



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

**TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN TELECOMUNICACIONES**

**TEMA: “PROTOTIPO ELECTRÓNICO BASADO EN VISIÓN ARTIFICIAL
PARA LA DETECCIÓN TEMPRANA DE LA ENFERMEDAD BOTRYTIS CINEREA
EN HOJAS DE GRANADILLA(PASSIFLORA LIGULARIS)”**



AUTOR: Ermel Aldair Robles Robles
DIRECTOR: MSc. Fabián Geovanny Cuzme Rodríguez
Ibarra-Ecuador
2026

UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. 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:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1003844881		
APELLIDOS Y NOMBRES:	ROBLES ROBLES ERMEL ALDAIR		
DIRECCIÓN:	Dr. Plutarco Larrea y Cristobal Tobar Subía		
EMAIL:	earoblesr@utn.edu.ec		
TELÉFONO FIJO:	XXXXXXXXXXXX	TELÉFONO MÓVIL:	0999670730
DATOS DE LA OBRA			
TÍTULO:	PROTOTIPO ELECTRÓNICO BASADO EN VISIÓN ARTIFICIAL PARA LA DETECCIÓN TEMPRANA DE LA ENFERMEDAD BOTRYTIS CINEREA EN HOJAS DE GRANADILLA(PASSIFLORA LIGULARIS)		
AUTOR (ES):	ROBLES ROBLES ERMEL ALDAIR		
FECHA: DD/MM/AAAA	25/02/2026		
SOLO PARA TRABAJOS DE GRADO			
PROGRAMA:	PREGRADO ☒ POSGRADO ☐		
TITULO POR EL QUE OPTA:	INGENIERO EN TELECOMUNICACIONES		
ASESOR /DIRECTOR:	MSc. Fabián Geovanny Cuzme Rodríguez MSc. Luis Edilberto Suárez Zambrano		

2. CONSTANCIAS

El autor (es) manifiesta (n) 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 (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 26 días del mes de febrero de 2026.

EL AUTOR:

Ermel Aldair Robles Robles

C.C: 1003844881

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 26 de febrero de 2026

MAGÍSTER FABIÁN GEOVANNY CUZME RODRÍGUEZ
DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN

CERTIFICA:

Que el presente trabajo de Titulación “PROTOTIPO ELECTRÓNICO BASADO EN VISIÓN ARTIFICIAL PARA LA DETECCIÓN TEMPRANA DE LA ENFERMEDAD BOTRYTIS CINEREA EN HOJAS DE GRANADILLA(PASSIFLORA LIGULARIS)”, ha sido desarrollado por el señor Robles Robles Ermel Aldair bajo mi supervisión.

Es todo cuanto puedo certificar en honor a la verdad.

MSc. Fabián Geovanny Cuzme Rodríguez

C.C.: 1311527012

DIRECTOR

DEDICATORIA

Dedico este logro, en primer lugar, a mis queridos padres, quienes con su amor incondicional, esfuerzo incansable y sacrificio constante han sido el pilar fundamental de mi vida. Gracias por creer en mí incluso cuando yo dudaba, por sostenerme en los momentos difíciles y por acompañarme en cada paso de esta etapa universitaria que, aunque larga y desafiante, hoy culmina gracias a ustedes. Sin su apoyo, comprensión y ejemplo de perseverancia, este sueño no habría sido posible.

A mi hermana, por ser mi compañera de vida, por sus palabras oportunas, su motivación sincera y su guía constante. Gracias por impulsarme a seguir adelante cuando el camino parecía cuesta arriba y por recordarme siempre quién soy y hacia dónde voy.

A mis sobrinos, quienes con su ternura, alegría y abrazos sinceros iluminan mis días. Ustedes son una fuente infinita de inspiración y la razón por la que deseo superarme cada día más, para ser un ejemplo del que puedan sentirse orgullosos.

Y a mis amigos, gracias por su apoyo incondicional, por las palabras de ánimo en los momentos de estrés, por las risas que aligeraron la carga y por cada experiencia compartida que convirtió este proceso en un recuerdo inolvidable.

KECV <3

Este logro no es solo mío, es de todos ustedes, porque en cada página escrita, en cada esfuerzo y en cada meta alcanzada, siempre estuvieron presentes en mi corazón.

Robles Robles Ermel Aldair

AGRADECIMIENTO

Expreso mi profundo agradecimiento al Msc. Fabián Cuzme, por su valiosa guía académica, paciencia y constante apoyo durante el desarrollo de esta investigación. Sus conocimientos y orientaciones fueron fundamentales para la culminación exitosa de este trabajo.

De igual manera, agradezco al ingeniero Luis Suárez, por su acompañamiento técnico, sus recomendaciones oportunas y su disposición para aportar con su experiencia en cada etapa del proyecto.

Su compromiso, profesionalismo y dedicación dejaron una huella importante en mi formación académica y profesional.

Robles Robles Ermel Aldair

RESUMEN EJECUTIVO

La presente investigación tuvo como objetivo diseñar e implementar un sistema electrónico basado en visión artificial para la detección temprana de la enfermedad *Botrytis cinerea* en hojas de granadilla (*Passiflora ligularis*), con el fin de mejorar la productividad del cultivo y reducir pérdidas económicas asociadas a la propagación de la enfermedad. El sistema desarrollado integra nodos de monitoreo compuestos por una Raspberry Pi, cámaras digitales, sensores ambientales y un módulo de alimentación autónomo mediante energía solar, permitiendo la captura y procesamiento de imágenes en tiempo real. Para la identificación de estados fitosanitarios de las hojas se implementó un modelo de aprendizaje profundo entrenado con un conjunto de datos etiquetado, logrando una clasificación eficiente entre hojas sanas y hojas afectadas.

La validación del sistema se realizó en condiciones reales de campo en tres bloques del cultivo con diferentes condiciones de manejo agronómico. Los resultados obtenidos evidenciaron una efectividad promedio del 88,40 % en la detección de hojas, con una concordancia significativa entre el sistema y la evaluación realizada por el agricultor. Asimismo, el análisis económico demostró que la inversión real requerida para la implementación del sistema a escala total del cultivo asciende a 2 251,96 USD, con un retorno de la inversión superior al 400 % anual, evidenciando su viabilidad técnica y económica. Finalmente, se concluye que el sistema propuesto constituye una herramienta tecnológica efectiva para la detección temprana de *Botrytis cinerea*, contribuyendo a la toma de decisiones oportunas en el manejo fitosanitario del cultivo y al fortalecimiento de la agricultura de precisión en pequeños y medianos productores.

Palabras Clave: Visión artificial, *Botrytis cinerea*, Agricultura de precisión, Procesamiento digital de imágenes, Detección temprana de enfermedades, Redes neuronales convolucionales (CNN), Inteligencia artificial, Clasificación de imágenes, Prototipo electrónico.

ABSTRACT

This research aimed to design and implement an electronic system based on artificial vision for the early detection of *Botrytis cinerea* disease in passion fruit leaves (*Passiflora ligularis*), to improve crop productivity and reduce economic losses associated with disease spread. The developed system integrates monitoring nodes composed of a Raspberry Pi, digital cameras, environmental sensors, and an autonomous solar-powered supply module, enabling real-time image acquisition and processing. A deep learning model was implemented to classify leaf health status, distinguishing between healthy and infected leaves using a labeled dataset.

System validation was conducted under real field conditions in three crop blocks with different agronomic management conditions. The results showed an average detection effectiveness of 88.40 %, with an important level of agreement between the automated system and the farmer's manual evaluation. Additionally, the economic analysis demonstrated that the real investment required for full-scale implementation is approximately USD 2,251.96, achieving a return on investment greater than 400 % annually, confirming the technical and economic feasibility of the proposed solution. The proposed system represents an effective technological tool for the early detection of *Botrytis cinerea*, supporting timely decision-making in crop health management and promoting the adoption of precision agriculture practices among small and medium-sized producers.

Keywords: Artificial vision, *Botrytis cinerea*, Precision agriculture, Digital image processing, Early disease detection, Convolutional neural networks (CNN), Artificial intelligence, Image classification, electronic prototype.

LISTA DE CONTENIDOS

1	CAPÍTULO I: Antecedentes	20
1.1	Tema	20
1.2	Problema.....	20
1.3	Objetivos	21
1.3.1	Objetivo General	21
1.3.2	Objetivos Específicos	21
1.4	Alcance.....	22
1.5	Justificación.....	24
2	CAPÍTULO II: FUNDAMENTACION TEORICA.....	29
2.1	Granadilla	29
2.1.1	Origen y Distribución Geográfica de la Granadilla.....	29
2.1.2	Características Físicas y Técnicas de la Granadilla.....	30
2.1.3	Cultivo y Sistema de Sostenimiento de Granadilla	31
2.1.4	Condiciones Agro-climatológicas para el Cultivo de Granadilla	31
2.1.5	Sistema de Cosecha de la Granadilla	32
2.1.6	Clasificación y Selección de la Granadilla.....	32
2.1.7	Comercialización de la Granadilla	34
2.2	Botrytis	34
2.2.1	Clasificación Taxonómica	34
2.2.2	Signos y Síntomas	35
2.2.3	Hospederos de B. cinérea	35

2.2.4	Manejo de la enfermedad	35
2.2.5	Principales problemas fitosanitarios.....	35
2.2.6	Control.....	36
2.3	Inteligencia Artificial.....	37
2.3.1	Reconocimiento Automático De Objetos	38
2.3.2	Etapas de Detección de Objetos de una Imagen.....	39
2.3.3	Aprendizaje Supervisado.....	40
2.3.4	Redes Neuronales Convolucionales (CNN).....	41
2.3.5	Función de Activación.....	42
2.3.6	Capa de Pooling.....	42
2.3.7	Capa Directamente Conectada	43
2.4	Aumento de datos en visión artificial.....	43
2.5	Prototipado electrónico.....	44
2.5.1	Componentes de Hardware	44
2.5.2	Componentes de Software.....	46
2.5.3	Almacenamiento de Datos.....	48
2.5.4	Google colab.....	49
3	CAPÍTULO III: Diseño del Sistema	50
3.1	Metodología	50
3.2	Análisis.....	51
3.2.1	Situación Actual	51

3.1.1	Técnicas de Recolección de Información.....	53
3.3	Requerimientos del sistema	57
3.3.1.	Nomenclatura de Requerimientos.....	57
3.3.2.	Requerimientos de Stakeholders.....	59
3.3.3.	Requerimientos del Sistema.....	60
3.3.4.	Requerimientos de Arquitectura	63
3.3.5.	Selección de Hardware	65
3.3.6.	Selección de Software.....	69
3.4	Diseño del sistema	77
3.4.1.	Diagrama General de Bloques del Sistema.....	79
3.4.2	Diagrama de Flujo.....	79
3.4.3	Descripción de escenarios.....	86
3.5	Diseño del hardware	88
3.5.1	Modelado del Adaptador de Voltaje en Proteus y en 3D.....	89
3.5.2	Diseño de la Estructura Externa	92
3.6	Desarrollo del software	96
3.6.1	Preprocesamiento de Imágenes.....	97
3.6.2	Entrenamiento con el Dataset Obtenido	98
3.6.3	Aumento de Datos Aplicando Transformaciones Geométricas Para Hojas Buenas.....	108
3.6.4	Aumento de Datos Aplicando Transformaciones Geométricas Para Hojas Malas.....	111

3.6.5	Entrenamiento de la Red Neuronal con el Robustecimiento del Dataset .	118
3.6.7	Diseño del Backend.....	132
3.7	Integración del Sistema	139
3.7.1	Foto Aérea de los Tres Bloques a Analizar.....	147
4	CAPÍTULO IV: Pruebas de Funcionamiento.....	149
4.1	Cronograma de Pruebas de Funcionamiento del Sistema	149
4.2	Pruebas funcionales	150
4.2.2.	Prueba de clasificación de bloques	150
4.2.3	Conteo Manual Técnico Realizado Para el Estudio.....	154
4.2.4	Comparación de los Bloques Estudiados.....	157
4.3	Detección Temprana de Hojas Malas en los bloques	158
4.4	Detección Visual de Hojas Malas en los Bloques realizado por el Agricultor	164
4.5	Análisis Comparativo Entre Bloques de la Parcela.....	165
4.6	Análisis de efectividad del sistema de visión artificial en condiciones reales de campo	167
4.7	Análisis Costo/Beneficio.....	169
4.7.1	Beneficios	173
4.7.2	Evaluación de la rentabilidad e impacto del sistema de detección temprana en el cultivo de granadilla	174
4.8	Comparación Cuantitativa Antes y Después de la Implementación del Sistema	178

4.8.1	Análisis de rentabilidad y retorno de la inversión aplicado a la totalidad del cultivo	178
4.8.2	Cálculo Del ROI Anual de Toda la Finca	182
4.9	Análisis Comparativo del Sistema Fijo de Visión Artificial y el Uso de Drones para la Detección Temprana de Botrytis cinerea	182
	CONCLUSIONES	186
	RECOMENDACIONES	188
4	BIBLIOGRAFÍA.....	190
5	ANEXOS.....	195
	ANEXO A: Formato de entrevista.....	195

LISTA DE TABLAS

Tabla 1 <i>Ficha técnica de la granadilla</i>	30
Tabla 2 <i>Exigencias de cultivo</i>	31
Tabla 3 <i>Clasificación de la granadilla</i>	33
Tabla 4 <i>Lista de participantes en el proyecto</i>	58
Tabla 5 <i>Abreviatura de los requerimientos</i>	58
Tabla 6 <i>Requerimientos de stakeholders</i>	59
Tabla 7 <i>Requerimientos del sistema</i>	61
Tabla 8 <i>Requerimientos de arquitectura</i>	63
Tabla 9 <i>Selección de placa embebida</i>	66
Tabla 10 <i>Selección de cámaras</i>	67
Tabla 11 <i>Selección de framework para procesamiento de imágenes</i>	69
Tabla 12 <i>Selección de lenguaje de programación</i>	70
Tabla 13 <i>Selección de tecnología inalámbrica</i>	75
Tabla 14 <i>Librerías usadas</i>	83
Tabla 15 <i>Comparativa de Épocas</i>	125
Tabla 16 <i>Conteo de Hojas que Serán Tomadas en Cuenta por Bloques</i>	158
Tabla 17 <i>Comparación de los Bloques Tanto para Hojas Detectadas por el Sistema y Conteo Manual</i>	166
Tabla 18 <i>Efectividad del Sistema De Visión Artificial por Bloque</i>	168
Tabla 19 <i>Costos de hardware</i>	169
Tabla 20 <i>Costos de software</i>	170
Tabla 21 <i>Costos de infraestructura</i>	170
Tabla 22 <i>Costos de ingeniería</i>	172
Tabla 23 <i>Costo general del sistema</i>	172

Tabla 24 <i>Antes y Después de la Implementación del Sistema</i>	178
Tabla 25 <i>Proyección económica del sistema: 10 % del sembradío vs. 100 % de la finca</i>	179
Tabla 26 <i>Costos de Hardware</i>	180
Tabla 27 <i>Comparación de soluciones tecnológicas para monitoreo de cultivo: sistema fijo vs. drones</i>	183

LISTA DE FIGURAS

Figura 1 <i>Arquitectura genérica del prototipo electrónico</i>	23
Figura 2 <i>Ciclo de Botrytis</i>	36
Figura 3 <i>Sistema de visión artificial</i>	37
Figura 4 <i>Detección de elementos</i>	39
Figura 5 <i>Clasificación lineal</i>	40
Figura 6 <i>Regresión Lineal</i>	41
Figura 7 <i>Estructura de una red convolucional</i>	41
Figura 8 <i>Ejemplo de funcionamiento de la capa de pool</i>	42
Figura 9 <i>Dataset almacenamiento de datos</i>	48
Figura 10 <i>Arquitectura de Google Colab</i>	49
Figura 11 <i>Fases de la metodología</i>	50
Figura 12 <i>Ubicación geográfica de la finca</i>	53
Figura 13 <i>Placa Seleccionada Raspberry Pi4</i>	66
Figura 14 <i>Cámara Elegida Wi-Fi Smart Camera</i>	68
Figura 15 <i>Framework Seleccionado Ultralytics YOLO</i>	70
Figura 16 <i>Lenguaje de Programación Seleccionado Python</i>	71
Figura 17 <i>Entorno de Desarrollo Seleccionado Google Colab</i>	72
Figura 18 <i>Plataforma Nube Seleccionada Firebase</i>	73

Figura 19 <i>Servicio en la Nube Seleccionado Google Cloud</i>	74
Figura 20 <i>Transmisión Inalámbrica Seleccionada WiFi</i>	75
Figura 21 <i>Medio de Acceso Seleccionado Plataforma Web</i>	76
Figura 22 <i>Diagrama de bloques general</i>	79
Figura 23 <i>Flujograma de preprocesamiento de imágenes</i>	81
Figura 24 <i>Diagrama de flujo del entrenamiento de la red neuronal</i>	82
Figura 25 <i>Flujograma de integración del sistema</i>	85
Figura 26 <i>División de la parcela</i>	87
Figura 27 <i>Escenarios de prueba</i>	88
Figura 28 <i>Modelado de control esquemático de voltajes.</i>	90
Figura 29 <i>Diseño de circuito PCB Layout</i>	90
Figura 30 <i>Adaptador de voltaje</i>	91
Figura 31 <i>Case del Prototipo Diseñado</i>	93
Figura 32 <i>Case del Prototipo Diseñado Impreso</i>	94
Figura 33 <i>Case con Panel Solar</i>	95
Figura 34 <i>Cámara Incorporada al Case</i>	96
Figura 35 <i>Captura de imagen</i>	98
Figura 36 <i>Dataset en la nube</i>	99
Figura 37 <i>Montaje de Google Drive en el entorno de Ejecución Colab</i>	99
Figura 38 <i>Permisos de uso Google Drive</i>	100
Figura 39 <i>Creación de directorios</i>	100
Figura 40 <i>División de datos en conjuntos de entrenamiento y validación</i>	102
Figura 41 <i>Carga del modelo YOLOv8</i>	103
Figura 42 <i>Detección de los objetos con el modelo YOLOv8</i>	104
Figura 43 <i>Resultados de Evaluación de Desempeño del entrenamiento</i>	105

Figura 44 <i>Matriz de Confusión</i>	106
Figura 45 <i>Curva de Confidencialidad</i>	108
Figura 46 <i>Transformación Geométrica de las Imágenes</i>	109
Figura 47 <i>Imagen Original Junto con sus Transformaciones</i>	110
Figura 48 <i>Aumento de Datos mediante Rotaciones y Almacenamiento de Imágenes</i>	111
Figura 49 <i>Aumento de Datos Mediante Transformaciones Geométricas en Imágenes Defectuosas</i>	112
Figura 50 <i>Función de Rotaciones</i>	112
Figura 51 <i>Función de Espejo</i>	113
Figura 52 <i>Función de Brillo y Espejo</i>	114
Figura 53 <i>Desplazamiento Aleatorio de Imágenes Mediante Transformación Afín</i> ..	115
Figura 54 <i>Zoom Aleatorio a las Imágenes</i>	116
Figura 55 <i>Código de Visualización de Transformada Aplicadas a las Imágenes</i>	117
Figura 56 <i>Salida de las Imágenes</i>	118
Figura 57 <i>Entrenamiento con 25 épocas</i>	119
Figura 58 <i>Resultados de Evaluación de Desempeño</i>	120
Figura 59 <i>Entrenamiento con 50 Épocas</i>	121
Figura 60 <i>Resultados de Evaluación de Desempeño</i>	121
Figura 61 <i>Entrenamiento con 75 Épocas</i>	122
Figura 62 <i>Resultados de Evaluación de Desempeño</i>	123
Figura 63 <i>Entrenamiento con 100 Épocas</i>	123
Figura 64 <i>Resultados de Evaluación de Desempeño</i>	124
Figura 65 <i>Comparativa General de Métricas Para la Selección del Número Optimo de Épocas</i>	126
Figura 66 <i>Gráfica de Exactitud</i>	128

Figura 67	<i>Gráfica de Pérdida</i>	129
Figura 68	<i>Matriz de Confusión del Modelo Entrenado</i>	131
Figura 69	<i>Valores Finales de Precisión, Recall y Fitness del Modelo</i>	132
Figura 70	<i>Captura y Transmisión de Video Mediante WebRTC y Ffmpeg</i>	133
Figura 71	<i>Transformación de Cada Frame para ser Enviado por WebRTC</i>	134
Figura 72	<i>Conexión del Cliente AsyncRTSPClient</i>	135
Figura 73	<i>Carga de Dependencias, Configuración y Variables Globales</i>	136
Figura 74	<i>Clase De Track De Video Procesado</i>	137
Figura 75	<i>Procesamiento de IA sobre los Frames Recibidos</i>	138
Figura 76	<i>Inicialización del Servidor Web y Ciclo Principal</i>	139
Figura 77	<i>Instancia de Máquina Virtual del Servidor Web</i>	140
Figura 78	<i>Interfaz Gráfica</i>	141
Figura 79	<i>Interfaz Visualización de Detecciones</i>	142
Figura 80	<i>Datos de Control de la Batería</i>	143
Figura 81	<i>Ingreso al Control</i>	144
Figura 82	<i>Configuración de Control</i>	145
Figura 83	<i>Nodo Integrado en la Parcela</i>	146
Figura 84	<i>División del Área Experimental</i>	147
Figura 85	<i>Foto Aérea de los 3 Bloques</i>	148
Figura 86	<i>Prueba de clasificación individual – Bad Leaf en un Umbral de 1</i>	150
Figura 87	<i>Bloque 1</i>	151
Figura 88	<i>Bloque 2</i>	152
Figura 89	<i>Bloque 3</i>	153
Figura 90	<i>Conteo de hojas en el Bloque 1</i>	155
Figura 91	<i>Conteo de hojas en el Bloque 2</i>	156

Figura 92	<i>Conteo de hojas en el Bloque 3</i>	157
Figura 93	<i>Detección Temprana Vista en el Bloque 1</i>	159
Figura 94	<i>Enfermedad Detectada en el Bloque 1</i>	160
Figura 95	<i>Detección Temprana Vista en el Bloque 2</i>	161
Figura 96	<i>Enfermedad Detectada en el Bloque 2</i>	162
Figura 97	<i>Detección Temprana Vista en el Bloque 3</i>	163
Figura 98	<i>Enfermedad Detectada en el Bloque 3</i>	164

CAPÍTULO I: Antecedentes

1.1 Tema

Prototipo electrónico basado en visión artificial para la detección temprana de la enfermedad *Botrytis cinerea* en hojas de granadilla(*Passiflora ligularis*)

1.2 Problema

El cultivo de la granadilla, al igual que otros cultivos de frutas, se enfrenta a la amenaza de varios hongos patógenos que residen en el suelo y pueden causar daños tanto en las raíces como en las hojas y los frutos (Mahmud, Zaman, Esau, Price, & Prithiviraj, 2019) razón por la cual la granadilla es un cultivo altamente demandante en términos de productos como insecticidas, fungicidas y plaguicidas debido a la necesidad de garantizar una producción estable. Esto se debe a la vulnerabilidad de la planta ante diversos patógenos y plagas que pueden afectar su crecimiento y rendimiento. La aplicación regular de estos productos es crucial para controlar la proliferación de insectos y hongos que podrían dañar la salud de la planta y comprometer la calidad y cantidad de la cosecha obtenida. (Molina et al., 2006)

El hongo *Botrytis* se caracteriza por su rápido crecimiento y la producción abundante de micelio gris y esporas sobre las lesiones, lo que le da su nombre a la enfermedad. Este patógeno se desarrolla con mayor facilidad en ambientes de alta humedad relativa, y el viento ayuda a la dispersión de las esporas. Puede permanecer en el suelo en forma de esclerocios, unas estructuras de resistencia, o en el tejido vegetal en descomposición a través del micelio. Como se puede apreciar en la figura 1, el árbol de causas-efectos sobre la detección tardía de la enfermedad *Botrytis cinerea* en hojas de granadilla, los síntomas de la enfermedad comienzan como pequeñas manchas pálidas o grisáceas en hojas y sépalos de la flor, o como lesiones oscuras, hundidas y con bordes definidos en los tallos. En las flores y frutos jóvenes,

provoca pudrición y cubre las estructuras con moho gris. El exceso de sombra puede favorecer el desarrollo de esta enfermedad.(Alarcon, et al. 2011)

En general, las condiciones ambientales favorables para el desarrollo de la enfermedad incluyen alta humedad y temperaturas entre 15 y 20°C (Agüero et al., 2018). Las prácticas agrícolas inadecuadas, como el exceso de riego o el espaciamiento insuficiente entre las plantas, también pueden contribuir a la proliferación del hongo (Toaquiza-Benavides et al., 2021).

