }
166 .call-to-action {
167     font-size: 1.16rem;
168     color: #185697;
169     margin-bottom: 22px;
170     margin-top: 6px;
171     background: linear-gradient(90deg, #e3f3fd 70%, #d2f2ea 100%);
172     padding: 20px 36px 18px 36px;
173     border-radius: 18px;
174     font-weight: 600;
175     border-left: 6px solid #42b6e7;
176     border-right: 6px solid #42e7c7;
```

```
176     box-shadow: 0 6px 22px #41b3f540, 0 1.5px 0 #22d09b33 inset;
177     letter-spacing: 0.7px;
178     transition: box-shadow 0.22s, transform 0.22s;
179     text-align: center;
180     position: relative;
181 }
182 .call-to-action::before {
183     content: "";
184     position: absolute;
185     left: 22px;
186     top: 18px;
187     font-size: 1.6rem;
188     color: #41b3f5b5;
189     filter: drop-shadow(0 0 3px #41b3f588);
190 }
191 @media (max-width: 600px) {
192     .call-to-action {
193         padding: 15px 10px;
194         font-size: 1.01rem;
195         border-radius: 12px;
196     }
197     .call-to-action::before {
198         left: 12px;
199         top: 13px;
200         font-size: 1.2rem;
201     }
202 }
203 .drag-drop-area {
204     border: 2.8px dashed #3896e4;
205     background: #f5fbff;
206     border-radius: 17px;
207     text-align: center;
208     padding: 40px 20px;
209     cursor: pointer;
210     margin-bottom: 22px;
```

```
211         transition: border-color 0.3s, background 0.3s;
212         position: relative;
213     }
214     .drag-drop-area.dragover {
215         border-color: #3fb77c;
216         background: #e2f5ea;
217     }
218     .drag-drop-area input[type="file"] {
219         display: none;
220     }
221     .drag-drop-text {
222         font-size: 1.3rem;
223         color: #1875c6;
224         margin-bottom: 8px;
225         font-weight: bold;
226     }
227     .drag-drop-subtext {
228         color: #7a8fa8;
229         font-size: 1.04rem;
230     }
231     .custom-file-btn {
232         background: linear-gradient(90deg,#1875c6,#41b3f5 90%);
233         color: #fff;
234         border: none;
235         padding: 11px 25px;
236         border-radius: 10px;
237         font-size: 1rem;
238         font-weight: bold;
239         cursor: pointer;
240         margin-top: 18px;
241         margin-bottom: 5px;
242         transition: background 0.2s, transform 0.2s;
243         box-shadow: 0 2px 10px #41b3f530;
244     }
245     .custom-file-btn:hover {
```

```
246     background: linear-gradient(90deg,#125291,#41b3f5 100%);
247     transform: scale(1.04);
248 }
249 .nombre-archivo {
250     color: #246eb8;
251     font-size: 0.99rem;
252     margin: 12px 0 0 0;
253     font-weight: bold;
254 }
255 .preview-multiple {
256     display: flex;
257     gap: 25px;
258     flex-wrap: wrap;
259     justify-content: center;
260     margin-top: 16px;
261 }
262 .preview-thumb {
263     position: relative;
264     border: 2px solid #86a62d44;
265     border-radius: 15px;
266     background: #f7fafb;
267     margin-bottom: 8px;
268     transition: box-shadow .23s;
269     box-shadow: 0 4px 14px #2da67413;
270     cursor: zoom-in;
271     overflow: hidden;
272 }
273 .preview-thumb img, .preview-thumb video {
274     display: block;
275     max-width: 160px;
276     max-height: 110px;
277     border-radius: 13px;
278 }
279 .preview-thumb:hover {
280     box-shadow: 0 6px 20px #1875c677;
```

```
281         z-index: 2;
282     }
283     .lightbox {
284         position: fixed;
285         top: 0; left: 0; width: 100vw; height: 100vh;
286         background: rgba(33,44,52,0.90);
287         display: flex; align-items: center; justify-content: center;
288         z-index: 9999;
289         animation: fadeIn .4s;
290     }
291     .lightbox img, .lightbox video {
292         max-width: 90vw;
293         max-height: 90vh;
294         border-radius: 17px;
295         box-shadow: 0 12px 44px #1875c655;
296     }
297     .lightbox .close-btn {
298         position: absolute;
299         top: 50px; right: 70px;
300         font-size: 2.4rem;
301         color: #fff;
302         background: rgba(24,117,198,0.22);
303         border: none;
304         border-radius: 16px;
305         padding: 5px 18px 3px 18px;
306         cursor: pointer;
307         z-index: 10001;
308     }
309     .progress-container {
310         width: 98%;
311         background-color: #e0e0e0;
312         border-radius: 15px;
313         overflow: hidden;
314         height: 21px;
315         margin: 20px auto 6px auto;
```