Las pérdidas por *Botrytis cinerea* en la granadilla pueden ser considerables, oscilando entre el 10% y el 50% de la cosecha(Huila, 2022). Estudios realizados en Ecuador han reportado porcentajes de afectación que varían entre el 15% y el 30% (Blas, 2017).

En vista de esto, surge la necesidad de desarrollar un sistema que permita recopilar datos sobre las anomalías en las hojas de granadilla y que, al mismo tiempo, facilite la creación de un modelo predictivo capaz de identificar de manera temprana la presencia de enfermedades. Esto podría contribuir a reducir el riesgo de daños en los cultivos y mejorar la eficiencia en la gestión de la producción de granadillas.

1.3 Objetivos

1.3.1 Objetivo General

- Diseñar un prototipo electrónico basado en visión artificial para la detección temprana de la enfermedad *Botrytis cinerea* en hojas de granadilla, para mejorar la gestión agrícola promoviendo prácticas sostenibles.

1.3.2 Objetivos Específicos

- Recopilar información bibliográfica relevante sobre la detección de la enfermedad *Botrytis cinerea* en hojas de granadilla utilizando visión artificial, incluyendo estudios sobre patrones de enfermedades foliares, métodos de diagnóstico tradicionales.

- Definir los requisitos para el diseño del prototipo electrónico utilizando una metodología bottom-up.
- Implementar un prototipo de detección de la enfermedad Botrytis cinerea que aplique técnicas de visión artificial en un ambiente de pruebas controlado.
- Realizar pruebas de funcionamiento del prototipo desarrollado para la detección de enfermedades en hojas de granadilla.

1.4 Alcance

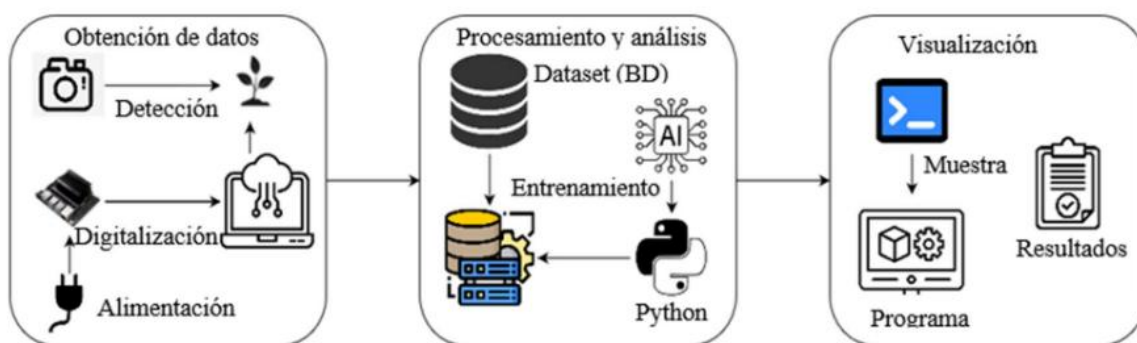
El proyecto se enfoca en el diseño, desarrollo e implementación de un prototipo electrónico que permita detectar la presencia de la enfermedad Botrytis cinerea en hojas de granadilla mediante visión artificial, con el objetivo de mejorar la gestión agrícola promoviendo prácticas sostenibles. Se seguirá la metodología Bottom-up, que facilita la unión de los diferentes componentes del sistema desde las especificaciones del cliente. Las fases que se desarrollarán son: investigación de requisitos, ciclo de verificación, integración de bloques, verificación del sistema y producto final.

En la primera fase de especificaciones, la investigación se llevará a cabo mediante un enfoque de investigación aplicada, utilizando teorías y principios de la fitopatología, biología de los hongos, patrones de enfermedades foliares, métodos de diagnóstico tradicionales y desarrollos en inteligencia artificial aplicada a la agricultura. Se examinarán artículos de investigación recientes sobre la detección temprana de enfermedades en plantas para comprender mejor el problema y las soluciones existentes. Se aplicarán métodos de recopilación de información bibliográfica, análisis de datos y diseño de sistemas basados en visión artificial y redes neuronales. En esta fase, se definirán detalladamente los requisitos y se identificarán los componentes necesarios para el diseño del prototipo electrónico.

En la Figura 1 se puede visualizar un esquema que representa la arquitectura genérica propuesta para la adquisición, procesamiento y visualización de datos del prototipo electrónico, en esta fase de ciclo de verificación y simulación, en relación con lo obtenido en la fase anterior, se seleccionarán los elementos adecuados para la captura de imágenes y el procesamiento de datos, se llevarán a cabo pruebas preliminares para validar la viabilidad técnica del proyecto. Se realizarán pruebas de concepto con los algoritmos de visión artificial para asegurar su funcionamiento adecuado y su capacidad para analizar las imágenes de manera precisa y confiable. Además, se evaluarán diferentes modelos de redes neuronales para determinar cuál es el más adecuado, considerando factores como la precisión y la eficiencia computacional.

Figura 1

Arquitectura genérica del prototipo electrónico



Nota: Se muestra la topología del funcionamiento del dispositivo electrónico, está dividido en tres etapas para comprender de mejor manera las actividades que se realiza en cada una de ellas.

En la fase de integración de bloques, se procederá a integrar los componentes individuales en un prototipo electrónico funcional y cohesivo. Se empleará un enfoque basado en captura de imágenes y procesamiento de datos, recopilados a través de un dispositivo de visión artificial conectado a una unidad de procesamiento central. Estos datos se utilizarán para entrenar un modelo de inteligencia artificial que pueda detectar la presencia de la enfermedad Botrytis cinerea en hojas de granadilla. Se utilizarán herramientas de desarrollo de código

abierto para el entrenamiento del modelo y la integración del sistema. Una vez entrenado y probado, el modelo se implementará en el prototipo electrónico para permitir la detección precisa de la enfermedad y proporcionar alertas proactivas para prevenir posibles daños en las plantas.

En la etapa de verificación del sistema y fase final, se llevarán a cabo pruebas exhaustivas del prototipo electrónico completo para asegurar su correcto funcionamiento y su conformidad con los requisitos establecidos en la detección de la enfermedad *Botrytis cinerea* en hojas de granadilla. Se realizarán pruebas de funcionamiento en condiciones reales de campo, donde el prototipo será utilizado para analizar muestras de hojas de granadilla y detectar la presencia de la enfermedad. Se evaluará la precisión del sistema y su capacidad para proporcionar alertas de manera efectiva, recopilando datos y retroalimentación de los agricultores o usuarios finales. Una vez completadas las pruebas y realizados los ajustes necesarios, se documentará todo el proceso de desarrollo, incluyendo los detalles de diseño, implementación y pruebas del prototipo electrónico.

1.5 Justificación

Las enfermedades de las plantas también representan un desafío significativo para la seguridad alimentaria global, ya que se estima que causan la pérdida de hasta un tercio de los cultivos alimentarios del mundo cada año (OIE, 2020). Esta situación genera escasez de alimentos, aumenta los precios y afecta la disponibilidad de una dieta saludable para las poblaciones más vulnerables (FAO, 2021).

Las enfermedades de las plantas representan una amenaza significativa para la seguridad alimentaria, el desarrollo sostenible y el bienestar humano a nivel global. Si bien no

existe un Objetivo de Desarrollo Sostenible (ODS) específico enfocado en esta problemática, su impacto se ve reflejado transversalmente en varios de ellos, tal como lo plantea la Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO) (FAO, 2021).

En primer lugar, las enfermedades de las plantas afectan directamente el Objetivo 1: Poner fin a la pobreza, ya que las pérdidas de cultivos ocasionadas por plagas y enfermedades pueden llegar hasta el 40% (FAO, 2021). Esto tiene un impacto devastador en la seguridad alimentaria mundial y los medios de vida de los agricultores, especialmente en países en desarrollo (FAO, 2021).

En segundo lugar, las enfermedades de las plantas también comprometen el Objetivo 2: Hambre cero, pues se estima que destruyen hasta un tercio de los cultivos alimentarios del mundo cada año (OIE, 2020). Esta situación genera escasez de alimentos, aumenta los precios y afecta la disponibilidad de una dieta saludable para las poblaciones más vulnerables (FAO, 2021).

Más allá de la seguridad alimentaria, las enfermedades de las plantas impactan negativamente en el Objetivo 3: Salud y bienestar. Algunas enfermedades son zoonóticas, es decir, pueden transmitirse de animales a humanos (FAO, 2021). Además, la pérdida de cultivos por enfermedades puede afectar la calidad de la dieta y contribuir a la malnutrición (FAO, 2021).

En relación con el Objetivo 12: Producción y consumo sostenibles, la promoción de prácticas agrícolas sostenibles que reduzcan el uso de pesticidas químicos nocivos y fomenten

el control biológico es crucial para proteger la salud del suelo y los ecosistemas (PNUMA, 2020).

Por otro lado, las enfermedades de las plantas ponen en riesgo el Objetivo 13: Acción por el clima, ya que el cambio climático está alterando los patrones de estas enfermedades, creando nuevas amenazas para los cultivos (OIE, 2020). La investigación para desarrollar variedades resistentes al clima es esencial para la adaptación de la agricultura al cambio climático y la protección de la seguridad alimentaria (OIE, 2020).

Si bien la Constitución del Ecuador de 2008 no cuenta con un capítulo o sección específica dedicada a la salud vegetal, sí establece diversos principios y disposiciones que reconocen la importancia de proteger los ecosistemas y la biodiversidad, elementos fundamentales para prevenir y controlar estas enfermedades (Asamblea Constituyente del Ecuador, 2008).

En este sentido, la Constitución respalda el derecho de la naturaleza a ser protegida (Artículo 1), el derecho a la alimentación (Artículo 415), la conservación de la biodiversidad (Artículo 416), la investigación y desarrollo de tecnologías sostenibles (Artículo 417) y la prevención y control de la contaminación ambiental (Artículo 418) (Asamblea Constituyente del Ecuador, 2008).

El avance continuo de la tecnología ofrece un sinfín de posibilidades para abordar los desafíos que enfrenta la agricultura, incluyendo el tratamiento y control de enfermedades que afectan los cultivos. En el caso específico de la enfermedad *Botrytis cinerea* en granadilla, la visión artificial emerge como una herramienta poderosa para la detección temprana,

permitiendo implementar estrategias de control más efectivas y sostenibles (Barbedo & Ling, 2019; Picon et al., 2020; Zhang et al., 2021).

La enfermedad *Botrytis cinerea*, comúnmente conocida como moho gris, representa una amenaza significativa para la producción de granadilla (*Passiflora edulis* f. *flavicarpa*) (Barbedo & Ling, 2019). Afecta hojas, flores y frutos, causando daños considerables que pueden ocasionar pérdidas económicas considerables para los agricultores (Picon et al., 2020). La detección temprana de esta enfermedad es crucial para implementar estrategias de control efectivas y minimizar su impacto (Barbedo & Ling, 2019).

Sin embargo, los métodos tradicionales de diagnóstico, como la inspección visual, son lentos, subjetivos y pueden pasar por alto los síntomas en etapas tempranas (Picon et al., 2020). Esto dificulta la toma de decisiones oportunas para el control de la enfermedad, lo que puede llevar a un aumento en las pérdidas de producción (Zhang et al., 2021).

En este contexto, la visión artificial emerge como una herramienta poderosa para la detección automatizada de *Botrytis cinerea* en hojas de granadilla (Barbedo & Ling, 2019; Picon et al., 2020; Zhang et al., 2021). Esta tecnología permite analizar imágenes de las hojas y detectar la presencia de la enfermedad mediante algoritmos de procesamiento de imágenes y aprendizaje automático (Barbedo & Ling, 2019).

También es importante como futuros profesionales, La investigación y el desarrollo de sistemas de visión artificial para la detección de enfermedades en la agricultura representan un campo de gran relevancia y con un futuro prometedor. Esta área de estudio tiene el potencial

de generar un impacto significativo en la producción de alimentos, la seguridad alimentaria y la sostenibilidad ambiental, contribuyendo a un futuro más próspero y saludable para el planeta.

CAPÍTULO II: FUNDAMENTACION TEORICA

En el segundo capítulo se presenta una recopilación de información relevante para el desarrollo de este proyecto. Se definen conceptos relacionados con la agricultura de precisión y su aplicación en la detección temprana de enfermedades en cultivos. Se incluye una descripción detallada del cultivo de granadilla (*Passiflora ligularis*), los parámetros que influyen en su desarrollo y producción, y las características de la enfermedad *Botrytis cinerea*. Finalmente, se abordan conceptos clave de visión artificial y aprendizaje automático, enfatizando su uso en la identificación y análisis de enfermedades en hojas de plantas.

2.1 Granadilla

La granadilla, también conocida como fruta de la pasión, pertenece a la familia de las pasifloráceas. De acuerdo con Ponce (2019) la granadilla es:

un bejuco trepador y enredador, de rápido crecimiento y sus raíces fibrosas y ramificadas, alcanzan una profundidad de 35 cm. El tallo es cilíndrico de coloración amarillo verdoso y de coloración claro en estado adulto. Las hojas son de forma acorazonada, verde intenso y nervaduras bien definidas en el envés (p. 11).

Es decir, es una planta trepadora que produce frutos amarillos y de distintas formas. El fruto se encuentra unido a un tallo cilíndrico que suele ir acompañado de hojas con forma acorazonada. Del mismo modo se puede agregar que es una planta de crecimiento rápido.

2.1.1 Origen y Distribución Geográfica de la Granadilla

La granadilla es originaria de Sudamérica, a nivel mundial, los principales destinos de exportación de la granadilla son países europeos como los Países Bajos, Italia, España, Portugal y Alemania, así como países asiáticos como Singapur y Hong Kong. Estas regiones, al carecer de un clima tropical adecuado para el cultivo de la granadilla, dependen de las importaciones para satisfacer la demanda local. Por esta razón, los principales exportadores de este producto

son países latinoamericanos que poseen las condiciones agroclimáticas ideales, abarcando desde México en el norte hasta Perú en el sur, específicamente de los estribos derivados de la cordillera andina. Cerdas & Castro (2018) mencionan que “el origen de la granadilla es América Tropical, por lo que se puede encontrar en forma silvestre desde México hasta Venezuela, y de Perú a Bolivia” (p. 11). Por tanto, su origen es de la cordillera andina. Además, el consumo de este producto se encuentra principalmente en la población local. No obstante, el incremento de la demanda de consumo por parte de otros países ha provocado un aumento de las áreas de cultivo.

Dentro de la provincia de Imbabura el cultivo de esta fruta está dispuesto en zonas con climas tropicales. Estas zonas van desde los 1600 a los 2700 metros sobre el nivel del mar (msnm). La producción de la granadilla, mayormente, se realiza por pequeños y medianos productores, quienes cosechan dos veces al año. De acuerdo con el manejo agronómico, la cosecha dura de dos a tres meses y se la realiza con una frecuencia de 15 días.

2.1.2 Características Físicas y Técnicas de la Granadilla

Algunas de las características físicas y técnicas de la granadilla se pueden encontrar en la Tabla 1.

Tabla 1

Ficha técnica de la granadilla

Características	Descripción
Familia	Passifloraceae
Variedades	Amarilla, morada y real
Nombre en inglés	Sweet granadilla, grenadia
Origen	Nativa
Lugar de origen	Sudamérica (Andes)
Hábito	Trepadora
Sinónimos y nombres comunes	Pasionaria, granadilla, parchita
Usos	Alimenticio: el fruto se consume en fresco

Nota. (Guerrero, 2018).

2.1.3 Cultivo y Sistema de Sostenimiento de Granadilla

La granadilla es una planta herbácea y de enredadera que requiere de un sistema de sostenimiento. El mejor sistema de tutorado es el emparrado. Con respecto a este sistema Ponce (2019) señala que:

en la periferia del lote se colocan postes de madera rolliza, vivos o muertos como: lechero, porotón, eucalipto, de 3 metros de largo, enterrados 1 metro, en la parte interna del emparrado, los postes se pueden colocar cada 10 m; posteriormente se refuerza con postes de menor resistencia como guadua, pambil, etc. En los postes principales se coloca alambre galvanizado de púa (de 3 hilos), y en sentido contrario a la pendiente se unen con alambre galvanizado número 12. Por último, para formar el enmallado o red, sobre la que van a descansar las ramas, se utiliza alambre galvanizado número 16, entrecruzándolos a una distancia de 50 cm entre cada uno (p. 13).

Por lo tanto, el emparrado es un sistema de soporte que permite que el cultivo de plantas enredaderas se produzca de manera controlada y optimizada.

2.1.4 Condiciones Agro-climatológicas para el Cultivo de Granadilla

Los cultivos de granadilla deben realizarse en suelos sueltos. Estos deben poseer una textura liviana, buena capacidad de drenaje, profundos y ricos en materia orgánica. También debe cumplir con otras condiciones, las cuales se encuentran presentes en la Tabla 2.

Tabla 2

Exigencias de cultivo

Agroecológicos	Características
Clima	Sub cálido, templado
Temperatura	12-17° (incluso 20°C)
Humedad	70 – 85%

Pluviosidad	600 – 1000 mm
Altitud	1800 – 2600 msnm
Requerimientos edáficos	
Textura	1800 – 2600 msnm
Acidez	pH 5.0 – 6.5
Tipo de suelo	Sueltos, aireados, ricos en materia orgánica
Salinidad	Es susceptible a ciertos niveles de salinidad

Nota. (Guerrero, 2014)

Todas estas características posibilitan que el cultivo de la granadilla se produzca de manera adecuada. Aquello provoca que se obtenga un buen producto.

2.1.5 Sistema de Cosecha de la Granadilla

En el proceso de recolección de los frutos es fundamental tomar en cuenta ciertas consideraciones, tales como época y tipo de cosecha.

Época. Se debe realizar cuando los frutos tienen, por lo menos, el 60% de la coloración amarilla clara. El color no debe ser pálido, puesto que impide obtener la calidad deseada. Además, la falta de coloración se asocia a la carencia de la composición de sus microelementos.

Tipo de Cosecha. La cosecha manual debe realizarse con guantes de algodón. Al momento de ejecutar el corte hay que conservar, por lo menos, 1,5 cm del pedúnculo. Esto se debe a que la cera de la cáscara es muy delicada y de esta depende la conservación de la calidad del fruto. Además, para recolectar el fruto se usan bolsas de yute, cajas de madera o mantas de algodón. (Ponce, 2019)

2.1.6 Clasificación y Selección de la Granadilla

El proceso de clasificación se puede realizar en tres partes. Este consiste en agrupar la fruta de acuerdo con el tamaño, estado y color de la cáscara. No obstante, en Colombia se aplica otro proceso de clasificación, tal y como se muestra en la Tabla 3. Mientras que el de selección

se elige la fruta que va a ser distribuida, mediante el descarte de aquellas que presenten daños biológicos, mecánicos o físico (Ponce, 2019).

Tabla 3

Clasificación de la granadilla

Calidad	Peso en gramos
Super extra	Más de 100g
Extra	De 90 a 100g
Primera	De 80 a 90g
Segunda	De 70 a 80g
Tercera	Menos de 70g

Nota. (Gonzales, 2013)

Una clasificación adicional es la realizada por OSTEPEC (2009), en la que toma en cuenta las dimensiones de la fruta. Las dimensiones se presentan de la siguiente forma:

- Extra mide de 7,6 en adelante
- La primera mide de 7,0 a 7,5 cm de diámetro
- La segunda mide de 6,5 a 6,9 cm de diámetro
- La tercera mide de 6,0 a 6,4 cm.

Una vez concluido el proceso de postcosecha se somete a la granadilla al proceso de empaque y transporte.

Empaque. Se empaca la fruta en cajas de cartón. Entre cada capa de fruta se dispone de una hoja de periódico para evitar el contacto entre granadillas. Esto evita que se manchen o rayen.

Transporte. Se transporta en camionetas y camiones. Al momento del transporte es fundamental que las frutas no queden expuestas al sol o a la lluvia de forma directa. Por lo tanto, deben ir cubiertas. Además, esto impedirá que las frutas desarrollen antracnosis (Ponce, 2019).

2.1.7 Comercialización de la Granadilla

La comercialización consiste en ofertar un producto. A este se le otorga condiciones y vías que permiten la distribución de este. Baca (2013) indica que “la comercialización es la actividad que permite al productor hacer llegar un bien o servicio al consumidor con los beneficios de tiempo y lugar” (p. 64). Por tanto, es todo el proceso que permite al usuario adquirir el producto.

La comercialización es el proceso que se sigue después que haya concluido la producción de la granadilla. En este proceso se deben conocer los componentes que integran la cadena de distribución. Es decir, desde la plantación de la granadilla hasta que llega a el usuario.

2.2 Botrytis

La botrytis es un hongo que afecta a las hojas, tallos, flores y frutos de una planta. Esto se debe a que es un agente causal del moho gris, el cual se puede encontrar durante el desarrollo del cultivo y la postcosecha. Además, al ser un patógeno versátil, es capaz de destruir cualquier tejido u órgano de la planta. Aquello provoca que esta muera inmediatamente y no se detecta brevemente (ADAMA, 2022).

2.2.1 Clasificación Taxonómica

La clasificación taxonómica hace referencia a los aspectos que permiten catalogar a un organismo. Agrios (1997) menciona que la clasificación de la Botrytis es la siguiente:

- **Nombre común:** Moho gris
- **Nombre científico:** Botrytis cinerea
- **Clase:** Deuteromycetes
- **Orden:** Moniliales
- **Género:** Botrytis
- **Especie:** Cinerea

2.2.2 Signos y Síntomas

La manifestación del moho gris se evidencia a partir de la parte afectada de la planta o del estado fisiológico que tengan los tejidos. Las hojas que se encuentren infectadas pueden no mostrar sintomatología aparente. No obstante, cuando las hojas infectadas maduren se producirá una cubierta gris aterciopelada (Mekawi et al., 2019).

2.2.3 Hospederos de *B. cinérea*

En el género *Botrytis* se encuentra la especie cinérea. Esta es una especie polífaga que afecta a más de 200 tipos de plantas en el mundo. El patógeno puede atacar hortalizas, como: tomate, lechuga, pimiento, zanahoria, pepino y berenjena; frutales, como: granadilla, fresa, uva, manzana, plátano, arándanos, naranjas, peras; y ornamentales, como crisantemo, rosa, clavel, violeta, begonia, dalia, entre otras (García, 2017).

2.2.4 Manejo de la enfermedad

El hongo puede permanecer por mucho tiempo en el suelo, por ejemplo, en desechos de plantas y frutos. Por tanto, para evitar su propagación se debe recolectar y quemar el material que se encuentre afectado. Además, resulta importante podar las ramas secas, deshierbar o hacer nuevas plantaciones. Esto se debe a que se favorece la aeración del cultivo y se evita el desarrollo de enfermedades. Asimismo, se recomienda fertilizar y cosechar los frutos en su óptimo punto de maduración, hay que tratar de no dejar madurarlos en la planta (Rondón, 2017).

2.2.5 Principales problemas fitosanitarios

El hongo también causa lesiones de color café claro. Como se muestra en la Figura 2 estas afectaciones suelen presentar un crecimiento afelpado de color café oscuro. En este crecimiento se puede observar las esporas y conidióforos del hongo que provoca la enfermedad. Además, cuando la enfermedad aparece en frutas que ya estaban formados el moho gris causa una pudrición húmeda. También hay que tener en cuenta que los frutos verdes son susceptibles

a la enfermedad si se presentan las condiciones adecuadas para el desarrollo del microorganismo (Tamayo, 2021).

Figura 2

Ciclo de Botrytis



Nota. (Jesús Martínez, 2024)

2.2.6 Control

Un buen control de la enfermedad depende del monitoreo permanente que se brinde al cultivo, sobre todo en épocas húmedas. Para evitarla hay que mantener una buena aireación dentro del cultivo, eliminar los tallos y ramas que presentan los síntomas de la enfermedad y manejar químicos con fungicidas (Rondón, 2019).

Control Química. Se basa en el empleo de fungicidas. Este es uno de los medios más usados para controlar el moho gris. No obstante, cabe resalta que Botrytis es conocida por su resistencia a los fungicidas, debido a su variabilidad genética y la producción de esporas (Matute, 2019).