```
316     }
317     .progress-bar {
318         height: 100%;
319         width: 0%;
320         background: linear-gradient(90deg, #2da674, #1875c6 90%);
321         transition: width 0.3s;
322     }
323     .fila-botones {
324         display: flex;
325         gap: 22px;
326         justify-content: center;
327         margin: 22px 0 8px 0;
328         flex-wrap: wrap;
329     }
330     .boton-accion {
331         padding: 13px 32px;
332         border: none;
333         border-radius: 12px;
334         font-size: 1rem;
335         font-weight: bold;
336         color: #fff;
337         cursor: pointer;
338         box-shadow: 0 2px 10px rgba(44,62,80,0.09);
339         transition: transform 0.18s, box-shadow 0.18s;
340         display: flex;
341         align-items: center;
342         gap: 9px;
343         text-decoration: none;
344         justify-content: center;
345     }
346     .boton-accion.verde {
347         background: linear-gradient(90deg,#2d572c,#2da674 90%);
348     }
349     .boton-accion.verde:hover {
350         background: linear-gradient(90deg,#256e1c 70%,#28965a 100%);
```

```
351         transform: scale(1.07);
352     }
353     .boton-accion.rojo {
354         background: linear-gradient(90deg,#c0392b,#e74c3c 90%);
355     }
356     .boton-accion.rojo:hover {
357         background: linear-gradient(90deg,#96281b 70%,#c0392b 100%);
358         transform: scale(1.07);
359     }
360     .boton-accion.azul {
361         background: linear-gradient(90deg,#5140af,#958dcc 90%);
362     }
363     .boton-accion.azul:hover {
364         background: linear-gradient(90deg,#6f64b8 70%,#5140af 100%);
365         transform: scale(1.07);
366     }
367     .filtros {
368         display: flex;
369         gap: 10px;
370         flex-wrap: wrap;
371         margin: 28px 0 10px 0;
372     }
373     .filtros input[type="search"], .filtros select {
374         padding: 10px 12px;
375         border-radius: 9px;
376         border: 1.5px solid #2da674;
377         background: #fafdf9;
378         font-size: 15px;
379         transition: box-shadow .2s;
380     }
381     .filtros input[type="search"]:focus, .filtros select:focus {
382         box-shadow: 0 0 0 2px #2da67455;
383         outline: none;
384     }
385     .vista-btn {
```

```
386     background: #eef3ea;
387     border: none;
388     border-radius: 8px;
389     font-size: 18px;
390     cursor: pointer;
391     transition: background 0.2s;
392     padding: 10px 15px;
393 }
394 .vista-btn.active, .vista-btn:hover {
395     background: #b2e2ce;
396 }
397 .galeria {
398     display: grid;
399     grid-template-columns: repeat(auto-fill, minmax(290px, 1fr));
400     gap: 28px;
401     margin-top: 24px;
402 }
403 .galeria-item {
404     background: #f5faf7;
405     padding: 14px;
406     border-radius: 15px;
407     box-shadow: 0 4px 12px rgba(0,0,0,0.07);
408     text-align: center;
409     position: relative;
410     transition: box-shadow 0.2s;
411 }
412 .galeria-item:hover {
413     box-shadow: 0 8px 18px rgba(45,166,116,0.10);
414 }
415 .galeria img, .galeria video {
416     max-width: 100%;
417     border-radius: 12px;
418     margin-bottom: 10px;
419     cursor: pointer;
420     box-shadow: 0 2px 10px #2da67411;
```