Control Cultural. Se refiere al manejo integrado de la enfermedad. Algunas de las medidas recomendadas dentro de este tipo de control son:

- Ampliar el espacio entre las plantas, con el fin de permitir el movimiento de aire y el secado de las plantas de fresa.
- El uso de riego a goteo es una práctica ideal para reducir problemas con el moho gris y se ha establecido como una práctica muy común.
- Hacer podas y deshojados al ras del tallo, con el fin de no dejar tocones que contribuyan al desarrollo de la enfermedad.
- Controlar los niveles de nitrógeno en el suelo, debido que los niveles elevados del mismo favorecen el desarrollo del moho gris (Matute, 2019).

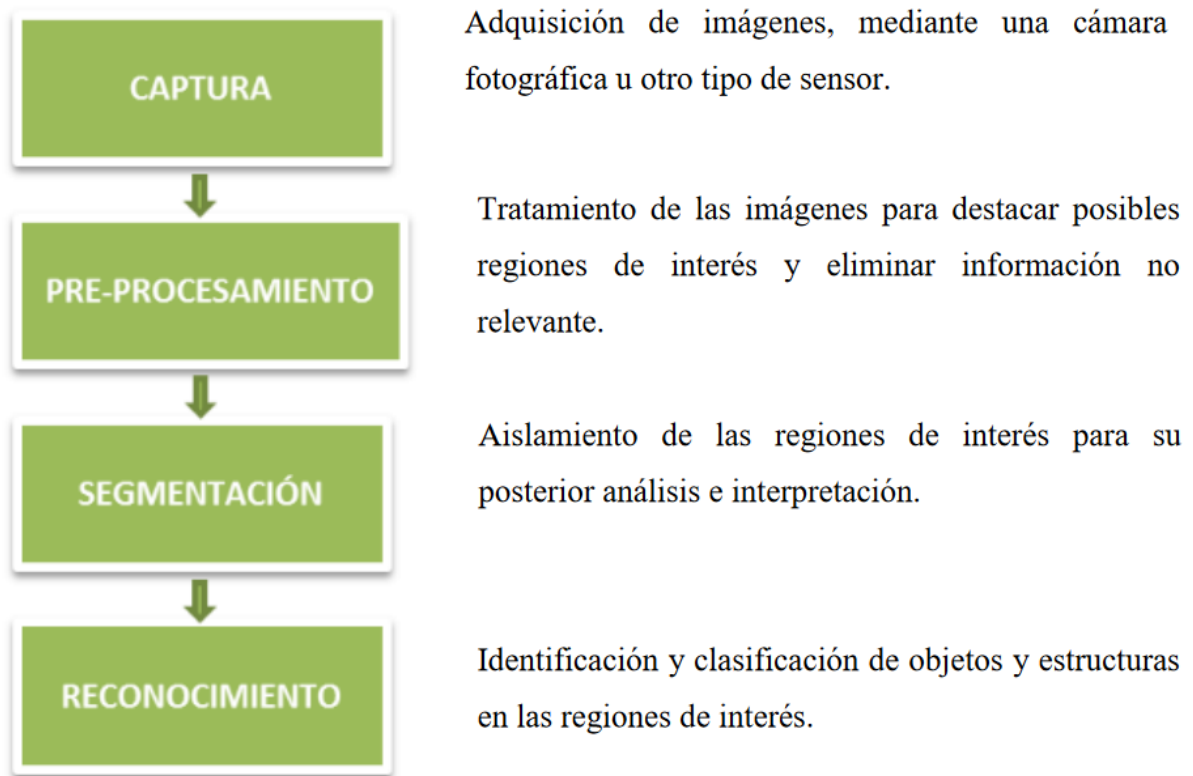
Control Biológico. Se usan organismos vivos con el fin de controlar las poblaciones que generan problemas. Para aplicarlo se deben tener conocimientos que permitan identificar qué organismo debe emplearse para controlar la enfermedad (García, 2017). Además, el control biológico permite la reducción de residuos, no genera problemas de resistencia, no supone riesgos para la salud humana y su aplicación no es una práctica compleja. Para ello es necesario ciertos conocimientos técnicos (Agrios, 2020).

2.3 Inteligencia Artificial

La visión artificial se desarrolla dentro de las aplicaciones de la inteligencia artificial para adaptarse a las necesidades del ser humano. Núñez (2016) define a la visión artificial como “una rama de la inteligencia artificial cuyo propósito es diseñar sistemas informáticos capaces de “entender” los elementos y características de una escena o imagen del mundo real” (p. 7). Al denotar estos sistemas se integra un proceso que permiten extraer información tanto numérica y simbólica a partir de una imagen. Con respecto a este proceso se pueden identificar los campos fundamentales que muestra la Figura 3.

Figura 3

Sistema de visión artificial



Nota. (Núñez, 2016)

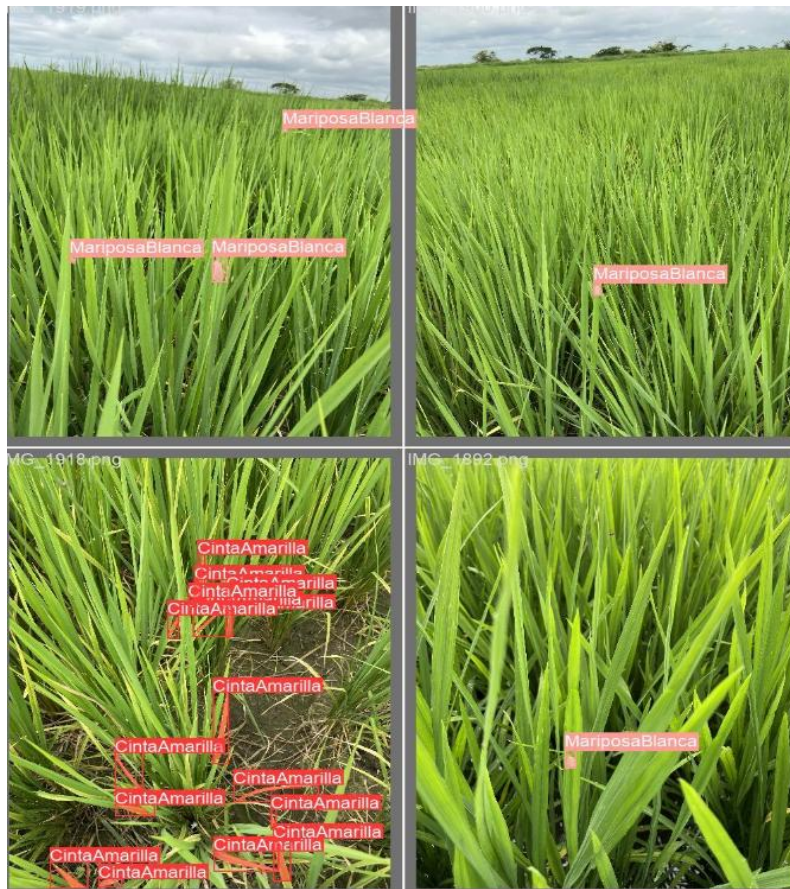
2.3.1 Reconocimiento Automático De Objetos

El reconocimiento automático de objetos es una fase muy importante, debido a que contiene muchas aplicaciones. Además, se relaciona con la detección de multitudes, conteo de personas e identidades. Asimismo, esta técnica está relacionada con el procesamiento de imágenes y el reconocimiento de patrones. Estos permiten detectar áreas de interés, lo que logra un funcionamiento óptimo.

En la identificación de objetos se debe considerar si el sistema funciona al distinguir la presencia de algún elemento designado en la etapa de clasificación de las imágenes. Luego, en la etapa de detección para saber cuál es el elemento que se designó. Además, en esta última etapa, debe considerar las características propias, tanto de morfología como de comportamiento. Un ejemplo de ello se evidencia en la Figura 4, en la que el sistema reconoce a tres hojas con diferentes características.

Figura 4

Detección de elementos



Nota. (Yostin Jahir Burgos Muñiz, 2024)

2.3.2 Etapas de Detección de Objetos de una Imagen

Las etapas de detección de objetos de una imagen son las siguientes:

Adquisición de Imagen. El sistema debe estar integrado con cámara, digitalizador e iluminación para poder obtener una imagen con óptimas prestaciones. Esta fase permite localizar los elementos que deben adecuarse antes de implementar un algoritmo complejo.

Preprocesamiento de las Imágenes. Se realiza el tratamiento de las imágenes y sirve para identificar algunas regiones o áreas que resulten de interés. Además, permite eliminar la información que ya no es necesaria.

Segmentación. Se etiqueta a las imágenes que tienen una gran limitación, debido a que se deben considerar las condiciones que dificulten las tareas complejas. Es por esta razón que

se debe aislar las regiones de interés para que en el transcurso del proceso se desarrolle de una mejor manera el análisis e interpretación. Aquí se denota a los pixeles pertenecientes al objeto con el valor de uno, y a los que pertenecen con calor de cero (Antontio, 2016).

Reconocimiento del Objeto. En este proceso se identifica los elementos que se encuentran en la entrada, que es en donde la imagen pasa por el clasificador y puede reconocer los objetos. Luego se obtiene un resultado a la salida del elemento de interés con una precisión mejor (Pineda, 2019).

2.3.3 *Aprendizaje Supervisado*

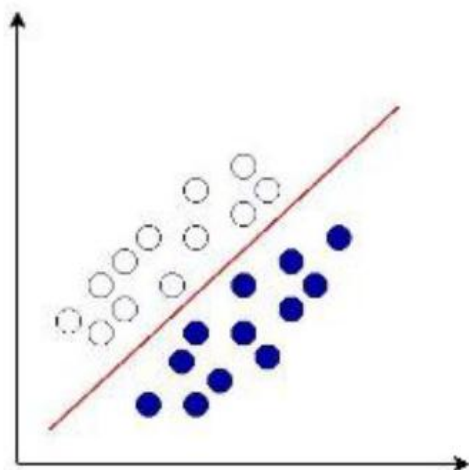
El aprendizaje supervisado consiste en la inferencia desde los datos de entrada y salida que fueron previamente clasificados. Posteriormente se entrena a los algoritmos a través de la modificación de pesos de la sinapsis. Esto permite obtener un resultado óptimo y eficaz (Pineda, 2019).

En relación con el dominio de salida, se obtiene dos tipos de algoritmos, los cuales son:

Clasificación. En relación con el resultado de sus muestras es de manera discreta. Es decir, una serie de eventos que pueden ser ciertos (Carlos & Pérez, 2018). Tal y como se evidencia en la Figura 5.

Figura 5

Clasificación lineal

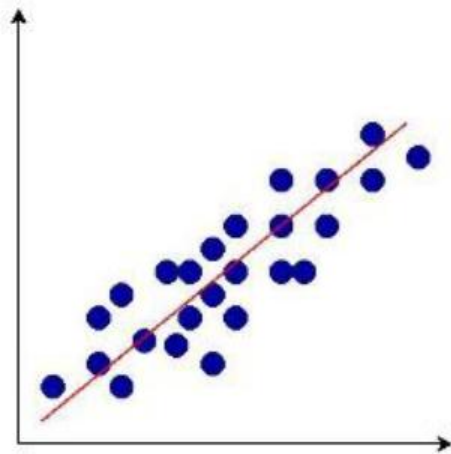


Nota. (Carlos & Pérez, 2018)

Regresión. Se requiere si es un valor continuo. Por lo tanto, se refiere a todo evento que no tenga interrupciones. Esto se puede evidenciar en la Figura 6.

Figura 6

Regresión Lineal



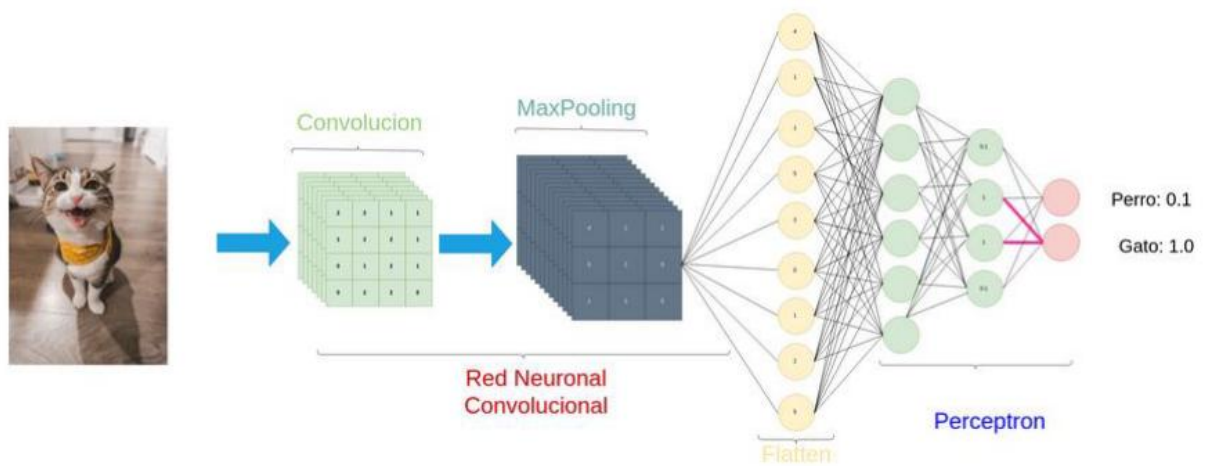
Nota. (Carlos & Pérez, 2018)

2.3.4 Redes Neuronales Convolucionales (CNN).

Es una red multicapa jerárquica que permite la extracción de características en capas ocultas. Por lo general, se conforman de 5 a 7 capas. Además, su arquitectura es similar a las redes neuronales ordinarias, puesto que integra pesos y sesgos para aprender (López, 2022). En la Figura 7 se visualiza la estructura de una red convolucional.

Figura 7

Estructura de una red convolucional



Nota. (López, 2022)

2.3.5 *Función de Activación*

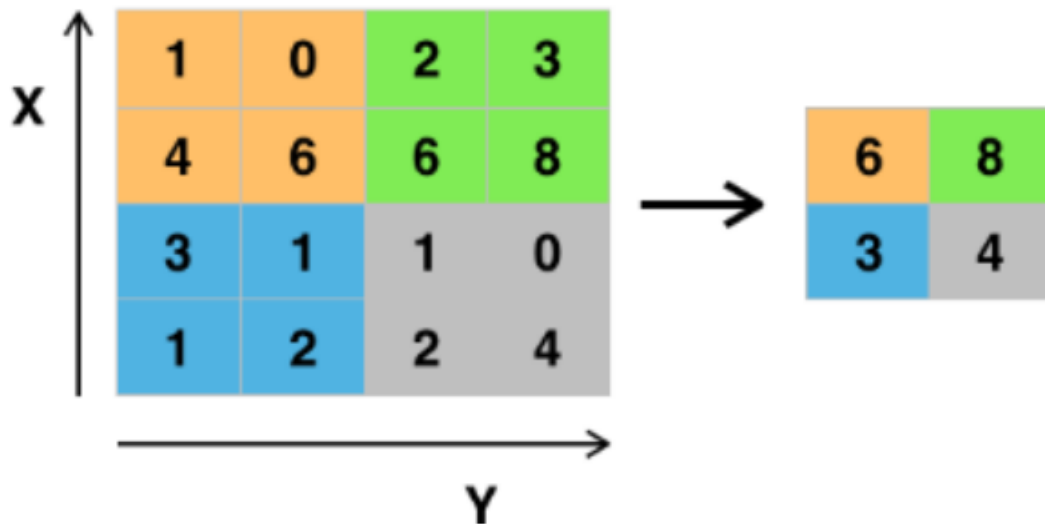
Al terminar la operación de convolución resulta como salida una función lineal. Por lo que, se debe evitar el encadenar varias capas de neuronas que tengan como salida la función lineal. Esto se debe a que si se intenta hacerlo la red colapsará y no se podrá trabajar en una red multicapa. Para solucionar este problema se debe usar funciones de activación para introducir la no linealidad (Enyinna et al. 2018).

2.3.6 *Capa de Pooling*

La capa de agrupación o pooling se encarga de simplificar la información en la salida de la capa convolucional. Por tanto, disminuye las dimensiones de los mapas de características con el propósito de conservar únicamente las características más relevantes, reducir parámetros y, por ende, el coste computacional (Nilsen, 2018). Tal como muestra la Figura 8.

Figura 8

Ejemplo de funcionamiento de la capa de pool



Nota. (Cicero, 2018)

2.3.7 Capa Directamente Conectada

Una vez concluida la serie de capas convolucionales y de pooling, se debe extraer las características de más alto nivel. Estas alimentan a las capas directamente conectadas, con el fin de realizar la tarea de clasificación según el conjunto de datos registradas en la etapa de entrenamiento. Además, para la tarea de clasificación multiclase se utiliza la función softmax. Esta normaliza los valores de la salida de la última capa completamente conectada, a las probabilidades de cada clase objetivo, donde cada valor se encuentra entre 0 y 1 y la suma de estos valores es 1 (Yamashita et al., 2018).

2.4 Aumento de datos en visión artificial

El aumento de datos es una técnica ampliamente utilizada en visión artificial y aprendizaje profundo cuyo objetivo es incrementar de manera artificial el tamaño y la diversidad de un conjunto de datos de entrenamiento sin necesidad de adquirir nuevas muestras. Esta estrategia permite mejorar la capacidad de generalización de los modelos, reduciendo el sobreajuste y aumentando su robustez frente a variaciones presentes en escenarios reales.

Dentro de las técnicas de aumento de datos más empleadas se encuentran las transformaciones geométricas, tales como la rotación, el escalado, la traslación y el volteo de

imágenes. En particular, la rotación de imágenes resulta especialmente relevante en aplicaciones de clasificación de objetos naturales, como hojas, debido a que la orientación del objeto durante la captura no afecta sus características fisiológicas o visuales intrínsecas. De este modo, al aplicar rotaciones en ángulos definidos (45° , 135° , 180° y 225°), se simulan distintas posiciones de la hoja, permitiendo que el modelo aprenda características invariantes a la orientación.(Shorten & Khoshgoftaar, 2019)

2.5 Prototipado electrónico

El prototipado electrónico es un proceso fundamental en el desarrollo de sistemas de monitoreo, combinando componentes de hardware y software para garantizar un funcionamiento eficiente. En el hardware, se utilizan microcontroladores y minicomputadoras que permiten la captura, procesamiento y transmisión de datos en tiempo real. En cuanto al software, diferentes lenguajes de programación facilitan la comunicación con los dispositivos y el manejo de la información. Además, se emplean herramientas de almacenamiento y plataformas en la nube para desarrollar, probar e implementar soluciones avanzadas, optimizando la supervisión remota y la toma de decisiones basada en datos.

2.5.1 Componentes de Hardware

En este apartado se representa una descripción de los principales componentes de hardware empleados en el desarrollo del sistema de monitoreo. Se abordan los dispositivos electrónicos esenciales, como microcontroladores y módulos de comunicación inalámbrica, los cuales permiten la recopilación, procesamiento y transmisión de datos en tiempo real. Estos elementos son fundamentales para garantizar un monitoreo eficiente y confiable, facilitando la integración de tecnologías avanzadas para la supervisión remota y la toma de decisiones basadas en datos preciso.

2.5.1.1 Microcontroladores

Los microcontroladores al ser dispositivos programables diseñados para ejecutar operaciones matemáticas complejas con gran rapidez funcionan como minicomputadoras integradas, permitiendo la captura, procesamiento monitoreo y control de datos a través de sus diversos puertos de entrada y salida. Gracias a su versatilidad y eficiencia, estos componentes son esenciales en sistemas embebidos, automatización y aplicaciones de monitoreo en tiempo real. (Carabalí Méndez, 2020)

2.5.1.1.1 Arduino

Arduino es una plataforma que se utiliza tanto como para software y hardware y es de código abierto, que integra un microcontrolador programable. La placa cuenta con una serie de pines de entrada y salida que facilitan la conexión de diversos sensores y actuadores, permitiendo la interacción con el entorno. La programación y configuración del microcontrolador se realizan a través del entorno de desarrollo integrado (IDE) de Arduino, utilizando el lenguaje C++. Además, la carga del código de la placa se efectúa mediante diferentes tipos de cables USB, lo que facilita el uso en proyectos de automatización, robótica y sistemas de monitoreo. (Kouhia, 2019).

2.5.1.1.2 Raspberry Pi

La Raspberry pi es un miniordenador de bajo costo que destaca por su diseño compacto y versátil. El precio varía dependiendo de cada modelo y según sus especificaciones, incluyendo la capacidad de memoria RAM, el almacenamiento interno y los módulos de comunicación integrados, como Bluetooth, Wi-Fi y Ethernet, así como la cantidad y tipo de puertos disponibles en la placa. Este dispositivo incorpora un procesador basado en la arquitectura ARM y memoria RAM, lo que optimiza su rendimiento para diversas aplicaciones. De forma predeterminada, utiliza Raspbian, un sistema operativo basado en Debian, aunque es

compatible con otras distribuciones de GNU/Linux, ampliando sus posibilidades de uso en proyectos de automatización, monitoreo y desarrollo tecnológico.

2.5.1.1.3 Nvidia Jetson Nano

La Nvidia Jetson Nano es una placa de desarrollo similar a la placa Raspberry Pi, pero específicamente está diseñada para tareas de deep learning. La GPU permite realizar procesos de inferencia de manera local y en tiempo real, aunque esta placa también puede utilizarse para entrenar redes neuronales, su diseño está más orientado a la ejecución de modelos ya entrenados, optimizando el rendimiento en aplicaciones de inteligencia artificial y visión por computadora.

2.5.2 Componentes de Software

A continuación, se detallan los diferentes softwares y lenguajes de programación que juegan un papel crucial en el desarrollo de un sistema de monitoreo inteligente, destacando las herramientas que proporcionan una mayor eficiencia y facilidad de manejo. Además, se abordan los aspectos relacionados con el alojamiento de los datos y los servidores web necesarios para garantizar el correcto funcionamiento y el acceso remoto del sistema.

2.5.2.1 Lenguajes de Programación

Todo hardware requiere de un lenguaje de programación para ejecutar las ordenes definidas en líneas de código. Por lo tanto, se detallan los diferentes lenguajes de programación utilizados en el desarrollo del proyecto, los cuales permiten interactuar con el hardware y llevar a cabo las funciones necesarias para su correcto funcionamiento.

2.5.2.1.1 Lenguaje de Programación Python

Python es un lenguaje de programación interpretado de alto nivel que ofrece soporte multiplataforma y multiparadigma. Se destaca por su sintaxis clara y legible, lo que facilita su comprensión y mantenimiento. Es un lenguaje de código abierto y orientado a objetos, lo que

permite el trabajo con funciones y la organización eficiente del código en proyectos de diversas complejidades.

2.5.2.1.2 Lenguaje de Programación C++

C++ es un lenguaje de programación basado en el lenguaje C, creado inicialmente por Bjarne Stroustrup. Es un lenguaje versátil y potente, ampliamente utilizado para el desarrollo de aplicaciones que requieren alto rendimiento. C++ es compatible con diversas plataformas, lo que lo convierte en una opción ideal para una amplia gama de proyectos. Sin embargo, uno de sus inconvenientes es la complejidad en el manejo de bibliotecas.

A continuación, se detallan las características principales de este lenguaje de programación:

- Ofrece compatibilidad con una amplia variedad de bibliotecas.
- Es un lenguaje orientado a objetos, lo que permite una organización modular y reutilizable del código.
- Brinda alta velocidad en la ejecución de programas, lo que lo hace ideal para aplicaciones de alto rendimiento.
- Es didáctico, lo que facilita su aprendizaje y comprensión, especialmente en entornos educativos y proyectos de formación.

2.5.2.1.3 Lenguaje de Programación JSON

JSON (JavaScript Object Notation) es un formato estándar basado en texto para la representación estructurada de datos, utilizando una sintaxis derivada de los objetos JavaScript. Es ampliamente utilizado para el intercambio de datos entre sistemas y se considera un lenguaje de transporte. A continuación, se detallan sus características principales:

- Fácil lectura y comprensión del código.
- Tamaño de transferencia ligero, lo que mejora la eficiencia en la transmisión de datos.
- Basado en JavaScript, lo que lo convierte en una opción ideal para aplicaciones web.

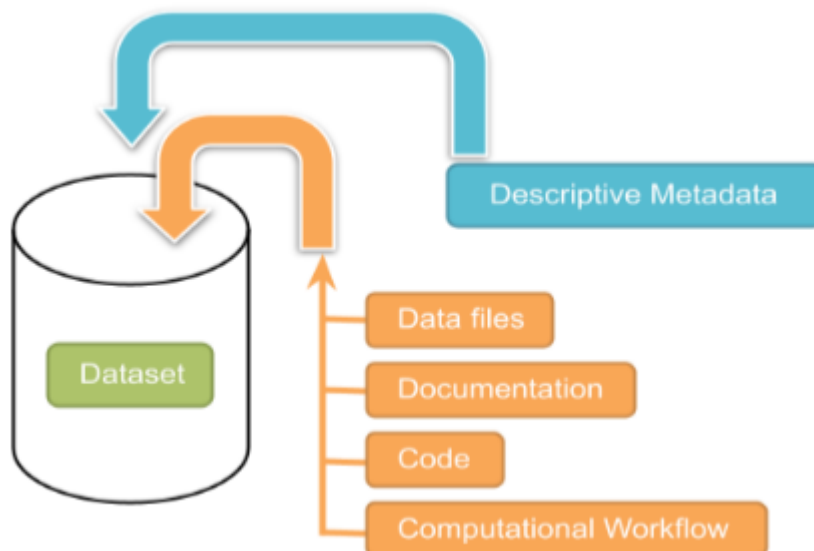
- Utiliza texto plano con codificación UTF-8, asegurando compatibilidad y facilidad de manejo en diversas plataformas.

2.5.3 Almacenamiento de Datos

El almacenamiento de datos (dataset) se refiere a un conjunto de datos, generalmente organizados en forma tabular, donde cada columna representa una variable y cada fila corresponde a un registro específico. Estos datos se seleccionan y almacenan con el fin de realizar investigaciones, análisis y tomar decisiones estratégicas como se muestra en la Figura 12. Al procesar estos conjuntos de datos, es posible extraer conocimientos y patrones ocultos que facilitan la comprensión de fenómenos y la identificación de tendencias basadas en evidencias. El almacenamiento de datos es ampliamente utilizado en áreas como estadística, investigación científica, análisis de datos y machine learning, donde la capacidad de manejar grandes volúmenes de información es esencial. (Deusto Formación., 2022)

Figura 9

Dataset almacenamiento de datos



Nota. (Institute for Quantitative Social Science, 2021)

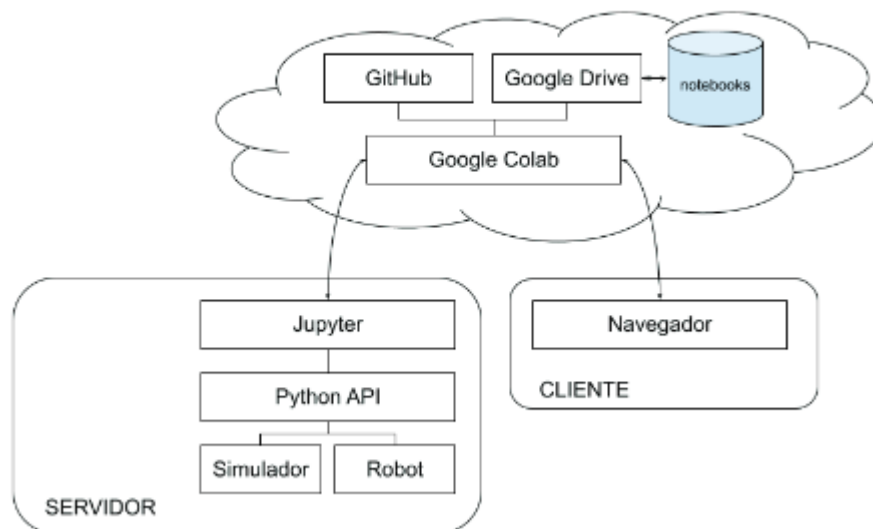
2.5.4 Google colab

Colab, también conocido como "Colaboratory", es un entorno de ejecución basado en servidores Jupyter, diseñado para integrarse fácilmente con Google Drive y GitHub. Permite ejecutar notebooks tanto en la nube como en un entorno local. Esta última opción es especialmente útil para ejecutar código en simuladores o robots, proporcionando un espacio flexible y accesible para el desarrollo y prueba de proyectos de programación.

En la Figura 13 se presenta la configuración de Google Colab conectada al entorno local del servidor, lo que permite una integración eficiente entre el entorno en la nube y los recursos locales para ejecutar y probar el código de manera más efectiva. (Enric Cervera, 2022)

Figura 10

Arquitectura de Google Colab



Nota. (Enric Cervera, 2022)

CAPÍTULO III: Diseño del Sistema

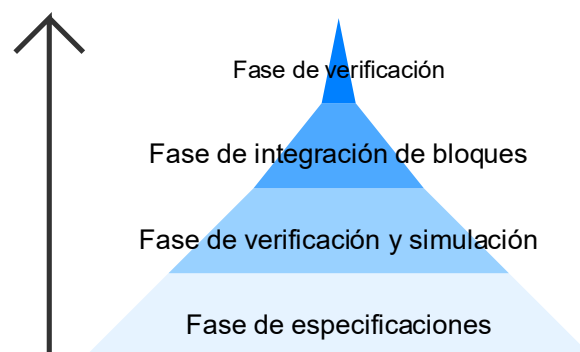
En el tercer capítulo se presenta la metodología empleada para el desarrollo del prototipo electrónico, comenzando con el análisis de los requisitos necesarios para la selección de los componentes de hardware y software. Estos son esenciales para crear un sistema que permita la detección temprana de la *Botrytis cinerea* en hojas de granadilla (*Passiflora ligularis*), utilizando inteligencia artificial. Además, se realizó un análisis de la situación actual en cuanto a los métodos de detección en cultivos, seguido de la identificación de los actores clave. A partir de esta información, se definen los requisitos del sistema para asegurar que el diseño, implementación y validación del prototipo cumplan con los criterios establecidos.

3.1 Metodología

El enfoque metodológico adoptado para el desarrollo de este proyecto es de tipo Bottom-Up, que se caracteriza por la construcción progresiva del sistema, comenzando desde los bloques individuales hasta alcanzar un sistema integrado y funcional. Este enfoque permite trabajar en la implementación y validación de cada componente de manera escalonada, asegurando su correcto funcionamiento antes de integrarlo al sistema final. A continuación, se describen las fases que estructuran este proceso como se muestra en la Figura 11, detallando cada una de las etapas clave del desarrollo.

Figura 11

Fases de la metodología



Nota: Adaptado de: (Muñoz, 2010)

En la etapa de recopilación de especificaciones, se realizó una revisión bibliográfica mediante un enfoque de investigación aplicada, lo que proporcionará una base sólida y fundamentada para el diseño del prototipo electrónico.

La fase de simulación y validación se enfocó en identificar los requisitos del sistema, creando los primeros diagramas y modelos que permitirán comprobar la viabilidad del diseño propuesto.

Durante la etapa de integración de bloques, se desarrolló y unió los distintos componentes del sistema, formando una estructura que permita su funcionamiento adecuado.

Por último, en la fase de verificación final, se llevó a cabo instalaciones y pruebas exhaustivas del sistema, con el objetivo de garantizar que todas las funciones operen correctamente y que se cumplan los objetivos establecidos al inicio del proyecto.

3.2 Análisis

En esta sección se llevó a cabo un análisis exhaustivo de la situación actual, complementado con información proveniente de investigaciones previas, para comprender las dimensiones, las condiciones ambientales y las posibles limitaciones que podrían surgir durante el desarrollo del proyecto. Asimismo, se definió los requisitos iniciales del sistema, abarcando tanto las necesidades de hardware como de software esenciales para su implementación.

3.2.1 Situación Actual

El presente proyecto se llevó a cabo en un huerto de granadillas (*Passiflora ligularis*) ubicado en la parroquia de Monte Olivo, comunidad Palmar Grande como muestra la Figura 12, en una zona de producción al aire libre. Las condiciones climáticas de la parroquia Monte Olivo presentan una notable variabilidad térmica a lo largo del año, con temperaturas que oscilan entre los 11 °C y 24 °C, y una media anual que varía entre 10 °C y 16 °C, dependiendo

de la altitud. El clima de la zona se clasifica como mesotérmico semihúmedo, con una precipitación anual que fluctúa entre 1 000 y 2 000 mm. La humedad relativa se mantiene elevada, con valores comprendidos entre el 65 % y el 85 %. Durante los meses de junio y julio, se registran los niveles más bajos de precipitación, con promedios mensuales de entre 112 y 153 mm, caracterizando una temporada relativamente seca.

Como se muestra en la Figura 12 en cuanto a su geografía, Monte Olivo se sitúa entre los 1 600 y 3 800 metros sobre el nivel del mar, distribuyéndose en tres pisos altitudinales bien diferenciados: una zona de valle, una franja intermedia con pendientes moderadas y una zona alta con pendientes pronunciadas. La parroquia abarca una superficie aproximada de 65,68 km².

El huerto cuenta con un sistema de riego tecnificado por aspersión, sin embargo, la falta de herramientas de monitoreo adecuadas dificulta la detección temprana de enfermedades como *Botrytis cinerea*, la cual afecta el crecimiento de las plantas y puede causar la caída prematura del fruto. Actualmente, no se dispone de una infraestructura especializada para monitorear la salud de las plantas ni de un sistema de evaluación para controlar la calidad del producto cosechado. Además, los datos existentes sobre las prácticas de fumigación y manejo de la planta son limitados, ya que provienen solo de los agricultores sin registros formales.

El prototipo electrónico basado en inteligencia artificial para la detección temprana de enfermedades en las hojas de la granadilla permitió no solo identificar la presencia de *Botrytis cinerea*, sino también proveer alertas sobre el estado de las plantas, las cuales podrán ser monitoreadas en tiempo real a través de una plataforma digital accesible desde dispositivos inteligentes. Con esta implementación, se logra un mejor rendimiento y una producción de mayor calidad, optimizando los recursos y satisfaciendo las necesidades del agricultor.

Ubicación geográfica de la finca



Para la recopilación de los requisitos del sistema se utilizó diversas técnicas de obtención de información, incluyendo entrevistas y análisis de soluciones existentes. Esto permite identificar los componentes de hardware más adecuados para el diseño del prototipo, así como las herramientas de software y los algoritmos óptimos para su aplicación en sistemas de visión artificial.

La entrevista realizada a la Sra. Marcia Robles, quien es la encargada de la plantación de granadilla en la parroquia de Monte Olivo, tiene el objetivo de recopilar información relevante sobre el ciclo de producción de la granadilla, los períodos de mayor incidencia de *Botrytis cinerea*, las prácticas de manejo utilizadas para su prevención y control, así como las necesidades y expectativas del agricultor e ingeniero agrónomo respecto a la implementación de un sistema de detección temprana basado en visión artificial.

El cuestionario emplea un enfoque mixto, combinando preguntas de tipo abierto y cerrado. Se detallan los cuestionamientos formulados para la recopilación de información y el cuestionario se muestra en su formato apropiado en el ANEXO A.

A continuación, se detallan las preguntas dirigidas al agricultor responsable de la plantación:

- ¿Cuánto tiempo lleva cultivando granadilla?
- ¿En qué temporada del año ha notado mayor incidencia de Botrytis cinerea en sus cultivos de granadilla?
- ¿Cuáles han sido las principales dificultades que ha enfrentado al tratar de controlar Botrytis cinerea en su cultivo?
- ¿Cómo identifica actualmente la presencia de Botrytis cinerea en sus cultivos?
- ¿Qué métodos utiliza para prevenir o tratar Botrytis cinerea en su cultivo?
- ¿Cuánto le afecta económicamente la presencia de Botrytis cinerea en sus cultivos?
- ¿Qué tan efectiva considera la inspección visual para detectar la enfermedad?
- ¿Qué espera de un sistema basado en visión artificial para la detección de Botrytis cinerea?
- ¿Cómo preferiría recibir alertas sobre la detección temprana de la enfermedad?
- ¿Estaría dispuesto a utilizar un prototipo tecnológico para la detección temprana de Botrytis cinerea en sus cultivos?

A continuación, y como parte del proceso de recopilación de información técnica y operativa, se presentan de manera detallada las preguntas dirigidas al ingeniero agrónomo encargado del proyecto. Estas interrogantes tienen como propósito profundizar en aspectos clave relacionados con la planificación, supervisión y manejo agronómico de la plantación, así

como en la toma de decisiones estratégicas que afectan directamente al desarrollo y rendimiento del cultivo.

- ¿Cuánto tiempo lleva trabajando en el sector agrícola?
- ¿En qué etapa fenológica de la planta de granadilla suele aparecer con mayor frecuencia Botrytis cinerea?
- ¿Cuáles considera que son los factores ambientales más determinantes para la proliferación de Botrytis cinerea en granadilla?
- ¿Qué métodos recomienda actualmente para la detección temprana de Botrytis cinerea?
- ¿Considera que la inspección visual tradicional es suficiente para la detección oportuna de Botrytis cinerea?
- ¿Ha trabajado con herramientas tecnológicas para la detección de enfermedades en cultivos?
- ¿Cree que un sistema basado en visión artificial podría mejorar la detección temprana de la enfermedad en comparación con los métodos actuales? ¿Por qué?
- ¿Qué características considera esenciales en un prototipo de detección de enfermedades en cultivos para que sea útil en el campo?
- ¿Qué tan dispuesto estaría a recomendar el uso de un sistema basado en visión artificial para la detección de Botrytis cinerea?
- ¿Qué otras enfermedades o problemas agrícolas cree que podrían beneficiarse del uso de visión artificial para su detección temprana?

3.1.1.2 Análisis de la entrevista

La entrevista realizada a la Sr. Marcia Robles encargada del cultivo de la granadilla tuvo como objetivo recopilar información clave sobre el cultivo de granadilla, incluyendo el

impacto de *Botrytis cinerea*, las condiciones ambientales que favorecen su aparición, los métodos de detección utilizados y las estrategias de manejo implementadas para su control.

De las preguntas 1, 2 y 3, se identificó que el agricultor cuenta con experiencia de entre 1 y 5 años en el cultivo de granadilla, y ha observado una mayor incidencia de *Botrytis cinerea* durante la temporada de lluvias. La principal dificultad enfrentada está relacionada con la resistencia de la enfermedad en ciertas temporadas, lo que requiere fumigaciones frecuentes en intervalos de 8 a 15 días.

Respecto a las preguntas 4, 5 y 7, se evidencia que actualmente el método de identificación de la enfermedad se basa principalmente en la inspección visual propia, considerada como moderadamente efectiva. Para el tratamiento y prevención, el agricultor utiliza tanto la aplicación de fungicidas como la eliminación de hojas afectadas, lo que demuestra un enfoque combinado pero limitado a métodos tradicionales.

De las preguntas 6, 8, 9 y 10, se desprende que aunque el impacto económico de la enfermedad es percibido como "poco", el agricultor muestra un claro interés en la implementación de nuevas tecnologías. Específicamente, espera que un sistema basado en visión artificial proporcione alertas tempranas sobre la presencia de la enfermedad, prefiriendo recibir estas notificaciones a través de mensajes SMS. Adicionalmente, manifestó disposición para utilizar un prototipo tecnológico para la detección temprana.

La Ingeniera Agrónoma Elizabeth Males, cuenta con más de 5 años de experiencia en el sector agrícola. De las preguntas 2, 3 y 4, se identificó que la etapa de floración es cuando aparece con mayor frecuencia la *Botrytis cinerea* en las plantas de granadilla. Los factores ambientales determinantes incluyen condiciones invernales y variaciones de temperatura que alteran el ciclo de vida de patógenos. Para la detección temprana, recomienda una observación constante de los cultivos, especialmente al ingresar al campo, para aplicar fungicidas oportunamente.

En cuanto a las preguntas 5, 6 y 7, el profesional considera que la inspección visual tradicional, aunque útil, podría mejorarse significativamente. A pesar de no haber trabajado con herramientas tecnológicas para la detección de enfermedades, reconoce el potencial de los sistemas basados en visión artificial para mejorar la detección temprana de la enfermedad en comparación con los métodos actuales.

Las respuestas a las preguntas 9 y 10 demuestran una alta disposición para recomendar el uso de sistemas basados en visión artificial para la detección de *Botrytis cinerea*. Además, identifica otras enfermedades que podrían beneficiarse de esta tecnología, como *Fusarium*, *Phytophthora infestans* (lancha), *Mycena citricolor* (ojo de pollo), y plagas como ácaros y trips.

3.3 Requerimientos del sistema

A continuación, basándose en la fase de verificación y simulación de la metodología se presentan los requerimientos esenciales para el desarrollo del prototipo, incluyendo las necesidades del sistema, los requisitos técnicos y las especificaciones de la arquitectura. Esto abarca tanto los elementos de hardware como de software necesarios para implementar el sistema de detección temprana de *Botrytis cinerea* en las hojas de la granadilla.

3.3.1. Nomenclatura de Requerimientos

Para asegurar una correcta identificación y clasificación de los requerimientos dentro del proyecto, se definió un sistema de nomenclatura basado en abreviaturas específicas. Este sistema permite una referencia rápida y organizada, optimizando la comprensión y el manejo de la información. Además, facilita la trazabilidad de cada requerimiento a lo largo del desarrollo del proyecto. La relación detallada de los participantes involucrados en esta etapa se encuentra en la Tabla 4.

Tabla 4*Lista de participantes en el proyecto*

Nº	Actores involucrados	
1	Ing. Fabián Cuzme	Tutor de tesis
2	Ing. Luis Suárez	Asesor de tesis
3	Ermel Robles	Tesista
4	Marcia Robles	Agricultor
5	Ing. Elizabeth Males	Ingeniera agrónoma

Cada requerimiento se identifica mediante una abreviatura, lo que permite una gestión más eficiente y una mejor comprensión del documento. La relación completa de estas abreviaturas y su significado se encuentra detallada en la Tabla 5.

Tabla 5*Abreviatura de los requerimientos*

Nomenclatura	Descripción
StSR	Requerimientos Stakeholders
SySR	Requerimientos del Sistema
SrSR	Requerimientos de Hardware y Software

Las tablas de requerimientos están organizadas en tres secciones: la primera asigna un identificador único a cada requerimiento, la segunda describe cada uno de ellos de manera detallada, y la tercera evalúa la importancia de implementarlos en el sistema, categorizándolos en tres niveles: Alta, Media y Baja.

3.3.2. Requerimientos de Stakeholders

Los requerimientos de los stakeholders comprenden las demandas y expectativas de los clientes, usuarios u otros interesados que deben ser considerados para la integración en el sistema. A continuación, en la Tabla 6, se presentan los requisitos tanto de los usuarios como operacionales.

Tabla 6

Requerimientos de stakeholders

Requerimiento de stakeholders (StSR)					
#	Requerimiento	Prioridad			StSR
					relacionados
		Alta	Media	Baja	
Requerimientos operacionales					
StSR1	Dispositivo de captura de imágenes con una resolución mínima de 5 megapíxeles.	X			
StSR2	Garantía de parámetros ópticos mediante una estructura sólida que permita capturar imágenes nítidas y de alta calidad.		X		StSR1
StSR3	Intervalo de captura de datos, asegurando la adquisición de imágenes durante períodos con condiciones óptimas de iluminación para maximizar la calidad visual.	X			StSR1, StSR2
StSR4	Disponibilidad de una fuente de energía confiable y accesible, ya sea mediante		X		

	conexión eléctrica estable, baterías recargables o paneles solares.		
StSR5	Resguardo de imágenes mediante sistemas de almacenamiento en bases de datos locales o plataformas en la nube.	X	StSR1, StSR3
Requerimientos de usuarios			
StSR6	Creación de una interfaz gráfica intuitiva con navegación optimizada para máxima usabilidad y eficiencia.	X	
StSR7	Provisión de un acceso rápido, intuitivo y sin dificultades a la interfaz de usuario.	X	StSR6
StSR8	Los datos deben mostrarse de forma clara y estructurada para facilitar su comprensión.	X	StSR5, StSR6, StSR7

3.3.3. Requerimientos del Sistema

En esta sección se describen los requerimientos iniciales del sistema (SySR), que especifican las funciones, limitaciones y el comportamiento esperado del sistema. Se incluyen los requisitos relacionados con su uso, interfaces, modos de operación, estados y aspectos físicos. Los detalles completos de los requerimientos del sistema se muestran en la Tabla 7.

Tabla 7*Requerimientos del sistema*

Requerimiento del sistema (SySR)				SySR
relacionados				
#	Requerimiento	Prioridad		
		Alta	Media	Baja
Requerimientos de interfaz				
SySR1	Implementación de soluciones basadas en conectividad inalámbrica para optimizar la transmisión de datos y la flexibilidad del sistema.	X		
Requerimientos de uso				
SySR2	Plataforma de fácil manejo, diseñada para ofrecer una experiencia de usuario intuitiva y sin complicaciones.	X		SySR1
SySR3	Disponibilidad inmediata de los resultados y parámetros analíticos, permitiendo una consulta rápida y eficiente de la información.		X	SySR1, SySR2
Requerimientos performance				
SySR4	El sistema debe funcionar sin afectar el crecimiento natural del cultivo.	X		
SySR5	Consumo energético reducido, optimizando la eficiencia y		X	SySR5

	minimizando el impacto en el consumo eléctrico.		
SySR6	Almacenamiento y procesamiento de datos mediante plataformas en la nube, facilitando la gestión remota y el acceso eficiente a la información.	X	SySR1, SySR3
SyR7	Métodos especializados para el análisis y tratamiento de imágenes.	X	SySR6
SySR8	Representación visual de los resultados mediante gráficos.	X	SySR2, SySR3, SySR7
Requerimientos de modo y estado			
SySR9	La interfaz gráfica ofrece una visualización detallada de la clasificación de cada lote del cultivo, facilitando su análisis y seguimiento.	X	SySR2, SySR3, SySR7, SySR8
Requerimientos físicos			
SySR10	Las cámaras deben instalarse en áreas libres de obstrucciones, asegurando una visión clara y sin impedimentos para una captura de imágenes eficiente.	X	SySR4
SySR11	Las cámaras deben contar con una cobertura protectora adecuada para salvaguardar su funcionamiento	X	SySR4, SySR10

	frente a las condiciones ambientales exteriores.		
SySR12	El sistema debe estar equipado con una tarjeta de memoria de capacidad adecuada, garantizando suficiente espacio de almacenamiento para evitar la saturación.	X	SySR6

3.3.4. *Requerimientos de Arquitectura*

Los requerimientos de arquitectura son fundamentales para identificar los componentes de hardware y software necesarios para la correcta implementación y funcionamiento del sistema. En la Tabla 8 se detallan los requerimientos relacionados con el diseño lógico, hardware, software y aspectos eléctricos.

Tabla 8

Requerimientos de arquitectura

Requerimiento de Arquitectura (SrSH)						
#	Requerimiento	Prioridad			SrSH relacionados	
		Alta	Media	Baja		
Requerimientos de diseño						
SrSH1	El hardware y el software deben ser compatibles para garantizar un rendimiento óptimo.	X				

SrSH2	Conexión eficiente entre la plataforma de procesamiento de imágenes, la base de datos y la interfaz gráfica.	X	SrSH1
SrSH3	Sistema de fácil acceso y gestión para los administradores.	X	SrSH2
Requerimientos lógicos			
SrSH4	Conexión directa a la base de datos.	X	SrSH2
SrSH5	Soporte para sistemas operativos Linux.	X	SrSH5
SrSH6	Soporte para varios lenguajes de programación.	X	SrSH1, SrSH5
Requerimientos de hardware			
SrSH7	Placa embebida	X	SrSH1
SrSH8	Cámara para exteriores con resolución no inferior a 5 megapíxeles.	X	SrSH1, SrSH7
SrSH9	Conexión a la red de internet.	X	SrSH2, SrSH4
SrSH10	Memoria RAM no inferior a 2GB.	X	SrSH7
SrSH11	Puerto para tarjeta SD.	X	SrSH7
Requerimientos de software			
SrSH13	Compatibilidad entre bibliotecas de visión artificial, procesamiento de imágenes y aprendizaje automático.	X	SrSH1, SrSH5, SrSH6
SrSH14	Licencias de código abierto.	X	SrSH13
SrSH15	Lenguaje de programación de alto nivel.	X	SrSH6
SrSH16	Programa especializado en el procesamiento de imágenes.	X	SrSH13, SrSH15

SrSH17	Entorno de desarrollo integrado (IDE) para programación.	X	SrSH15
SrSH18	Soporte para diferentes lenguajes de programación.	X	SrSH6, SrSH15
SrSH19	Procesamiento gráfico basado en técnicas de visión artificial basado en GPU pero sin dependencia de este.	X	SrSH13, SrSH16
Requerimientos eléctricos			
SrSH20	Fuente de alimentación eléctrica adaptable a infraestructura con baterías.	X	SrSH7

3.3.5. Selección de Hardware

En este apartado se define el hardware más adecuado para el proyecto, considerando los requerimientos identificados en las etapas previas. La selección se realiza mediante un análisis comparativo, asignando una calificación binaria (1 si cumple con el requisito, 0 si no cumple). El componente con la mayor puntuación será elegido para garantizar su compatibilidad y efectividad en la implementación del sistema como se puede ver en la Tabla 9 y Tabla 10 respectivamente.