```
421     }
422     .galeria-item p {
423         font-size: 13px;
424         word-break: break-all;
425         color: #1d2d2d;
426     }
427     .menu-btn {
428         position: absolute;
429         top: 11px;
430         right: 15px;
431         font-size: 20px;
432         background: #e8f4ec;
433         border: none;
434         border-radius: 50%;
435         width: 34px;
436         height: 34px;
437         display: flex;
438         align-items: center;
439         justify-content: center;
440         cursor: pointer;
441     }
442     .menu-btn:hover { background: #b2e2ce; }
443     .dropdown-content {
444         display: none;
445         position: absolute;
446         right: 10px;
447         top: 44px;
448         background-color: #fff;
449         box-shadow: 0 8px 24px rgba(44,62,80,0.19);
450         border-radius: 13px;
451         z-index: 3;
452         min-width: 160px;
453     }
454     .dropdown-content a, .dropdown-content form button {
455         color: #2d572c;
```

```
456     padding: 13px 20px;
457     text-decoration: none;
458     display: block;
459     background: none;
460     border: none;
461     text-align: left;
462     font-family: 'Montserrat', sans-serif;
463     font-size: 14px;
464     width: 100%;
465     cursor: pointer;
466     border-bottom: 1px solid #f0f0f0;
467 }
468 .dropdown-content a:last-child, .dropdown-content form button:last-
469     child {
470     border-bottom: none;
471 }
472 .dropdown-content a:hover, .dropdown-content form button:hover {
473     background-color: #eef6f0;
474 }
475 .fecha-label {
476     margin-top: 48px;
477     font-size: 18px;
478     font-weight: bold;
479     color: #169155;
480 }
481 .toast {
482     position: fixed;
483     bottom: 36px;
484     right: 36px;
485     background: linear-gradient(90deg,#25a064,#8e44ad 80%);
486     color: #fff;
487     padding: 18px 28px;
488     border-radius: 15px;
489     font-weight: bold;
490     box-shadow: 0 7px 18px rgba(0, 0, 0, 0.15);
```

```
490         opacity: 0;
491         pointer-events: none;
492         z-index: 999;
493         font-size: 1.08rem;
494         transition: opacity 0.5s, transform 0.5s;
495         transform: translateY(20px);
496         display: flex;
497         align-items: center;
498         gap: 12px;
499     }
500     .toast.show {
501         opacity: 1;
502         pointer-events: auto;
503         transform: translateY(0);
504     }
505     .toast button {
506         margin-left: 18px;
507         padding: 2px 13px;
508         border: none;
509         background: #fff;
510         color: #8e44ad;
511         border-radius: 7px;
512         font-weight: 600;
513         font-size: 1rem;
514         cursor: pointer;
515     }
516     @media (max-width: 600px) {
517         .container { padding: 10px; }
518         .preview-multiple { gap: 8px; }
519         .lightbox .close-btn { top: 9px; right: 10px;}
520         .fila-botones { flex-direction: column; gap: 13px; }
521     }
522 </style>
523 </head>
524 <body>
```

```

525
526 <div class="container">
527     <div class="header-portada">
528         <div class="header-logo-portada">
529             
530         </div>
531     <div class="header-titulos-portada">
532         <div class="titulo-utn">UNIVERSIDAD TÉCNICA DEL NORTE</div>
533         <div class="subtitulo-carrera">CARRERA DE INGENIERÍA EN
                    TELECOMUNICACIONES</div>
534     <div class="header-estudiante-portada">
535         <strong>Autor:</strong> <span style="color:#353434;">Farid
                    Avila</span><br>
536         <strong>Director:</strong> <span style="color:#353434;">Msc
                    . Carlos Vásquez</span><br>
537     </div>
538 </div>
539 <div class="header-logo-portada">
540     
541 </div>
542 </div>
543
544 <h2 class="titulo-centrado">
545     SISTEMA INTELIGENTE PARA LA DETECCIÓN AUTOMATIZADA DE ROYA CON IA
546 </h2>
547 <div class="call-to-action">
548     Para iniciar, por favor arrastra y suelta tus <span style="color
                    :#185697;">imágenes</span> o <span style="color:#185697;">videos
                    </span> aquí...
549 </div>
550 <form id="form-upload" action="{_url_for('cargar_archivo')}" method=
                    "POST" enctype="multipart/form-data" onsubmit="subirArchivos(event)"
                    >

```