- **Placa embebida**

La implementación de una placa embebida es esencial para llevar a cabo operaciones relacionadas con visión artificial. Este dispositivo debe satisfacer los requerimientos técnicos del proyecto, por lo que se realiza una comparación detallada entre diversas opciones. Los resultados de esta evaluación se presentan en la tabla 9.

Selección de placa embebida

Selección de placa embebida

Figura 13

Placa Seleccionada Raspberry Pi4



Nota. (González García, 2015)

- **Cámara**

El desarrollo del proyecto requiere el uso de una cámara con amplia capacidad de enfoque, diseñada para condiciones de exteriores, y con una resolución adecuada que cumpla con los estándares necesarios para aplicaciones de visión artificial. Los resultados de esta evaluación se presentan en la tabla 10.

Tabla 10

Selección de cámaras

	REQUISITOS							Val.
	StS	StS	SyS	SyS	SrS	SrS	SrS	
	R1	R5	R11	R12	R8	R9	R10	
Ezviz C8C Lite	0	1	1	1	1	1	1	6
WiFi Smart Camera	1	1	1	1	1	1	1	7

Cámara Ptz Wifi, 1080p	0	1	1	1	1	1	1	6
Cámara Cw8 con Wifi	1	1	1	1	1	1	1	7
Cumple “1” No cumple “0”								

Elección: La cámara seleccionada para el proyecto es el modelo WiFi Smart Camera, diseñada para exteriores. Este dispositivo ofrece alta claridad de video, un campo de visión de 360 grados que permite cubrir áreas extensas, y la capacidad de moverse tanto horizontal como verticalmente. Además, cuenta con opciones para almacenar videos de forma local o en la nube, lo que mejora su versatilidad y funcionalidad.

Figura 14

Cámara Elegida Wi-Fi Smart Camera



Nota. (Mercado Libre Ecuador, n.d.)

3.3.6. Selección de Software

La selección del software como se indica en la Tabla 11 se basa en los requerimientos del sistema y la arquitectura, evaluando opciones mediante una tabla comparativa. Este análisis permite identificar el lenguaje de programación de alto nivel, la red de entrenamiento y la tecnología de transmisión que mejor satisfacen las necesidades del proyecto.

- **Frameworks de procesamiento de imágenes**

La selección del framework para procesamiento de imágenes se fundamenta en diversos requerimientos estratégicos. El requerimiento SrSH13 asegura la compatibilidad entre las bibliotecas de visión artificial, procesamiento de imágenes y aprendizaje automático; el requerimiento SrSH16 garantiza un programa especializado en el procesamiento de imágenes; el requerimiento SySR7 establece métodos especializados para el análisis y tratamiento de imágenes; y el requerimiento SySR8 facilita la representación visual de resultados mediante gráficos. En la tabla 12 se presenta una evaluación comparativa de distintos frameworks analizados en función de estos requerimientos.

Tabla 11

Selección de framework para procesamiento de imágenes

Framework		REQUISITOS				
		SrSH13	SrSH16	SySR7	SySR8	VALOR
Ultralytics YOLO		1	1	1	1	4
Roboflow		1	1	1	0	3
PyTorch Vision		1	0	1	1	3
OpenCV	DNN	1	1	0	0	2
Module						

Detectron2	1	1	1	0	3
Cumple “1” No cumple “0”					

Elección: En la tabla se presenta una evaluación comparativa de distintos frameworks, destacando Ultralytics YOLO como la opción óptima al satisfacer todos los criterios establecidos. Este framework proporciona capacidades superiores de detección de objetos, procesamiento eficiente y visualización de resultados, ofreciendo la solución más completa para las necesidades del sistema de captura y análisis de imágenes del cultivo.

Figura 15

Framework Seleccionado Ultralytics YOLO



Nota. (Vina, 2024)

- **Lenguaje de programación de alto nivel**

Tabla 12

Selección de lenguaje de programación

Lenguaje de programación	REQUISITOS				
	SrSH 11	SrSH 12	SrSH 13	SrSH 15	VALOR
C++	1/2	1	0	1	2.5
Python	1	1	1	1	4
Java Script	1	1	1	1	4

Java	1	1	0	1	3
Cumple “1” No cumple “0”					
Elección: El lenguaje de programación seleccionado es Python, ya que es ampliamente utilizado en aplicaciones de visión artificial. Además, ofrece soporte para la corrección de errores mediante diversos entornos de desarrollo (IDEs) y cuenta con una extensa colección de bibliotecas, como NumPy, OpenCV, SciPy y PyBrain, que facilitan el procesamiento de imágenes y el aprendizaje automático.					

Figura 16

Lenguaje de Programación Seleccionado Python



Nota. (Ilustre Colegio Oficial de Geólogos, 2025)

- **Selección de entorno de desarrollo**

REQUISITOS								
Entorno	de	SrSHS	SrSHS	SrSHS	SrSHS	SrSHS	SrSHS	VALOR
desarrollo		13	14	15	16	17	18	
Google Colab		1	1	1	1	1	1	6
Jupyter Notebook		1	1	1	0	1	1	5
Thonny IDE		1	1	1	0	1	1	5

Visual Studio	1	0	1	1	1	1	5
Code							
Elección: Google Colab ha sido seleccionado como entorno de desarrollo por su compatibilidad con bibliotecas de visión artificial, soporte para múltiples lenguajes de programación, entorno especializado en procesamiento de imágenes, disponibilidad como entorno de desarrollo integrado (IDE), y su enfoque colaborativo en la nube, facilitando el trabajo distribuido.							

Figura 17

Entorno de Desarrollo Seleccionado Google Colab



Nota. (Mohammed, 2021)

- Selección de plataforma en la nube

NUBE	REQUISITOS							VALOR
	SySRS	SySRS	SySRS	SrSHS	SrSHS	SrSHS	SrSHS	
	3	6	12	4	17	9	13	
Firestore	1	1	1	1	1	1	1	6
AWS	1	1	0	1	1	1	1	5
Azure	1	1	0	1	1	1	1	5
Dropbox	0	1	0	0	1	0	0	2

Elección: Se selecciona Firebase como plataforma en la nube por ofrecer conectividad en tiempo real, capacidad de almacenamiento y procesamiento de datos, representación visual clara, soporte para bases de datos y compatibilidad con bibliotecas de visión artificial.

Figura 18
Plataforma Nube Seleccionada Firebase



Nota. (Okooone Ltd., 2025)

- Selección de proveedor de servicios en la nube

		REQUISITOS						
Proveedor	de	SySRS	SySRS	SrSHS	SrSHS	SrSHS	SrSHS	VALOR
servicios	en la	1	3	6	13	15	19	
nube								
Amazon	Web	1	1	1	1	1	0	5
Services	AWS							
Google	Cloud	1	1	1	1	1	1	6

Microsoft	1	1	0	1	1	0	5
Azure							
IBM Cloud	1	0	1	1	1	0	4
Elección: Google Cloud fue la opción seleccionada por cumplir plenamente con los requerimientos de conectividad inalámbrica, disponibilidad inmediata de datos, procesamiento en la nube, compatibilidad con bibliotecas de inteligencia artificial, lenguaje de alto nivel, y procesamiento gráfico avanzado sin dependencia exclusiva de GPU.							

Figura 19

Servicio en la Nube Seleccionado Google Cloud



Nota. (Ausum Cloud, 2024)

- **Tecnología de transmisión inalámbrica**
La selección de la tecnología de transmisión inalámbrica se basa en varios requerimientos clave. El requerimiento SySR1 asegura la conectividad con tecnologías de comunicación inalámbrica; el requerimiento SrSH2 garantiza la comunicación eficiente con la plataforma de procesamiento de imágenes y la interfaz gráfica; y el requerimiento SrSH5 establece la conexión entre la base de datos y la interfaz gráfica para la visualización de

resultados. En la tabla 13 se presenta una comparación detallada de distintas tecnologías inalámbricas evaluadas en función de estos requerimientos.

Tabla 13

Selección de tecnología inalámbrica

Plataforma IoT	REQUISITOS			
	SySR 1	SrSH 2	SrSH 5	VALOR
Wi-Fi (802.11 n)	1	1	1	3
LoraWAN	1	0	1	2
GSM	1	0	1	2
Cumple “1” No cumple “0”				
Elección: La tecnología de transmisión inalámbrica seleccionada es Wi-Fi, ya que ofrece la capacidad de establecer un nodo de comunicación como punto de acceso, proporcionando conectividad estable para la transmisión de datos.				

Figura 20

Transmisión Inalámbrica Seleccionada WiFi



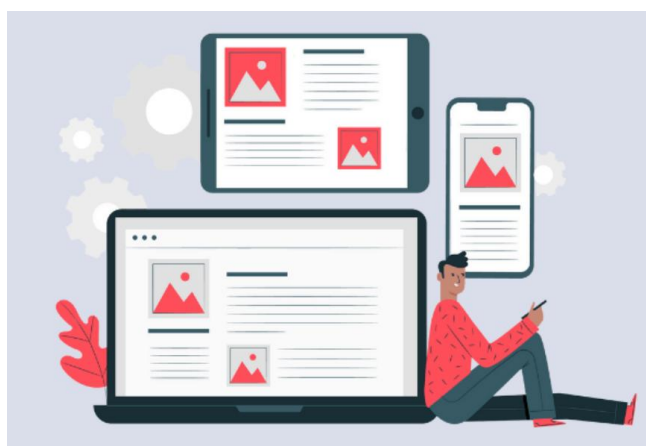
Nota (Crespo Moreno, 2017)

- Selección del medio de acceso

		REQUISITOS						
Medio de acceso		SySRS	SySRS	SySRS	SySRS	SrSHS	SrSHS	VALOR
		2	3	6	8	15	19	
Plataforma web		1	1	1	1	1	1	6
Aplicación móvil		1	1	1	1	1	0	5
SMS (mensajes texto)		1	1	0	1	1	0	5
Elección: El medio de acceso seleccionado fue la plataforma web que es una forma sencilla en la cual se puede evidenciar los datos obtenidos o divulgados en cualquier dispositivo que obtenga una conexión inalámbrica WiFi y poder hacerlo de forma remota.								

Figura 21

Medio de Acceso Seleccionado Plataforma Web



Nota. (Facchin, 2022)

3.4 Diseño del sistema

Para tener concordancia con la metodología propuesta durante la fase de integración de bloques; el sistema integral para la detección temprana de la enfermedad *Botrytis cinerea* en hojas de granadilla se compone de tres módulos principales: Captura y Procesamiento, Red, y Distribución. Cada uno de estos módulos cumple un papel fundamental en la arquitectura del prototipo electrónico basado en visión artificial.

El módulo de Captura y Procesamiento constituye el primer eslabón en la cadena de detección. Utiliza una cámara IP conectada mediante WiFi que captura imágenes de alta resolución de las hojas de granadilla en el campo. Estas imágenes son transmitidas a una Raspberry Pi, que funciona como unidad de procesamiento principal. La Raspberry Pi ejecuta algoritmos desarrollados en Python que realizan un análisis preliminar de las imágenes capturadas. Para la detección de patrones asociados con *Botrytis cinerea*, se implementa el framework Ultralytics YOLO que permite identificar y segmentar las áreas potencialmente afectadas por la enfermedad en tiempo real. Adicionalmente, se utiliza el protocolo RTSP para garantizar una transmisión fluida y continua del video desde la cámara IP hacia la unidad de procesamiento.

El módulo de Red es responsable de la comunicación y el almacenamiento de datos generados por el sistema. Un router funciona como gateway, proporcionando conectividad a internet y estableciendo el enlace entre el módulo de captura y la infraestructura en la nube. La tecnología Websocket permite una comunicación bidireccional en tiempo real entre el sistema y las aplicaciones cliente, facilitando actualizaciones instantáneas sobre el estado de los cultivos. La infraestructura cloud, específicamente Google Cloud Platform, alberga servicios avanzados de procesamiento que ejecutan modelos de inteligencia artificial más robustos, capaces de realizar análisis más complejos que los ejecutados localmente en la Raspberry Pi.

Complementariamente, Cloud Firestore actúa como base de datos NoSQL en tiempo real, almacenando información sobre las detecciones realizadas y permitiendo un seguimiento histórico de la enfermedad en los cultivos.

El módulo de Distribución se encarga de presentar los resultados del análisis a los usuarios finales de manera accesible y comprensible. Utiliza el framework Ionic para el desarrollo de aplicaciones móviles multiplataforma, creando interfaces de usuario intuitivas y funcionales. Cordova permite convertir el desarrollo web en aplicaciones móviles nativas, optimizadas principalmente para el sistema operativo Android. Estas aplicaciones permiten a los agricultores visualizar el estado actual de sus cultivos, recibir alertas tempranas sobre detecciones de Botrytis cinerea, y acceder a estadísticas e históricos de datos. La interfaz web complementa la versión móvil, ofreciendo funcionalidades adicionales para análisis más detallados.

El flujo de funcionamiento del sistema inicia con la captura de imágenes de las hojas de granadilla mediante la cámara IP. Estas imágenes son transmitidas a la Raspberry Pi, donde se ejecutan algoritmos de procesamiento inicial para detectar posibles signos de Botrytis cinerea. Los resultados de este análisis preliminar se envían a través del router a la infraestructura cloud, donde se ejecutan algoritmos más sofisticados de inteligencia artificial que confirman o refinan las detecciones. Los resultados finales se almacenan en Cloud Firestore y se transmiten mediante Websockets a las aplicaciones cliente. Finalmente, los usuarios reciben notificaciones y pueden visualizar los resultados a través de aplicaciones desarrolladas con Ionic/Cordova, disponibles para dispositivos Android o mediante interfaz web.

Este sistema integral permite la detección temprana y precisa de Botrytis cinerea en cultivos de granadilla, ofreciendo a los agricultores la posibilidad de implementar medidas preventivas antes de que la enfermedad se propague. Esto resulta en una reducción significativa del uso de fungicidas y minimiza las pérdidas económicas asociadas con la enfermedad,

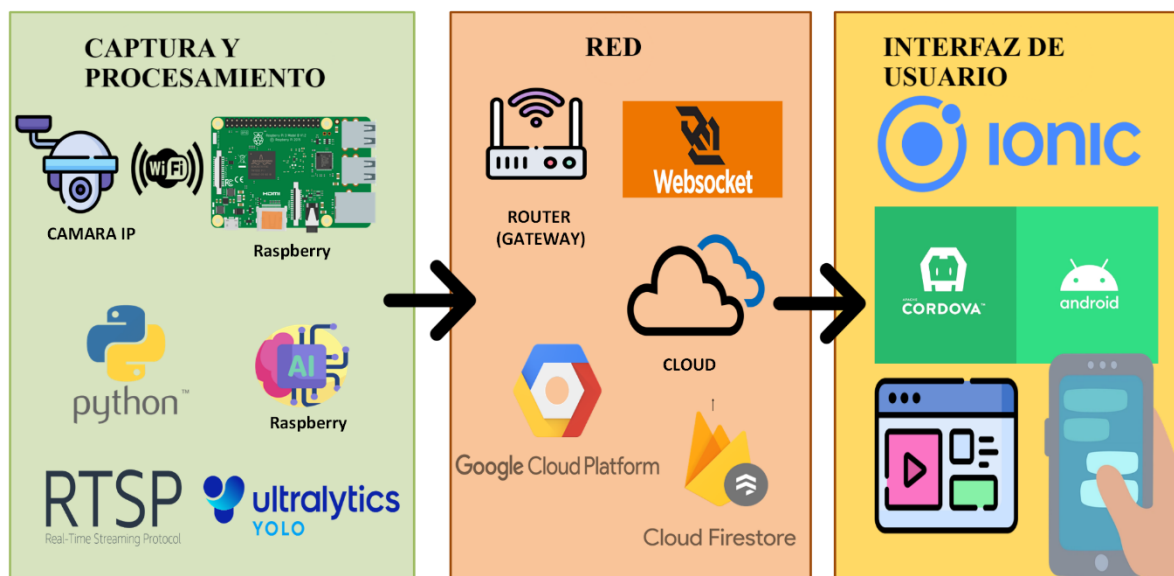
contribuyendo a prácticas agrícolas más sostenibles y eficientes, alineadas con los objetivos de la investigación.

3.4.1. Diagrama General de Bloques del Sistema

La Figura 22 presenta un diagrama general que ilustra los procesos principales del sistema, proporcionando una visión clara de su funcionamiento integral. Más adelante, se analizará en detalle cada uno de los bloques que lo componen.

Figura 22

Diagrama de bloques general



Nota. Autoría Propia

3.4.2 Diagrama de Flujo

En esta sección se detallan los diagramas de flujo correspondientes a los procesos clave del sistema: preprocesamiento de las imágenes, entrenamiento de la red neuronal y clasificación. Cada diagrama representa de manera estructurada las etapas y decisiones involucradas en dichos procesos.

3.4.2.1. Flujograma del Preprocesamiento de Imágenes

Las rotaciones de imágenes constituyen una técnica esencial de aumento de datos que potencia significativamente los modelos de aprendizaje profundo al desarrollar invariancia rotacional. Esta propiedad crítica permite que los algoritmos identifiquen con precisión patógenos como la *Botrytis cinerea* independientemente de la orientación en que se presenten en las hojas de granadilla, replicando así la diversidad natural de orientaciones foliares que se encuentran en condiciones reales de cultivo.

La visión artificial aplicada a fitopatología ha demostrado consistentemente que los conjuntos de datos enriquecidos mediante rotaciones sistemáticas mejoran sustancialmente tanto la precisión como la robustez de los modelos de detección temprana. Esta metodología permite que las redes neuronales convolucionales (CNN) aprendan características diagnósticas invariantes a la orientación, un factor determinante cuando se busca desarrollar sistemas de diagnóstico aplicables en campo. (Quiroga, 2023)

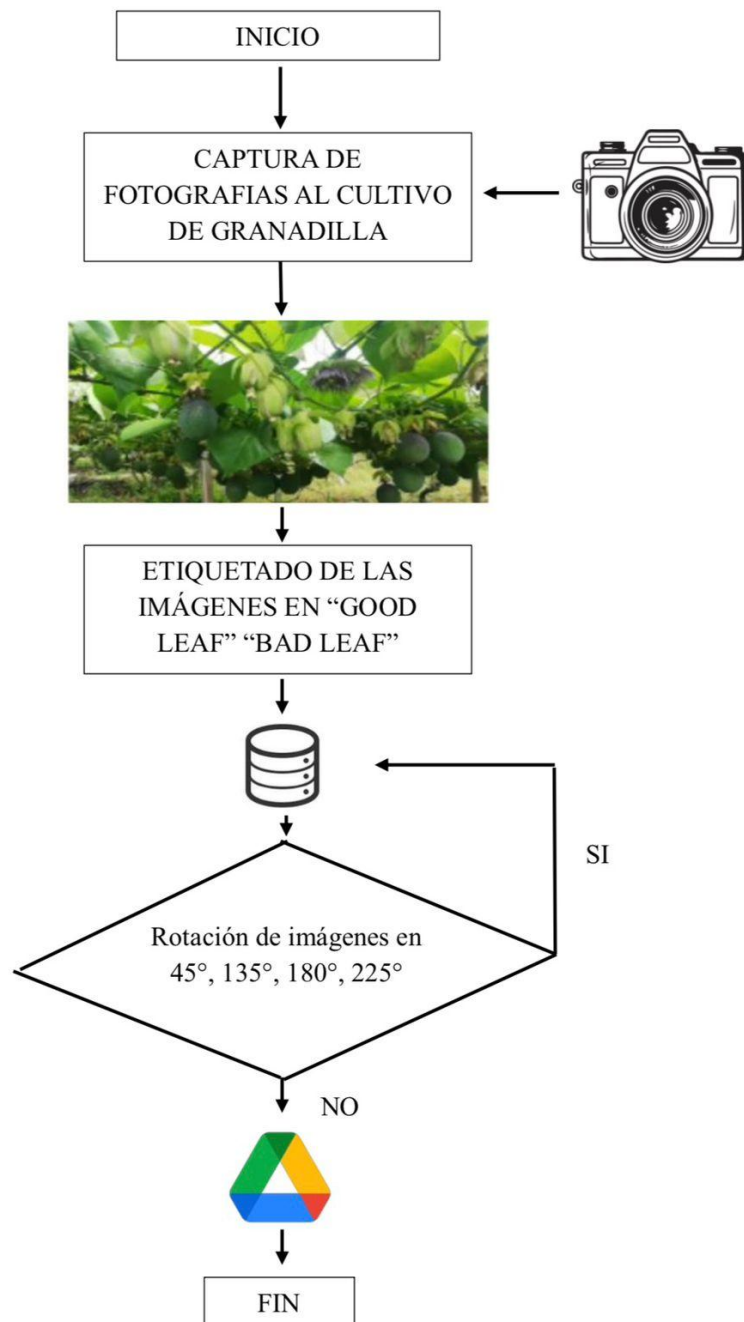
En el contexto específico de esta investigación sobre la detección temprana de *Botrytis cinerea* en cultivos de granadilla, la implementación estratégica de rotaciones en ángulos precisos de 45°, 135°, 180° y 225° ha permitido expandir considerablemente la diversidad y representatividad del conjunto de datos original. Este enfoque metodológico no solo compensa efectivamente las limitaciones inherentes a la disponibilidad reducida de muestras iniciales, sino que además fortalece la capacidad del modelo para generalizar el reconocimiento de síntomas patológicos en múltiples orientaciones. (Quiroga, 2023)

El diagrama de la figura 23 inicializa el proceso, adquiere las imágenes de los lotes mediante toma de foto con la cámara y se guardan en Google Drive, estas mismas pasan por un proceso de extracción de las imágenes las cuales serán delimitadas y tendrán un etiquetado entre buenas y malas para la implementación; a continuación antes del entrenamiento estas imágenes tendrán una rotación de entre 45° a 225° para obtener un dataset más robusto, dentro

de este dataset se obtendrá y dividirá en un 70% de imágenes para el entrenamiento de la red neuronal, un 15% para el testeo de la misma y el 15% restante para la validación de los datos, por ultimo estos mismo se guardan en la nube de Google Drive para ser almacenadas.

Figura 23

Flujograma de preprocesamiento de imágenes



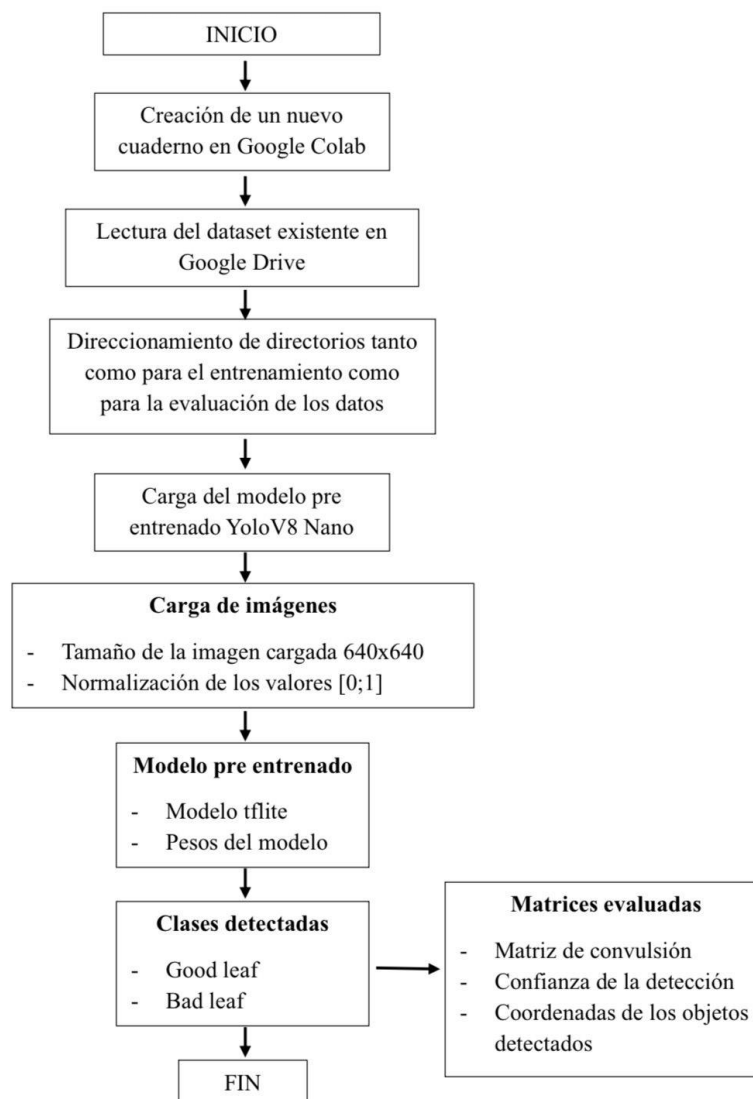
Nota. Autoría Propia

3.4.2.2. Flujograma de Entrenamiento de la Red

Para el entrenamiento de la red neuronal se utilizará el servidor “Colab” perteneciente a Google, en donde se realizan los procesos de la convolución para el uso de la red neuronal yolov8n, para estos mismos se clasificarán en dos tipos de hojas; “Good leaf” y “Bad leaf”, consiguiendo así un modelo preentrenado que es a lo que se quiere llegar, en la Figura 24 se puede observar el diagrama de flujo del entrenamiento de la red neuronal.

Figura 24

Diagrama de flujo del entrenamiento de la red neuronal



Nota. Autoría Propia

3.4.2.3 Flujograma de Integración del Sistema

Después del modelo preentrenado, se lo integra al sistema embebido Raspberry Pi4, en el cual se van a importar todos los archivos del modelo preentrenado que contiene el mapa de caracteres con etiquetas conocidas como labels y con las configuraciones de las capas de entrenamiento

Se importan las librerías dentro de la Raspberry las cuales serán necesarias para trabajar con las aplicaciones de la visión artificial, estas librerías se detallan en la Tabla 14 con cada una de sus funcionalidades.

En la Figura 25 se presenta el siguiente diagrama de flujo con el cual se puede ver el proceso dispuesto antes, lo cual facilitara la comprensión de todas las etapas a realizar

Tabla 14

Librerías usadas

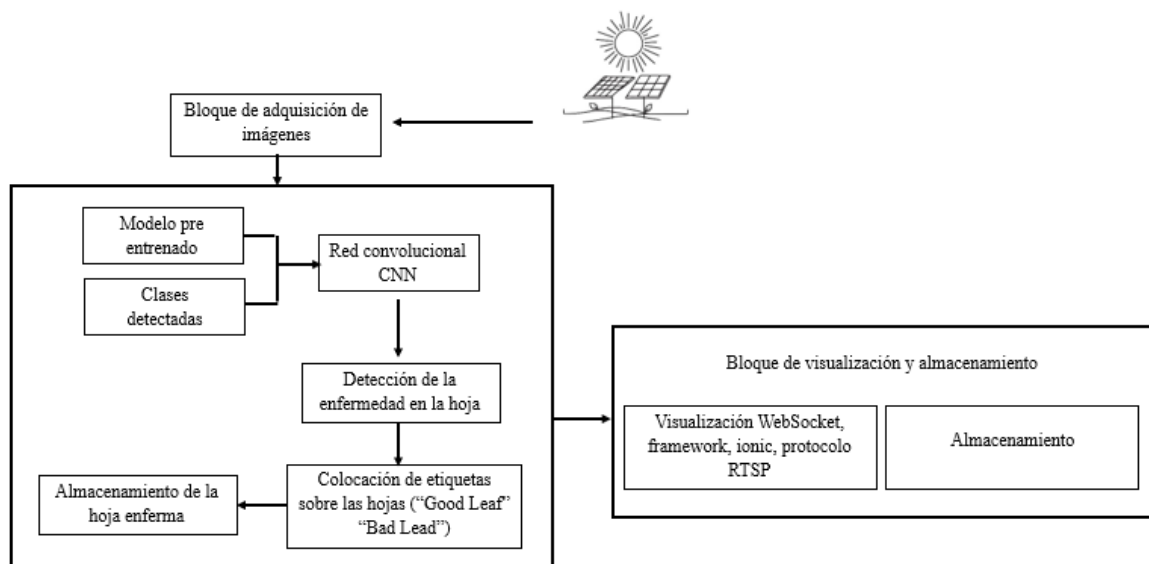
Librerías usadas	
Nombre	Funcionalidad
Os	Este módulo proporciona acceso a diversas funcionalidades del sistema operativo, ofreciendo información sobre el entorno y facilitando la gestión de directorios, así como la lectura y escritura de archivos.
Random	La librería Random de Python se utiliza para generar números aleatorios y realizar operaciones como seleccionar elementos al azar de una lista, mezclar secuencias y generar distribuciones estadísticas, lo que

	resulta útil en simulaciones y aplicaciones que requieren imprevisibilidad.
Shutil	Proporciona funciones para manipular archivos y directorios, como copiar, mover, renombrar y eliminar archivos, así como trabajar con archivos comprimidos. Es útil para gestionar y automatizar tareas relacionadas con el sistema de archivos.
Torch	Es una librería de aprendizaje profundo en Python que ofrece una amplia variedad de herramientas para la construcción y entrenamiento de redes neuronales. Está diseñada para ser flexible y eficiente, permitiendo a los desarrolladores implementar modelos de inteligencia artificial y procesamiento de datos complejos de manera rápida y escalable.
Ultralytics	Librería de visión artificial que facilita la implementación y entrenamiento de modelos de detección de objetos utilizando redes neuronales profundas. Es conocida por su facilidad de uso, optimización de modelos y su integración con el framework YOLO (You Only Look Once), permitiendo a los usuarios

	crear aplicaciones de visión por computadora de alta precisión y eficiencia.
Cv2 (OpenCV)	Es una librería especializada en visión artificial, que facilita el desarrollo de aplicaciones para el procesamiento y análisis de imágenes.
Matplotlib	Facilita la creación de gráficos bidimensionales personalizados, como diagramas de barras, utilizando listas o arreglos como datos de entrada.

Figura 25

Flujograma de integración del sistema



Nota. Autoría Propia

3.4.3 Descripción de escenarios

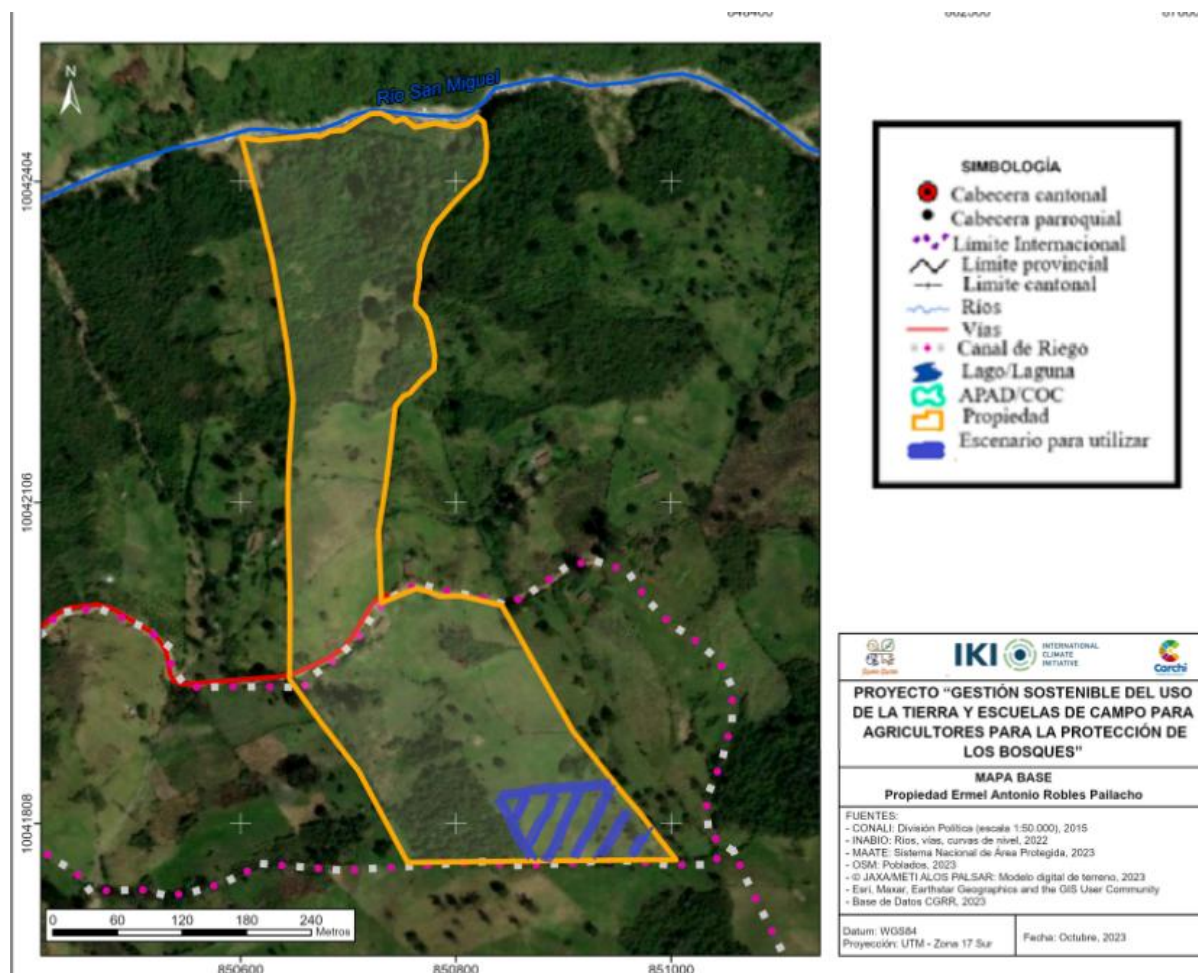
Antes de llevar a cabo la implementación del "Prototipo electrónico basado en visión artificial para la detección temprana de la enfermedad *Botrytis cinerea* en hojas de granadilla(*Passiflora ligularis*)" se han diseñado escenarios de prueba que se muestran en la Figura 26 para evaluar su desempeño en un entorno de producción controlado. Este proceso tiene como objetivo verificar la efectividad del sistema en la detección de lotes afectados y asegurar su funcionalidad.

Para garantizar una detección temprana y precisa de *Botrytis cinerea* en hojas de granadilla (*Passiflora ligularis*), es esencial contar con condiciones controladas que permitan evaluar la efectividad del prototipo electrónico basado en visión artificial. Los escenarios como se muestra en la figura 26 definidos en este proyecto están diseñados para diferenciar hojas afectadas y no afectadas, asegurando un análisis detallado y eficiente que contribuya al manejo oportuno de esta enfermedad.

La parcela fue dividida en cuatro partes iguales con el objetivo de facilitar un manejo más ordenado y eficiente del área de estudio. Esta división estratégica permitirá, en el desarrollo futuro de la tesis, realizar un análisis más detallado y comparativo de los resultados obtenidos en cada sección. Además, esta estructura favorece la implementación de distintos tratamientos o condiciones experimentales en cada subparcela, lo cual enriquecerá la interpretación de los datos y contribuirá a la solidez de las conclusiones. Esta metodología también facilitará el monitoreo continuo de las variables de interés y garantizará una mejor distribución del trabajo durante todo el proceso investigativo.

Figura 26

División de la parcela



Nota. (IKI (International Climate Initiative), 2023)

El cultivo de granadilla no demanda cuidados extremadamente rigurosos, ya que se adapta adecuadamente a diversos tipos de suelos y climas. Sin embargo, su desarrollo óptimo ocurre en temperaturas moderadas, entre 15 y 18°C. Para maximizar su crecimiento y resistencia a enfermedades como *Botrytis cinerea*, se recomienda utilizar suelos ricos en nutrientes, bien drenados, libres de malezas y partículas como piedras o rocas.

En cuanto a la humedad del cultivo, esta planta no tolera períodos prolongados de sequía y es particularmente sensible a la insuficiencia de humedad. Para evitar el estrés en las raíces, es esencial mantener un nivel moderado y constante de humedad en el suelo.

El manejo adecuado del riego en el cultivo de granadilla es fundamental para garantizar y preservar la calidad de la producción como se ve en la figura 27. En este caso, se emplea exclusivamente el método de riego por aspersión, el cual permite distribuir el agua uniformemente sobre las plantas y el suelo. Este método asegura un nivel de humedad adecuado para el desarrollo del cultivo, contribuyendo a minimizar el estrés hídrico y favoreciendo condiciones óptimas para la implementación del sistema de visión artificial.

Figura 27

Escenarios de prueba



a) Escenario No apto



b) Escenario Apto

3.5 Diseño del hardware

En el presente proyecto se desarrolló el modelado, diseño e implementación de un adaptador de voltaje, utilizando software de simulación como Proteus para el esquema electrónico y la visualización en 3D del sistema. Se diseñó un circuito con regulación de voltaje tipo Buck Step-Down que permite reducir una entrada de 12 V a 5 V, adecuada para alimentar dispositivos como la Raspberry Pi, y se incorporó un transistor MOSFET controlado desde esta placa base para gestionar el encendido y apagado de la carga. Tras validar su funcionamiento, se procedió al diseño del PCB (Layout), integrando de forma ordenada todos los componentes pasivos y activos, con el fin de asegurar eficiencia y confiabilidad en la distribución de energía.

Posteriormente, se construyó un prototipo físico montado en un case impreso en 3D, el cual aloja la Raspberry Pi 4 Model B+, el módulo ATtiny85 como regulador de señal, sensores de control, conectores de entrada/salida y un panel solar de 12 V–200 mA que permite mantener la carga de la batería tipo TYX9-BS mediante carga de mantenimiento. El sistema incluye una cámara IP YOOSEE 360°, que recibe alimentación controlada mediante el adaptador de voltaje y se comunica vía Wi-Fi para monitoreo remoto, optimizando el consumo energético al habilitar su operación solo bajo condiciones específicas mediante control lógico desde la Raspberry Pi.

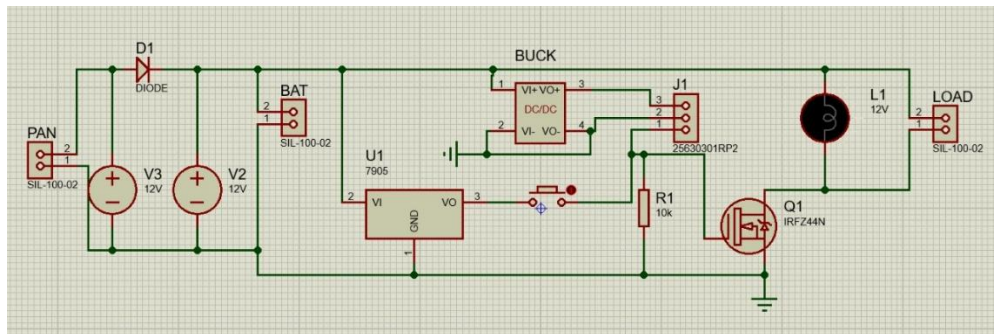
3.5.1 Modelado del Adaptador de Voltaje en Proteus y en 3D

En la Figura 28 se muestran los componentes necesarios para el desarrollo de la placa encargada de la distribución de voltajes hacia los distintos elementos del sistema. Las fuentes de voltaje representadas en el diagrama simulan la alimentación del prototipo, mientras que el regulador de voltaje tipo step-down (Buck) convierte los 12V de entrada a 5V de salida, necesarios para ciertos componentes como la Raspberry Pi.

El circuito también incluye una señal que simula una activación desde la Raspberry Pi, la cual controla un transistor MOSFET, permitiendo el paso de corriente hacia la carga. Además, se incorpora una señal que permite activar o desactivar la carga, emulando el funcionamiento del sistema bajo distintas condiciones de operación.

Figura 28

Modelado de control esquemático de voltajes.



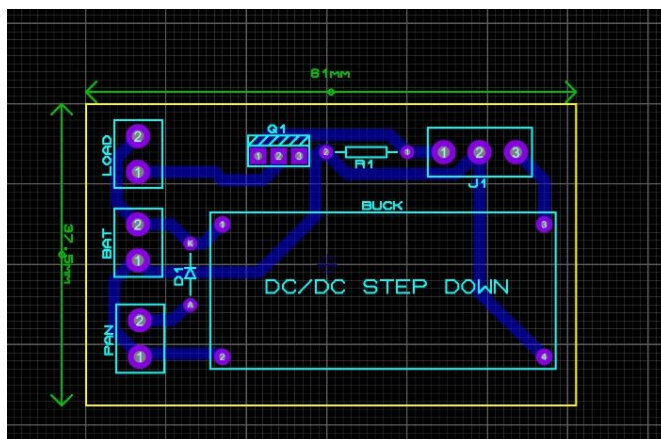
Nota. Autoría Propia

3.5.1.1 Diseño de circuito PCB Layout

Una vez finalizado el diseño y la simulación del circuito, se procede a la implementación de los componentes pasivos y activos en el diseño del PCB (*Layaout*). Esta etapa permite integrar de manera ordenada y funcional todos los elementos electrónicos, asegurando su correcta ubicación y conexión. Además, el uso de la interfaz gráfica del software de diseño facilita la visualización y verificación estructural del circuito, lo cual es fundamental antes de proceder a la fabricación de la placa como lo muestra la figura 29.

Figura 29

Diseño de circuito PCB Layaout



Nota. Autoría Propia

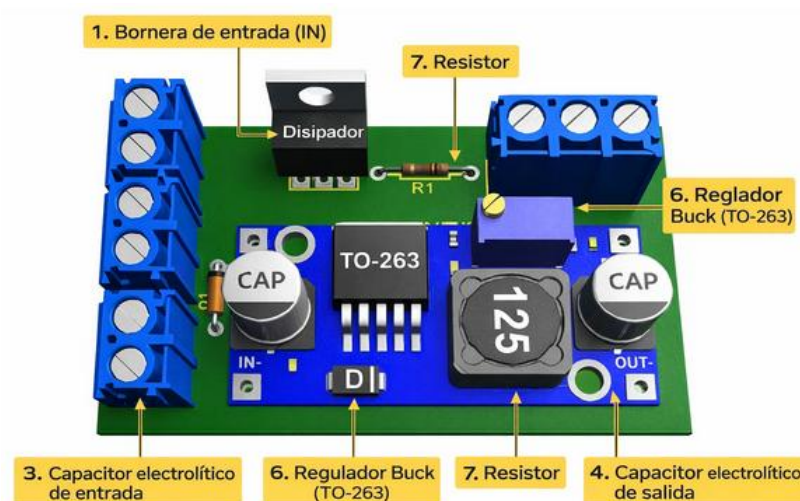
3.5.1.2 Adaptador de Voltaje

El sistema cuenta con un transistor MOSFET tipo NPN, que permite controlar o modificar la salida de la cámara de forma eficiente. Se incorpora además un convertidor Buck Step-Down, el cual reduce el voltaje de entrada de 12V a una salida de 5V, utilizada para la alimentación de la Raspberry Pi.

También se incluye un diodo de protección, cuya función es evitar posibles picos de voltaje inverso que puedan dañar los componentes. El prototipo presenta tres conectores: uno destinado a la conexión de una batería, otro para la salida de alimentación hacia la cámara, y un tercero para la entrada de voltaje externa, utilizada en caso de que la batería no esté disponible o no se encuentre cargada como se muestra en la figura 30.

Figura 30

Adaptador de voltaje



Nota. Autoría Propia

3.5.2 *Diseño de la Estructura Externa*

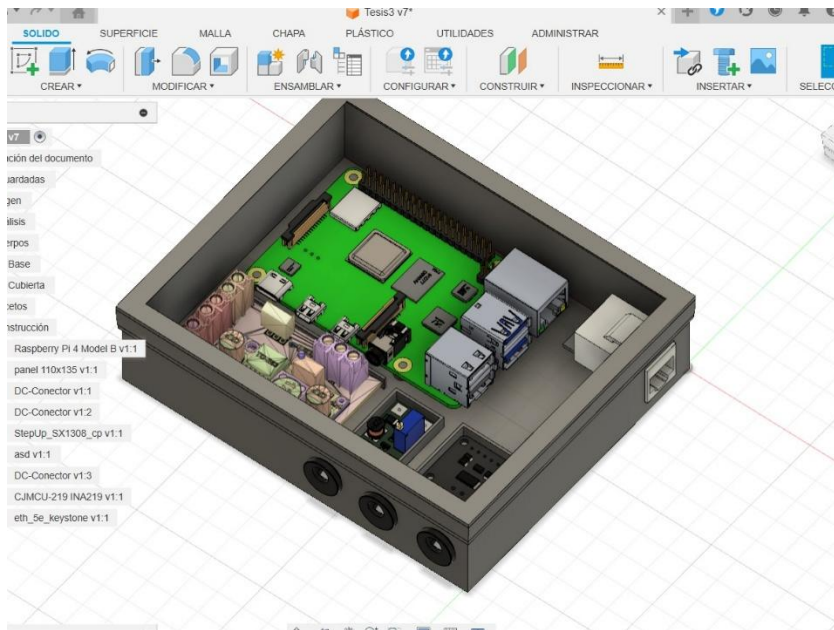
La Raspberry Pi implementada en este proyecto requiere una alimentación estable de 5 voltios con una corriente de hasta 3 amperios, valor que puede variar ligeramente dependiendo de la carga de trabajo y los periféricos conectados. Esta demanda energética se encuentra delimitada de acuerdo con las funciones específicas que ejecuta la Raspberry dentro del sistema. Para lograr esta adaptación de voltaje y corriente, se incorpora una pequeña placa basada en el microcontrolador ATtiny85, cuya función principal es permitir una regulación eficiente y precisa de las señales provenientes de la batería, adaptándolas a los niveles que requiere la Raspberry Pi para su correcto funcionamiento.

Adicionalmente, el sistema cuenta con una cámara de videovigilancia que opera a un voltaje nominal de 12 voltios. Esta cámara presenta un consumo promedio de 2 amperios durante su funcionamiento normal, aunque puede experimentar picos de corriente de hasta 5 amperios, especialmente durante procesos de inicialización o cambios bruscos en la iluminación, como la activación de infrarrojos o el cambio de resolución como lo muestra la Figura 31 en donde se evidencia el case del prototipo.

Se diseñó una carcasa rectangular con dimensiones externas aproximadas de 137.4 mm de largo, 112.4 mm de ancho y 35.7 mm de alto. Las paredes presentan un grosor de aproximadamente 2.6 mm, lo cual proporciona rigidez estructural adecuada para alojar componentes electrónicos. En una de las paredes laterales se incorporaron tres orificios circulares, alineados horizontalmente, que permiten la conexión de cables o la instalación de componentes como conectores o sensores. En el interior de la base se diseñaron estructuras de soporte rectangulares y cilíndricas, posiblemente destinadas a fijar placas electrónicas, módulos o elementos específicos del sistema. El diseño también incluye una perforación lateral adicional, probablemente destinada al paso de tornillos o indicadores LED.

Figura 31

Case del Prototipo Diseñado



Nota. Autoría Propia

En la figura 32 se muestra el montaje físico del sistema dentro del case impreso en 3D. En la parte superior se encuentra la Raspberry Pi 4 Model B+, encargada del procesamiento principal. En la parte inferior izquierda se observa el módulo Buck Step-Down que reduce el voltaje de entrada de 12V a 5V para alimentar la Raspberry. A su lado, se encuentra un pequeño circuito de control con ATtiny85, responsable de regular señales de activación. También se incluyen sensores y puertos accesibles montados en el frontal de la caja, los cuales permiten la interacción externa con el sistema.