```

551 <div style="margin-bottom: 18px; text-align: center;">
552   <label for="nombrePersonalizadoUpload" style="font-weight: bold;
553     color: #169155; font-size: 1.09rem;">
554     Nombre personalizado para el archivo:
555   </label>
556   <input type="text" id="nombrePersonalizadoUpload" name="
557     nombrePersonalizado"
558     placeholder="Ejemplo: Hoja Invernadero 1" maxlength="50"
559     style="padding: 7px 12px; border-radius: 7px; border: 1.5px
560     solid #2da674; font-size: 1rem; width: 60%; margin-top:
561     6px;">
562 </div>
563 <div id="drag-drop-area" class="drag-drop-area" onclick="document.
564   getElementById('archivo').click()"
565   ondragover="dragOverHandler(event)" ondragleave="
566   dragLeaveHandler(event)" ondrop="dropHandler(event)">
567   <div class="drag-drop-text"> Suelta tus archivos aquí</div>
568   <div class="drag-drop-subtext"> 0 haz clic para seleccionarlos
569   desde tu compu</div>
570   <button type="button" class="custom-file-btn" onclick="document
571     .getElementById('archivo').click(); event.stopPropagation();"
572   >Seleccionar archivo(s)</button>
573   <input id="archivo" name="archivo" type="file" accept="image/*,
574     video/*" multiple>
575   <div id="nombre-archivo" class="nombre-archivo"></div>
576 </div>
577 <div id="preview" class="preview-multiple"></div>
578 <div id="spinner" style="display: none;">
579   <div class="progress-container">
580     <div class="progress-bar" id="progressBar"></div>
581   </div>
582   <p id="progressText">Subiendo archivos... </p>
583 </div>
584 <div class="fila-botones">

```

```

575     <button type="submit" class="boton-accion_verde"> Subir y
        detectar</button>
576     <a href="{url_for('dashboard')}" class="boton-accion_rojo">
        Volver al Menú Principal</a>
577     <a href="{url_for('ver_historial')}" class="boton-accion_
        azul"> Historial</a>
578     </div>
579 </form>
580
581 {% if archivos_por_fecha %}
582 <div class="filtros">
583     <input type="search" placeholder="Buscar archivo..." oninput="
        filtrarArchivos(this.value)">
584     <select onchange="filtrarPorTipo(this.value)">
585         <option value=""> Tipo</option>
586         <option value="imagenes"> Imágenes</option>
587         <option value="video"> Videos</option>
588     </select>
589 </div>
590
591 <h2 style="margin-top: 24px;">Resultados Procesados:</h2>
592 {% for fecha, tipo_archivos in archivos_por_fecha.items() %}
593 <div class="fecha-label"> {{ fecha }}</div>
594 {% if tipo_archivos.imagenes %}
595 <h3> Imágenes</h3>
596 <div class="galeria">
597     {% for archivo in tipo_archivos.imagenes %}
598     <div class="galeria-item">
599         
600         <button class="menu-btn" onclick="toggleDropdown(this)" ></
        button>
601     <div class="dropdown-content">

```

```
602     <a href="{_{url_for('obtener_resultado',_filename=archivo.
        filename)}_}" download> Descargar</a>
603     <form method="POST" action="{_{url_for('eliminar_resultado
        ',_filename=archivo.filename)}_}">
604         <button type="submit" style="color:#e74c3c;_font-weight
                :bold;"> Eliminar</button>
605     </form>
606 </div>
607 {% if archivo.nombre_personalizado %}
608     <p style="color:#169155;_font-weight:bold;_font-size:0.97
        rem;">{{ archivo.nombre_personalizado }}</p>
609 {% endif %}
610     <p style="font-size:12px;">{{ archivo.filename }}</p>
611 </div>
612 {% endfor %}
613 </div>
614 {% endif %}
615 {% if tipo_archivos.videos %}
616 <h3> Videos</h3>
617 <div class="galeria">
618     {% for archivo in tipo_archivos.videos %}
619     <div class="galeria-item">
620         <video controls onclick="mostrarLightbox(this)">
621             <source src="{_{url_for('obtener_resultado',_filename=
                archivo.filename)}_}" type="video/mp4">
622             Tu navegador no soporta el video.
623         </video>
624         <button class="menu-btn" onclick="toggleDropdown(this)" ></
                button>
625         <div class="dropdown-content">
626             <a href="{_{url_for('obtener_resultado',_filename=archivo.
                filename)}_}" download> Descargar</a>
627             <form method="POST" action="{_{url_for('eliminar_resultado
                ',_filename=archivo.filename)}_}">
```