Figura 32

Case del Prototipo Diseñado Impreso



Nota. Autoría Propia

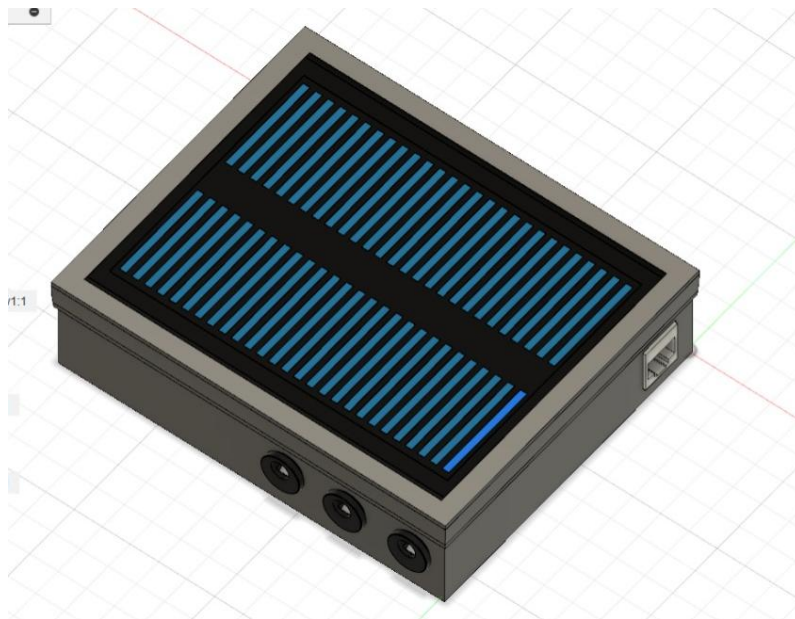
3.5.2.1 Implementación del Case con Panel Solar

El prototipo desarrollado incorpora un panel fotovoltaico con una tensión nominal de 12 voltios y una corriente máxima de 200 miliamperios, lo que le permite entregar una potencia pico de hasta 2.4 vatios bajo condiciones estándar de prueba.

Aunque la corriente que proporciona el panel puede parecer limitada en comparación con la demanda total del sistema, su función dentro del diseño es fundamental. En primer lugar, permite mantener la carga de la batería mediante el proceso conocido como “trickle charge” o carga de mantenimiento. La batería utilizada en el sistema, una TYX9-BS de 12 voltios y 9 amperios-hora, pierde entre un 2 % y un 4 % de su capacidad mensualmente debido a la autodescarga natural. La energía entregada por el panel solar es suficiente para compensar esta pérdida cuando el sistema está en estado de reposo, lo que contribuye a mantener la batería en niveles óptimos sin necesidad de conectarla constantemente a una fuente externa; esta función se la puede ver en la figura 33.

Figura 33

Case con Panel Solar



Nota. Autoría Propia

Como parte fundamental del sistema de monitoreo desarrollado, se implementó una cámara IP del tipo YOOSEE 360° como se muestra en la figura 34, la cual ofrece funciones avanzadas de videovigilancia, conectividad inalámbrica y rotación panorámica. Esta cámara fue seleccionada por su capacidad de cubrir un ángulo de visión amplio mediante rotación horizontal de 360 grados y vertical de hasta 90 grados, lo que permite monitorear entornos en tiempo real con una sola unidad, eliminando la necesidad de múltiples cámaras fijas.

La conexión física de la cámara se realiza a través de un canal de alimentación exclusivo integrado en la placa de distribución de voltaje del sistema. Dicha placa, diseñada a medida, incluye un transistor MOSFET tipo NPN que permite activar o desactivar el suministro de energía hacia la cámara mediante una señal lógica controlada desde la Raspberry Pi. Este mecanismo facilita una gestión inteligente de la energía, permitiendo, por ejemplo, apagar la

cámara en condiciones de bajo nivel de batería o durante horarios predefinidos para conservar recursos.

A nivel funcional, la cámara YOOSEE 360° se comunica mediante conexión Wi-Fi, lo cual permite visualizar en tiempo real la imagen capturada desde una aplicación móvil o desde un entorno web vinculado al sistema. Esta característica es clave para la supervisión remota del prototipo, especialmente en contextos donde el monitoreo en sitio no es posible. Se verificó también su compatibilidad con redes Wi-Fi estándar de 2.4 GHz, y se configuraron parámetros de seguridad como el cifrado WPA2 para asegurar la transmisión de datos.

Figura 34

Cámara Incorporada al Case



Nota. Autoría Propia

3.6 Desarrollo del software

Para esta sección se describe las siguientes etapas que van a ser parte del desarrollo del software para el sistema de visión artificial para la detección temprana de la Botrytis cinerea en hojas de la granadilla. En este procedimiento se comienza con el análisis de las imágenes que fueron tomadas enfocándose en tener capturas individuales y colectivas de cada una de las hojas. Se realiza un primer entrenamiento con una base de datos inicial en la que se obtiene

376 imágenes para entrenamiento y 95 imágenes para validación dando un total de 471 imágenes en la base de datos.

Posteriormente, se realiza un mejor robustecimiento de las imágenes mediante algunas transformaciones geométricas, como es la rotación en 45° , 135° , 180° y 225° dando un mejor robustecimiento de la base de datos. A continuación, las imágenes se procesan para obtener una separación entre hojas buenas y hojas malas asignándoles una etiqueta a estas mismas para el entrenamiento de la red neuronal. Después, se lleva a cabo el entrenamiento de esta red neuronal en el cual se obtiene un modelo preentrenado, este mismo se lo integra en la Raspberry Pi 4. Por último, mediante la ejecución de un script, el sistema comienza integrando la cámara en la página y después se puede comenzar con la ejecución de la detección, diferenciando entre hojas buenas y hojas malas con la condición de la enfermedad Botrytis cinerea.

3.6.1 Preprocesamiento de Imágenes

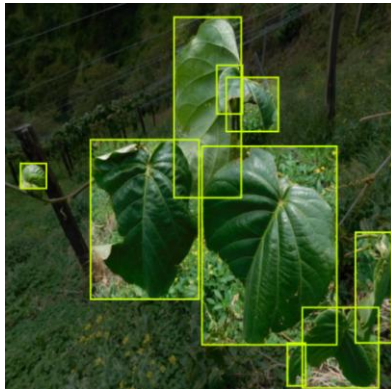
En este apartado se describen cada uno de los procesos realizados para obtener el conjunto de datos el cual su prioridad es el alimentar y entrenar la red neuronal que se enfoca en la detección temprana de Botrytis cinerea en hojas de granadilla. Entre algunos de los procedimientos principales se incluye el procesamiento de una gran cantidad de hojas entre buenas y malas; y la extracción de las imágenes, las rotaciones de estas para el robustecimiento del dataset, y el etiquetado correspondiente de cada imagen.

3.6.1.1 Procesamiento de Lotes y Extracción de Imágenes Individuales

Antes del entrenamiento de la red neuronal, se lleva el preprocesamiento de las imágenes obtenidas mediante fotografías. Este proceso se enfoca en clasificar las imágenes según el estado de las hojas, en la Figura 35 se muestran las imágenes diferenciando entren hojas en buen estado y hojas en mal estado. Esto permite la fiabilidad de la mejora de la información y facilita el análisis de los lotes.

Figura 35

Captura de imagen



a) Hojas en buen estado



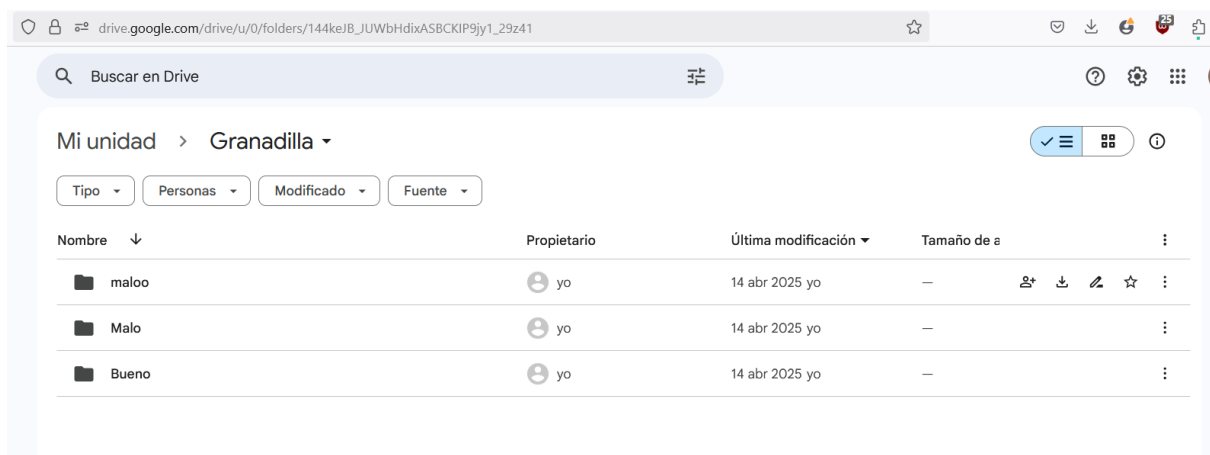
b) Hojas en mal estado

3.6.2 Entrenamiento con el Dataset Obtenido

En este primer entrenamiento se obtuvo un total de 471 imágenes en la base de datos que se dividen en 105 hojas malas y 366 hojas buenas de las cuales. De las cuales se utilizará el 80% de las mismas para el entrenamiento dando un total de 376 imágenes y el 20% para validación dando un total de 95 imágenes. Ya una vez con este dataset subido a la nube como se muestra en la figura 36 se puede crear un nuevo cuaderno, el entrenamiento de la red neuronal se lo realiza mediante el software Google colab, el cual tiene la capacidad para ejecutar código de Python en el mismo navegador, y es adecuado para tareas de tipo aprendizaje y análisis de datos sin tener un coste adicional.

Figura 36

Dataset en la nube

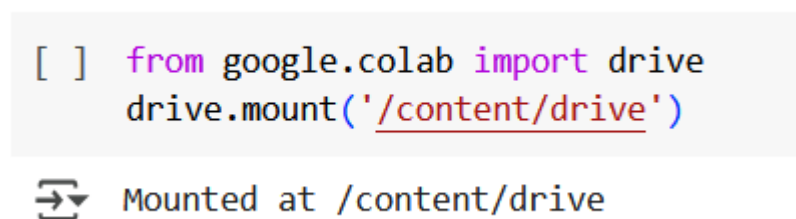


Nota. Autoría Propia

Como primer paso se debe cargar el dataset que previamente se cargó en Google drive, como se muestra en la figura 37.

Figura 37

Montaje de Google Drive en el entorno de Ejecución Colab

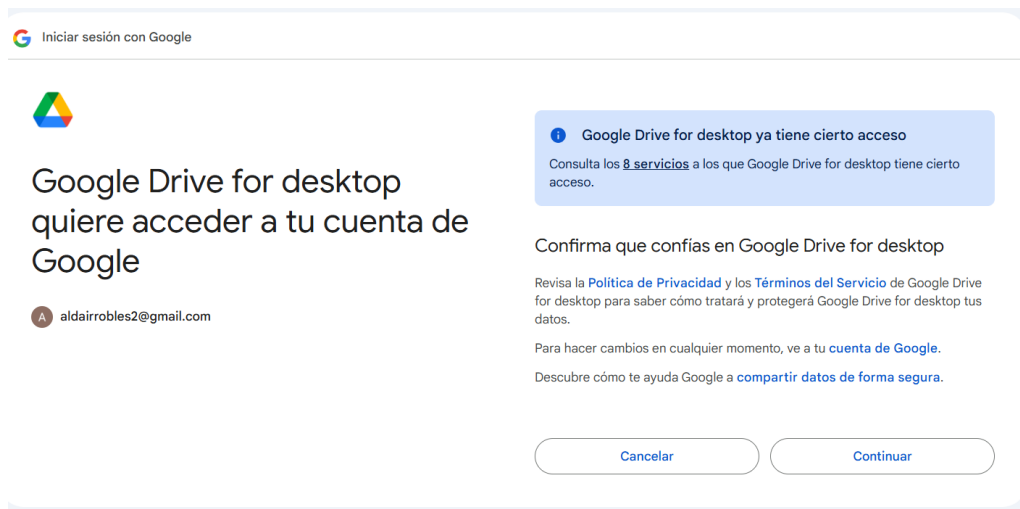


Nota. Autoría Propia

En la figura 38 se muestran los permisos que son necesarios para el uso de la herramienta Google drive. Una vez dados los permisos se puede crear y modificar los archivos para el entrenamiento.

Figura 38

Permisos de uso Google Drive



Nota. Autoría Propia

A continuación, se direccionan los directorios, de forma que exista un directorio para pruebas, test y validación, como se muestra en la figura 39.

Figura 39

Creación de directorios

```
# Ruta de las imágenes y etiquetas generadas
image_folder = os.path.join(dataset_path, "train/images")
label_folder = os.path.join(dataset_path, "train/labels")

# Crear las carpetas para 'train' y 'val' si no existen
train_image_folder = os.path.join(dataset_path, "t/images")
val_image_folder = os.path.join(dataset_path, "v/images")
train_label_folder = os.path.join(dataset_path, "t/labels")
val_label_folder = os.path.join(dataset_path, "v/labels")

# Crear las carpetas si no existen
os.makedirs(train_image_folder, exist_ok=True)
os.makedirs(val_image_folder, exist_ok=True)
os.makedirs(train_label_folder, exist_ok=True)
os.makedirs(val_label_folder, exist_ok=True)
```

Nota. Autoría Propia

Luego, los datos generados en la figura 34, se mezclan aleatoriamente los nombres de los archivos de imagen y luego se dividen en dos grupos 80% se destinan para el entrenamiento y el 20% se usan para la validación como se muestra en la figura 40.

Seguido a esto la función se recibe en una lista de imágenes y copia cada imagen y su archivo de etiqueta en formato .txt desde las carpetas de origen *src_image_folder* y *src_label_folder* a las carpetas de destino correspondientes *dst_image_folder* y *dst_label_folder*.

Si la imagen existe se copia a la carpeta de destino y seguido a esto se copia su archivo .txt de etiqueta correspondiente si no existe el archivo .txt, se muestra una advertencia y se elimina la imagen, ya que no puede entrenarse sin etiqueta.

Si no se encuentra la imagen pero si existe la etiqueta .txt se elimina esa etiqueta, ya que está huérfana (sin imagen asociada).

En la llamada de la función para el entrenamiento y la validación se copian los archivos a sus respectivas carpetas según su propósito: entrenamiento o validación.

Al final, se muestra un mensaje que indica cuántas imágenes fueron copiadas para cada conjunto; 2092 imágenes para entrenamiento, 523 imágenes para validación.

Figura 40

División de datos en conjuntos de entrenamiento y validación

```
# Función para copiar las imágenes y sus anotaciones
def copy_files(image_list, src_image_folder, src_label_folder, dst_image_folder, dst_label_folder):
    for image_name in image_list:
        # Copiar la imagen
        image_path = os.path.join(src_image_folder, image_name)
        if os.path.isfile(image_path):
            shutil.copy(image_path, os.path.join(dst_image_folder, image_name))

        # Copiar la etiqueta correspondiente si existe
        label_name = image_name.replace('.jpg', '.txt')
        label_path = os.path.join(src_label_folder, label_name)
        if os.path.isfile(label_path):
            shutil.copy(label_path, os.path.join(dst_label_folder, label_name))
        else:
            print(f"Advertencia: No se encontró la etiqueta para {image_name}, eliminando archivo jpg.")
            os.remove(os.path.join(src_image_folder, image_name)) # Eliminar la etiqueta si no hay imagen
    else:
        # Si la imagen no existe, eliminar su archivo de etiqueta si está presente
        label_name = image_name.replace('.jpg', '.txt')
        label_path = os.path.join(src_label_folder, label_name)
        if os.path.isfile(label_path):
            print(f"Advertencia: No se encontró la imagen {image_name}, eliminando archivo de etiqueta.")
            os.remove(label_path) # Eliminar la etiqueta si no hay imagen

# Copiar las imágenes y etiquetas de entrenamiento y validación
copy_files(train_images, image_folder, label_folder, train_image_folder, train_label_folder)
copy_files(val_images, image_folder, label_folder, val_image_folder, val_label_folder)

print(f"División completada: {len(train_images)} imágenes para entrenamiento, {len(val_images)} imágenes para validación.")
```

División completada: 376 imágenes para entrenamiento, 95 imágenes para validación.

Nota. Autoría Propia

En la figura 41 se puede ver la verificación tanto de la librería PyTorch como de una GPU disponible. Si la hay se imprime su nombre correspondiente, si no la hay se informa solamente que se utilizara la CPU, esto es importante porque el entrenamiento con GPU es mucho más rápido que con CPU.

Se importa el módulo YOLO de Ultralytics y se carga un modelo preentrenado muy liviano llamado YOLOv8n, que es ideal para dispositivos con recursos limitados o para entrenamientos más rápidos.

Aquí se inicia el entrenamiento con los siguientes parámetros:

- data: Ruta al archivo data.yaml, que contiene la configuración del dataset (clases, rutas a imágenes y etiquetas).

- epochs=50: Se entrenará el modelo durante 50 ciclos completos (épocas).
- batch=4: Se procesarán 4 imágenes a la vez (más grande = más rápido, pero más consumo de memoria).
- imgsz=640: Cada imagen se redimensiona a 640x640 píxeles.
- device=0: Se indica que se use la GPU (si está disponible).
- augment=True: Activa aumentación de datos, como rotaciones o recortes aleatorios, para mejorar la generalización del modelo.

Figura 41

Carga del modelo YOLOv8

```
[ ] import torch

# Verificar si la GPU está disponible
if torch.cuda.is_available():
    print("GPU está disponible:", torch.cuda.get_device_name(0))
else:
    print("GPU no está disponible.")
```

GPU está disponible: Tesla T4

```
from ultralytics import YOLO

# Cargar modelo preentrenado (Tiny YOLOv8)
model = YOLO("yolov8n.pt") # YOLOv8 Nano (Tiny)

# Entrenar el modelo
model.train(
    data="/content/drive/MyDrive/Alda2/data.yaml", # Archivo YAML del dataset
    epochs=50, # Más épocas para entrenar
    batch=4, # Tamaño del lote
    imgsz=640, # Tamaño de las imágenes
    device=0, # Usar GPU
    augment=False
)
```

Nota. Autoría Propia

El primero comando de la figura 42 elimina la carpeta `/content/runs/`, que es donde YOLO guarda automáticamente los resultados de entrenamiento y detección. Así se evita acumular archivos innecesarios.

Después de que se ejecuta el entrenamiento o la detección, este comando copia la carpeta `/content/runs` al Google Drive, en la ruta `/MyDrive/runs6`, para guardar los resultados .

YOLO de Ultralytics, se carga del modelo entrenado previamente, ubicado en la ruta del archivo `best.pt`, que contiene los pesos con mejor desempeño durante el entrenamiento en donde se especifica la ruta hacia una imagen de prueba que se quiere analizar.

Se realiza la detección de objetos en la imagen, y se establece un umbral de confianza del 80% (`conf=0.80`), lo que significa que solo se mostrarán las detecciones con alta probabilidad. Se renderiza directamente en Colab con la imagen original y los cuadros delimitadores marcando los objetos detectados como se muestra en la figura 43.

Figura 42

Detección de los objetos con el modelo YOLOv8

```
!rm -R /content/runs/

import locale
locale.getpreferredencoding = lambda: "UTF-8"
!cp -R /content/runs /content/drive/MyDrive/runs6

from ultralytics import YOLO

# Ruta a tu modelo entrenado
model_path = '/content/drive/MyDrive/runs6/detect/train/weights/best.pt'

# Cargar el modelo YOLOv8
model = YOLO(model_path)

# Ruta a la imagen de prueba
image_path = '/content/drive/MyDrive/Alda/t/images/DSCN1109_rot_180_JPG.rf.f5bd837b2e3e59e77baab0ef1b8fff67.jpg'

# Realizar la detección en la imagen de prueba
results = model(image_path, conf=0.80)

# Acceder al primer resultado (si hay más de un resultado)
result = results[0]

# Mostrar la imagen con las detecciones
result.show()
```

Nota. Autoría Propia

3.6.2.1 Resultados del Primer Entrenamiento Realizado

En el primer entrenamiento, se puede observar en la figura 43 el umbral de confianza registrado fue del 54% para la clase "Bad Leaf" y del 73% para la clase "Good Leaf". Estos valores indican que el modelo presenta un desempeño limitado, especialmente en la detección de hojas en mal estado. El nivel de confianza obtenido, particularmente en la clase "Bad Leaf", es relativamente bajo y no resulta suficiente para considerar que el modelo ha alcanzado un estado óptimo de aceptación o confiabilidad. Esto sugiere la necesidad de realizar ajustes adicionales, como la ampliación del conjunto de datos, una mejor calidad de las anotaciones, o una optimización de los hiperparámetros, con el fin de mejorar la precisión y la robustez del modelo en futuros entrenamientos.

Figura 43

Resultados de Evaluación de Desempeño del entrenamiento

```
[ 0.99691, 0.99691, 0.99691, ..., 0, 0, 0]], 'Confidence', 'Recall']
fitness: 0.6523097014053173
keys: ['metrics/precision(B)', 'metrics/recall(B)', 'metrics/mAP50(B)', 'metrics/mAP50-95(B)']
maps: array([ 0.54716, 0.73037])
names: (0, 'bad leaf', 1, 'good leaf')
plot: True
results_dict: {'metrics/precision(B)': 0.7920361131079289, 'metrics/recall(B)': 0.6672231067629404, 'metrics/mAP50(B)': 0.7742088276033046, 'metrics/mAP50-95(B)': 0.6387653540499854, 'fitness': 0.6523097014053173}
save_dir: PosixPath('runs/detect/train')
speed: {'preprocess': 0.31457746315362256, 'inference': 6.891597326315797, 'loss': 0.002236663158328156, 'postprocess': 5.6956247578981625}
task: 'detect'
```

Nota. Autoría Propia

Durante el proceso de evaluación del primer entrenamiento en la figura 44, se generó una matriz de confusión normalizada que permite visualizar el rendimiento del modelo al clasificar las imágenes en tres categorías: *bad leaf* (hoja en mal estado), *good leaf* (hoja en buen estado) y *background* (fondo).

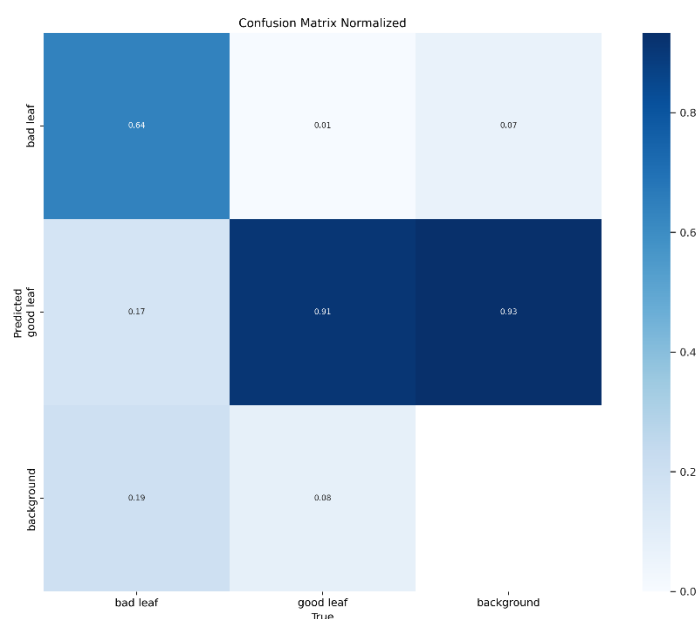
En términos generales, el modelo demuestra un desempeño destacado al clasificar correctamente las hojas en buen estado, alcanzando un valor de 0.91, lo que indica que el 91 % de las hojas buenas fueron correctamente identificadas por el modelo. Sin embargo, se observó una ligera confusión con otras clases, como un 1 % que fue clasificado erróneamente como *bad leaf* y un 8 % como *background*.

Por otro lado, el reconocimiento de las hojas en mal estado presenta un rendimiento más limitado. Solo el 64 % de las hojas en esta categoría fueron correctamente detectadas, mientras que el 17 % se clasificó erróneamente como hojas buenas y el 19 % como fondo. Este comportamiento evidencia que el modelo tiene dificultades para identificar con precisión las hojas en mal estado, posiblemente debido a una menor representación de esta clase en los datos de entrenamiento o a la similitud visual con las otras categorías.

Un aspecto importante que revela la matriz de confusión es la tendencia del modelo a clasificar erróneamente zonas del fondo como hojas buenas. Específicamente, el 93 % del fondo fue etiquetado como *good leaf*, lo cual afecta significativamente la fiabilidad del sistema, especialmente en escenarios donde es fundamental diferenciar claramente entre objeto y fondo.