```
628         <button type="submit" style="color:#e74c3c; font-weight
           :bold;"> Eliminar</button>
629     </form>
630 </div>
631 {% if archivo.nombre_personalizado %}
632     <p style="color:#169155; font-weight:bold; font-size:0.97
           rem;">{{ archivo.nombre_personalizado }}</p>
633     {% endif %}
634     <p style="font-size:12px;">{{ archivo.filename }}</p>
635 </div>
636 {% endfor %}
637 </div>
638 {% endif %}
639 {% endfor %}
640 {% endif %}
641 </div>
642
643 <div id="toast" class="toast"> Archivo procesado correctamente
644     <button onclick="cerrarToast()">OK</button>
645 </div>
646
647 <!-- Lightbox Modal -->
648 <div id="lightbox" class="lightbox" style="display:none;" onclick="
           cerrarLightbox(event)">
649     <button class="close-btn" onclick="cerrarLightbox(event)" ></button>
650     <div id="lightbox-content"></div>
651 </div>
652
653 <script>
654     const inputArchivos = document.getElementById('archivo');
655     const dragDropArea = document.getElementById('drag-drop-area');
656     const nombreArchivo = document.getElementById('nombre-archivo');
657     const preview = document.getElementById('preview');
658
659     // Manejo drag & drop
```

```
660 function dragOverHandler(e) {
661     e.preventDefault();
662     dragDropArea.classList.add('dragover');
663 }
664 function dragLeaveHandler(e) {
665     e.preventDefault();
666     dragDropArea.classList.remove('dragover');
667 }
668 function dropHandler(e) {
669     e.preventDefault();
670     dragDropArea.classList.remove('dragover');
671     const files = e.dataTransfer.files;
672     inputArchivos.files = files;
673     mostrarPreviews();
674 }
675 inputArchivos.addEventListener('change', mostrarPreviews);
676
677 // Mostrar previews de imágenes/videos seleccionados
678 function mostrarPreviews() {
679     let archivos = inputArchivos.files;
680     preview.innerHTML = "";
681     nombreArchivo.textContent = archivos.length > 1
682         ? `${archivos.length} archivos seleccionados`
683         : archivos.length === 1
684             ? archivos[0].name
685             : "";
686     for (let archivo of archivos) {
687         const tipo = archivo.type;
688         const url = URL.createObjectURL(archivo);
689         let thumb = document.createElement('div');
690         thumb.className = 'preview-thumb';
691         if (tipo.startsWith('image/')) {
692             const img = document.createElement('img');
693             img.src = url;
694             img.onclick = () => mostrarLightboxPreview(url, false);
```

```
695         thumb.appendChild(img);
696     } else if (tipo.startsWith('video/')) {
697         const vid = document.createElement('video');
698         vid.src = url;
699         vid.controls = true;
700         vid.onclick = () => mostrarLightboxPreview(url, true);
701         thumb.appendChild(vid);
702     }
703     preview.appendChild(thumb);
704 }
705 }
706
707 // Mostrar preview grande en lightbox
708 function mostrarLightboxPreview(src, isVideo) {
709     const lightbox = document.getElementById('lightbox');
710     const content = document.getElementById('lightbox-content');
711     content.innerHTML = '';
712     if (isVideo) {
713         const vid = document.createElement('video');
714         vid.src = src;
715         vid.controls = true;
716         vid.autoplay = true;
717         content.appendChild(vid);
718     } else {
719         const img = document.createElement('img');
720         img.src = src;
721         content.appendChild(img);
722     }
723     lightbox.style.display = 'flex';
724 }
725
726 // Cerrar lightbox modal
727 function cerrarLightbox(e) {
728     if (!e || e.target === document.getElementById('lightbox') || e.
        target.classList.contains('close-btn')) {
```

```
729     document.getElementById('lightbox').style.display = 'none';
730   }
731 }
732
733 // Subir archivos via AJAX con barra de progreso
734 function subirArchivos(e) {
735     e.preventDefault();
736     if (!inputArchivos.files.length) return;
737     document.getElementById('spinner').style.display = "block";
738     document.getElementById('progressBar').style.width = "0%";
739     document.getElementById('progressText').textContent = "Subiendo_
       archivos...";
740     const formData = new FormData();
741     for (const archivo of inputArchivos.files) { formData.append('
       archivo', archivo); }
742     const nombrePersonalizado = document.getElementById('
       nombrePersonalizadoUpload').value;
743     formData.append('nombrePersonalizado', nombrePersonalizado);
744     const xhr = new XMLHttpRequest();
745     xhr.open("POST", "{{_url_for('cargar_archivo')_}}", true);
746     xhr.upload.onprogress = function(e) {
747         if (e.lengthComputable) {
748             let porc = Math.round((e.loaded / e.total) * 100);
749             document.getElementById('progressBar').style.width = porc +
               "%";
750             document.getElementById('progressText').textContent = `
               Subiendo archivos... ${porc}%`;
751         }
752     };
753     xhr.onload = function() {
754         if (xhr.status === 200) {
755             document.getElementById('progressBar').style.width = "100%"
               ;
756             document.getElementById('progressText').textContent = "
               Proceso de detección de Roya completado!";
```

```
757         setTimeout(() => { location.reload(); }, 1500);
758     } else {
759         document.getElementById('progressText').textContent = "
760             Error al subir archivos.";
761     }
762 };
763 xhr.onerror = function() { document.getElementById('progressText').
764     textContent = "Error de red al subir."; };
765 xhr.send(formData);
766 }
767
768 // Control dropdown menú en cada archivo
769 function toggleDropdown(btn) {
770     const menu = btn.nextElementSibling;
771     menu.style.display = (menu.style.display === "block") ? "none" : "
772         block";
773     document.querySelectorAll('.dropdown-content').forEach(el => {
774         if (el !== menu) el.style.display = 'none';
775     });
776     event.stopPropagation();
777 }
778
779 window.onclick = function(event) {
780     if (!event.target.matches('.menu-btn')) {
781         document.querySelectorAll('.dropdown-content').forEach(el => {
782             el.style.display = 'none'; });
783     }
784 }
785
786 // Mostrar toast de confirmación
787 function mostrarToast() {
788     const toast = document.getElementById("toast");
789     toast.classList.add("show");
790 }
791
792 // Cerrar toast y limpiar preview + reset botón subir
```

```
788 function cerrarToast() {
789     const toast = document.getElementById("toast");
790     toast.classList.remove("show");
791     document.querySelector(".boton-accion.verde").disabled = false;
792     document.getElementById("spinner").style.display = "none";
793     setTimeout(()=> {
794         document.getElementById("preview").innerHTML = "";
795         window.scrollTo({ top: document.querySelector(".galeria")?.
796             offsetTop || 0, behavior: "smooth" });
797     }, 200);
798 }
799 // Alternar vista lista/cuadrícula (función placeholder, puedes
800     implementar)
801 let vistaLista = false;
802 function alternarVista(btn) {
803     vistaLista = !vistaLista;
804     document.body.classList.toggle('vista-lista', vistaLista);
805     if (btn) btn.classList.toggle('active', vistaLista);
806 }
807 // Filtro de archivos por nombre
808 function filtrarArchivos(valor) {
809     const items = document.querySelectorAll('.galeria-item');
810     items.forEach(item => {
811         const nombre = item.querySelector('p').textContent.toLowerCase
812             ();
813         item.style.display = nombre.includes(valor.toLowerCase()) ? '
814             block' : 'none';
815     });
816 }
817 // Filtro de archivos por tipo (imagen/video)
818 function filtrarPorTipo(tipo) {
819     const items = document.querySelectorAll('.galeria-item');
```

```
819     items.forEach(item => {
820         const esImagen = item.querySelector('img') !== null;
821         const esVideo = item.querySelector('video') !== null;
822         if (!tipo || (tipo === 'imagen' && esImagen) || (tipo === '
            video' && esVideo)) {
823             item.style.display = 'block';
824         } else {
825             item.style.display = 'none';
826         }
827     });
828 }
829
830 // Mostrar imagen o video grande en lightbox
831 function mostrarLightbox(elemento) {
832     const src = elemento.tagName === 'IMG'
833         ? elemento.src
834         : elemento.querySelector('source')?.src || elemento.currentSrc;
835     const isVideo = elemento.tagName === 'VIDEO';
836     const lightbox = document.getElementById('lightbox');
837     const content = document.getElementById('lightbox-content');
838     content.innerHTML = '';
839     if (isVideo) {
840         const vid = document.createElement('video');
841         vid.src = src;
842         vid.controls = true;
843         vid.autoplay = true;
844         content.appendChild(vid);
845     } else {
846         const img = document.createElement('img');
847         img.src = src;
848         content.appendChild(img);
849     }
850     lightbox.style.display = 'flex';
851 }
852 </script>
```

```
853 </body>  
854 </html>
```

Listing 8: Fragmento del archivo clima.py