Figura 44

Matriz de Confusión



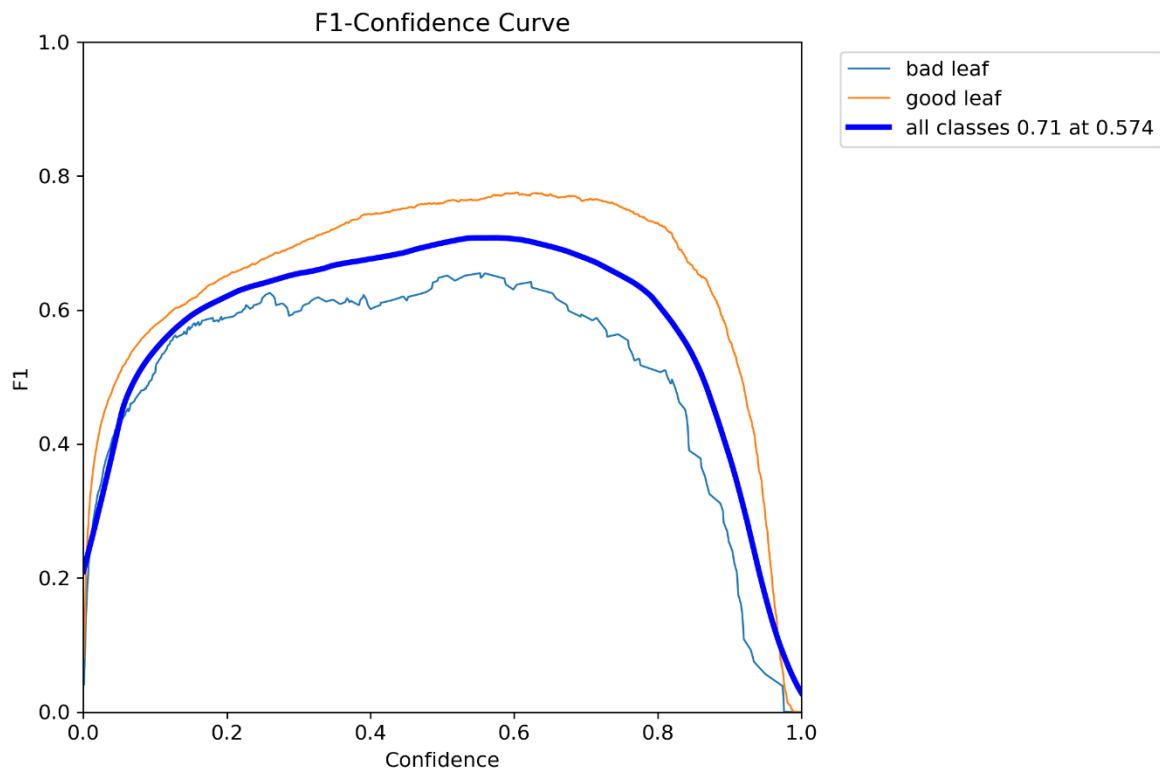
Nota. Autoría Propia

Además de la matriz de confusión, se generó la curva F1-Confidence en la figura 45, la cual proporciona una visión detallada sobre el comportamiento del modelo en función del nivel de confianza de sus predicciones.

En el gráfico se visualizan tres curvas: una correspondiente a la clase *bad leaf*, otra a la clase *good leaf*, y una curva promedio de todas las clases, destacada en azul grueso. Se observa que la clase *good leaf* alcanza los mejores valores de F1, superando los 0.75 en su punto más alto y mostrando una curva más estable a lo largo del eje de confianza. Esto evidencia que el modelo tiene una mayor capacidad de reconocer correctamente las hojas en buen estado con altos niveles de certeza.

En contraste, la clase *bad leaf* muestra un desempeño notablemente inferior. La curva asociada a esta categoría es más irregular y alcanza un valor máximo de F1 más bajo, indicando una menor precisión y/o sensibilidad en la identificación de hojas en mal estado.

El modelo alcanza su punto óptimo general en la intersección donde la curva azul gruesa (que representa todas las clases) alcanza su valor máximo. Este punto ocurre en un umbral de confianza de aproximadamente 0.574, con una puntuación F1 promedio de 0.71. Con estos datos aún no se alcanza un nivel aceptable de precisión por lo que se propone un aumento tanto de imágenes como de épocas en los siguientes entrenamientos.

Figura 45*Curva de Confidencialidad*

Nota. Autoría Propia

3.6.3 Aumento de Datos Aplicando Transformaciones Geométricas Para Hojas Buenas

Para el procedimiento de aumento de datos aplicando las transformadas geométricas se manejan cambio de rotación en las imágenes de 45°, 135°, 180°, 225°.

Dentro de la ubicación en donde se encuentra la base de datos en Google drive “input_path = "/content/drive/MyDrive/Database/Bueno"” la cual es la carpeta que contiene las imágenes originales en donde se obtiene las 5 primeras imágenes mostrada en la figura 46.

Figura 46*Transformación Geométrica de las Imágenes*

```

#aumento de Datos para imagenes Buenas
import os
import matplotlib.pyplot as plt
from PIL import Image

# Ruta de la carpeta Bueno (entrada)
input_path = "/content/drive/MyDrive/Database/Bueno"

# Obtener las primeras 5 imágenes
image_files = [f for f in os.listdir(input_path) if f.endswith('.JPG')]

# Función para aplicar solo rotaciones
def augment_image(image):
    augmented_images = []

    # Aplicar rotaciones
    for angle in [45,135,180,225]:
        augmented_images.append(image.rotate(angle))

    return augmented_images

# Seleccionar solo las primeras 5 imágenes
sample_images = image_files[:5]

# Mostrar una imagen con sus rotaciones
for img_name in sample_images:
    img_path = os.path.join(input_path, img_name)
    image = Image.open(img_path)

    # Aplicar las rotaciones
    augmented_images = augment_image(image)

    # Crear la figura y mostrar la imagen original y las rotaciones
    num_plots = len(augmented_images) + 1 # 1 para la imagen original + las rotaciones
    fig, axes = plt.subplots(1, num_plots, figsize=(18, 3)) # Ajustar el número de subgráficas
    axes = axes.ravel()

    # Mostrar la imagen original
    axes[0].imshow(image)
    axes[0].set_title(f"Original {img_name}")
    axes[0].axis('off')

    # Mostrar las rotaciones
    for i, aug_img in enumerate(augmented_images):
        axes[i + 1].imshow(aug_img)
        axes[i + 1].set_title(f"Rotación {i}")
        axes[i + 1].axis('off')

    plt.tight_layout()
    plt.show()

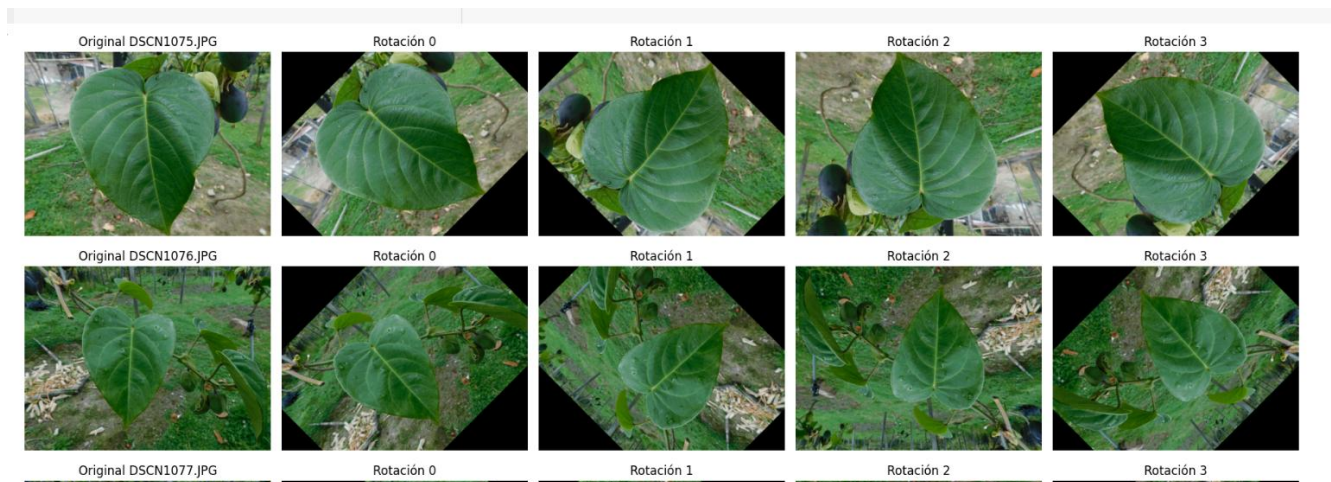
```

Nota. Autoría Propia

Las imágenes que serán mostradas dentro de Google colab para una mejora visualización, se muestran en la primera casilla la imagen original y después a eso cada una de las rotaciones que ha dado la imagen y son mostradas en la figura 47.

Figura 47

Imagen Original Junto con sus Transformaciones.



Nota. Autoría Propia

En la segunda parte aplicada se definen las rutas de entrada y salida; las cuales son las rutas en donde se toman las imágenes dentro de Google drive, con el directorio `“os.listdir(input_path)”` y la ruta de salida con el directorio `“if f.endswith('.JPG')”` que es la carpeta en donde se guardarán las nuevas imágenes para el entrenamiento.

Se aplican las rotaciones tanto de 45° , 135° , 180° , 225° , se las guarda en un formato .JPG y se etiqueta recibiendo un nombre único mediante la función `“_rot_{angle}”`, se devuelven las rutas de las imágenes generadas.

Por último la imagen original se verá cargada y mediante el argumento `“augment_image()”` se aplican las rotaciones y las guarda en el directorio de salida, la Figura 48 es la que muestra el aumento de las imágenes anteriormente explicadas.

Figura 48*Aumento de Datos mediante Rotaciones y Almacenamiento de Imágenes*

```

import os
from PIL import Image

# Ruta de la carpeta Bueno (entrada) y la carpeta de salida
input_path = "/content/drive/MyDrive/Database/Bueno"
output_path = "/content/drive/MyDrive/Database/Bueno_Augmented" # Ruta de salida

# Crear la carpeta de salida si no existe
os.makedirs(output_path, exist_ok=True)

# Obtener las imágenes en la carpeta de entrada
image_files = [f for f in os.listdir(input_path) if f.endswith('.JPG')]

# Función para aplicar solo rotaciones
def augment_image(image, image_name):
    augmented_images = []

    # Aplicar rotaciones y guardar las imágenes rotadas
    for angle in [45, 135, 180, 225]:
        rotated_image = image.rotate(angle)
        # Guardar la imagen rotada con un nombre único
        rotated_image_name = f"{image_name.replace('.JPG', '')}_rot_{angle}.JPG"
        rotated_image_path = os.path.join(output_path, rotated_image_name)
        rotated_image.save(rotated_image_path)
        augmented_images.append(rotated_image_path)

    return augmented_images

# Procesar todas las imágenes en la carpeta de entrada
for img_name in image_files:
    img_path = os.path.join(input_path, img_name)
    image = Image.open(img_path)

    # Aplicar las rotaciones y guardar las imágenes
    augmented_images = augment_image(image, img_name)

    # Mostrar las imágenes originales y las rotadas
    print(f"Imágenes aumentadas para {img_name}:")
    print("\n".join(augmented_images))

print("Aumento de datos completado. Las imágenes rotadas han sido guardadas en la carpeta de salida.")

```

Nota. Autoría Propia

3.6.4 Aumento de Datos Aplicando Transformaciones Geométricas Para Hojas Malas

En este caso para el aumento de datos aplicando transformaciones geométricas por hojas malas, se utiliza el mismo concepto de llamar a la carpeta dentro de Google drive llamada “/content/drive/MyDrive/Database/Malo”. Se obtiene las 5 primeras imágenes de la carpeta de entrada y los lista según los archivos .JPG. Seguido a esto se utiliza la función “*augment_image*” para almacenar los nombres de los archivos generados, la Figura 49 muestra todos estos parámetros que fueron utilizados.

Figura 49*Aumento de Datos Mediante Transformaciones Geométricas en Imágenes Defectuosas*

```

import os
import random
from PIL import Image, ImageEnhance, ImageOps
import matplotlib.pyplot as plt

# Ruta de la carpeta Malo (entrada) y la carpeta de salida
input_path_malo = "/content/drive/MyDrive/Database/Malo"
output_path_malo = "/content/drive/MyDrive/Database/Malo_Augmented" # Ruta de salida

# Crear la carpeta de salida si no existe
os.makedirs(output_path_malo, exist_ok=True)

# Obtener las primeras 5 imágenes en la carpeta de entrada
image_files_malo = [f for f in os.listdir(input_path_malo) if f.endswith('.JPG')][:5]

# Función para aplicar varias técnicas de aumento de datos
def augment_image(image, image_name):
    augmented_images = []

```

Nota. Autoría Propia

3.6.4.1 Rotación de las Imágenes

La siguiente parte del código en la Figura 50 se puede ver como los ángulos 45°, 135°, 180° y 225° generan una nueva versión para cada una de las rotaciones de las imágenes. Luego crea un nombre de archivo para cada una de las imágenes rotadas agregando un nuevo sufijo “_rot_{angle}.JPG”, al nombre que tenía en un principio. Concluye esta ruta guardando en la carpeta de la salida “output_path_malo”, y guarda cada imagen en dicha salida. Finalmente, se añade los nombres a las imágenes generadas a la lista “augmented_images” permitiendo así que los archivos creados tengan un fácil seguimiento.

Figura 50

Función de Rotaciones

```
# Función para aplicar varias técnicas de aumento de datos
def augment_image(image, image_name):
    augmented_images = []

    # Aplicar rotaciones
    for angle in [45, 135, 180, 225]:
        rotated_image = image.rotate(angle)
        rotated_image_name = f"{image_name.replace('.JPG', '')}_rot_{angle}.JPG"
        rotated_image_path = os.path.join(output_path_malo, rotated_image_name)
        rotated_image.save(rotated_image_path)
        augmented_images.append(rotated_image_path)
```

Nota. Autoría Propia

3.6.4.2 Función Espejo (Flipping horizontal y vertical)

En la Figura 51 se puede observar como el código genera dos versiones espejadas de la imagen original; una utiliza un reflejo horizontal “*FLIP_LEFT_RIGHT*”, y la otra utiliza un reflejo vertical “*FLIP_TOP_BOTTOM*”. Luego asigna nuevos nombres a estos archivos en los cuales se agrega los sufijos “*_flipped_h.JPG*” y “*_flipped_v.JPG*”, respectivamente. Cada una de las imágenes modificadas se guardan dentro de la carpeta “*augmented_images*” para llevar un registro de los archivos generados.

Figura 51

Función de Espejo

```
# Flipping horizontal y vertical
flipped_h = image.transpose(Image.FLIP_LEFT_RIGHT)
flipped_v = image.transpose(Image.FLIP_TOP_BOTTOM)
flipped_h_name = f"{image_name.replace('.JPG', '')}_flipped_h.JPG"
flipped_v_name = f"{image_name.replace('.JPG', '')}_flipped_v.JPG"
flipped_h.save(os.path.join(output_path_malo, flipped_h_name))
flipped_v.save(os.path.join(output_path_malo, flipped_v_name))
augmented_images.extend([flipped_h_name, flipped_v_name])
```

Nota. Autoría Propia

3.6.4.3 Función de Brillo y Espejo

El código modifica el brillo y el contraste de la imagen original como se muestra en la Figura 52 . Primero se utiliza “*ImageEnhance.Brightness*” para causar un aumento de brillo de la imagen en un 50% y esta misma se guarda utilizando el sufijo “*_bright.JPG*”. luego a esto se aplica “*ImageEnhance.Contrast*” que al igual que el brillo se aumentó el contraste en un 50% y guarda la imagen con el sufijo “*_contrast.JPG*”. Ambas imágenes se almacenan en la carpeta “*output_path_malo*”, y estos nombres se añaden a la lista “*augmented_images*” para el registro de los archivos generados.

Figura 52

Función de Brillo y Espejo

```
# Cambiar el brillo
enhancer = ImageEnhance.Brightness(image)
bright_image = enhancer.enhance(1.5) # Aumentar brillo
bright_image_name = f"{image_name.replace('.JPG', '')}_bright.JPG"
bright_image.save(os.path.join(output_path_malo, bright_image_name))
augmented_images.append(bright_image_name)

# Cambiar el contraste
enhancer = ImageEnhance.Contrast(image)
contrast_image = enhancer.enhance(1.5) # Aumentar contraste
contrast_image_name = f"{image_name.replace('.JPG', '')}_contrast.JPG"
contrast_image.save(os.path.join(output_path_malo, contrast_image_name))
augmented_images.append(contrast_image_name)
```

Nota. Autoría Propia

3.6.4.4 Desplazamiento Aleatorio (Translation)

En el código se aplica un desplazamiento aleatorio como se muestra en la Figura 53 dentro de un rango de -10 a 10 píxeles en los ejes X y Y. primeramente se obtendrá el tamaño de la imagen y se generan valores aleatorios para el desplazamiento horizontal “*offset_x*” y vertical “*offset_y*”. Este crea una matriz de transformación para aplicar el desplazamiento y

genera una nueva imagen desplazada. Esta misma se guarda dentro de la carpeta “*output_path_malo*” con el sufijo “*_translated.JPG*”, se añade el nombre a la lista para generar un archivo de registro.

Figura 53

Desplazamiento Aleatorio de Imágenes Mediante Transformación Afín

```
# Desplazamiento (Translation) - Corregido
max_offset = 10
width, height = image.size
offset_x = random.randint(-max_offset, max_offset)
offset_y = random.randint(-max_offset, max_offset)

# Usar `Image.transform` para mover la imagen
translation_matrix = [1, 0, offset_x, 0, 1, offset_y] # Desplazamiento en X, Y
translated_image = image.transform(image.size, Image.AFFINE, translation_matrix)
translated_image_name = f"{image_name.replace('.JPG', '')}_translated.JPG"
translated_image.save(os.path.join(output_path_malo, translated_image_name))
augmented_images.append(translated_image_name)
```

Nota. Autoría Propia

3.6.4.5 Aplicación de Zoom Aleatorio en Imágenes

En la figura 54 el código aplica un zoom aleatorio a la imagen, el cual hará que este mismo se amplíe entre 1.2x y 1.5x del tamaño que tenía originalmente. Luego, se selecciona una región del tamaño original haciéndolo mediante coordenadas aleatorias dentro de la imagen ampliada y se recorta para mantener dimensiones originales. Al obtener la imagen resultante se la guarda dentro de la carpeta “*output_path_malo*” con el sufijo “*_zoomed.JPG*” y su nombre se añade a la lista para registrar el archivo que previamente fue generado.

Figura 54*Zoom Aleatorio a las Imágenes*

```

# Zoom aleatorio (Redimensionar y recortar)
scale_factor = random.uniform(1.2, 1.5) # Zoom entre 1.2x y 1.5x
resized_image = image.resize((int(width * scale_factor), int(height * scale_factor)))
left = random.randint(0, resized_image.width - width)
top = random.randint(0, resized_image.height - height)
right = left + width
bottom = top + height
zoomed_image = resized_image.crop((left, top, right, bottom))
zoomed_image_name = f"{image_name.replace('.JPG', '')}_zoomed.JPG"
zoomed_image.save(os.path.join(output_path_malo, zoomed_image_name))
augmented_images.append(zoomed_image_name)

```

Nota. Autoría Propia

3.6.4.6 Visualización de Transformaciones Aplicadas a las Imágenes

Como muestra la figura 55 al inicio se abre la imagen a utilizar y se obtiene su ruta de acceso, se pasan las imágenes y el nombre de su función con “*augment_image()*” que devuelve una lista de imágenes aumentadas.

Figura 55

Código de Visualización de Transformada Aplicadas a las Imágenes

```
# Mostrar las transformaciones de las primeras 5 imágenes
for img_name in image_files_malo:
    img_path = os.path.join(input_path_malo, img_name)
    image = Image.open(img_path)

    # Aplicar las transformaciones y obtener las imágenes aumentadas
    augmented_images = augment_image(image, img_name)

    # Crear la figura para mostrar las imágenes originales y aumentadas
    num_plots = len(augmented_images) + 1 # 1 para la imagen original + las aumentadas
    fig, axes = plt.subplots(1, num_plots, figsize=(18, 3)) # Ajustar el número de subgráficas
    axes = axes.ravel()

    # Mostrar la imagen original
    axes[0].imshow(image)
    axes[0].set_title(f"Original {img_name}")
    axes[0].axis('off')

    # Mostrar las imágenes aumentadas
    for i, aug_img in enumerate(augmented_images):
        aug_image = Image.open(os.path.join(output_path_malo, aug_img))
        axes[i + 1].imshow(aug_image)
        axes[i + 1].set_title(f"Aug {i+1}")
        axes[i + 1].axis('off')

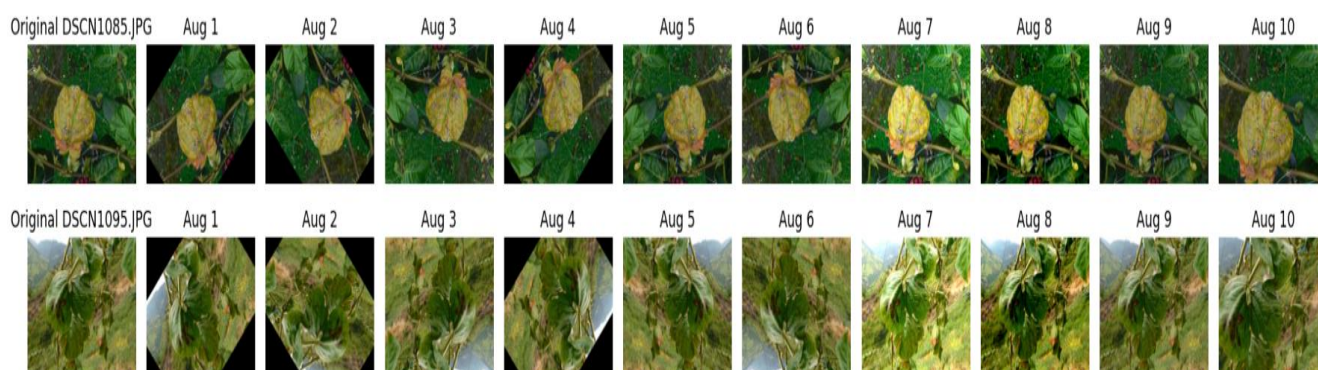
    plt.tight_layout()
    plt.show()

print("Transformaciones aplicadas y visualizadas para las primeras 5 imágenes.")
```

Nota. Autoría Propia

Utilizando matplotlib se crea una figura con subgráficas, el número de estas es igual a las imágenes aumentadas más 1. Se ajusta el tamaño para que todas las imágenes sean visibles.

Con la ayuda de “*plt.tight_layout()*” se ajusta el diseño de las subgráficas y “*plt.show()*” mostrara la figura con todas las imágenes como en la Figura 56 a continuación.

Figura 56*Salida de las Imágenes*

Nota. Autoría Propia

Con el aumento de datos la cantidad de imágenes correspondientes a hojas en buen estado aumentó de 366 a 1464, mientras que las imágenes de hojas en mal estado pasaron de 105 a 1050. Esta expansión fue crucial para balancear las clases y reducir el sesgo en el entrenamiento del modelo. Al contar con una mayor cantidad de ejemplos representativos de ambas categorías, se procede a hacer un nuevo entrenamiento para ver la mejora de este.

3.6.5 Entrenamiento de la Red Neuronal con el Robustecimiento del Dataset

Una vez que se cuenta con un dataset más robusto y equilibrado, se procede a realizar una serie de pruebas experimentales. El objetivo principal de estas pruebas es evaluar el impacto del número de épocas en el rendimiento del modelo, observando cómo varía su capacidad de clasificación conforme se incrementa el número de ciclos de entrenamiento. Para ello, se realizaron múltiples entrenamientos ajustando el parámetro de épocas, lo que permitió analizar la evolución del desempeño del modelo en términos de métricas como la precisión, la matriz de confusión y la curva F1-Confidence. Esta etapa es fundamental para determinar el

punto óptimo en el que el modelo alcanza su mejor nivel de generalización sin incurrir en sobreajuste.

3.6.5.1 Pruebas del Entrenamiento con las Primeras 25 Épocas.

Una vez mejorado y ampliado el conjunto de datos, se procedió a realizar un nuevo entrenamiento del modelo. En esta ocasión, se utilizó un total de 1,865 imágenes para el conjunto de entrenamiento, lo que representa un aumento considerable respecto a la versión anterior del dataset. Además, se configuró el entrenamiento para ejecutarse durante 25 épocas como se muestra en la figura 57, lo cual permitió al modelo tener más oportunidades de ajustar sus pesos y aprender de forma más profunda las características relevantes de cada clase.

Figura 57

Entrenamiento con 25 épocas

```
from ultralytics import YOLO

# Cargar modelo preentrenado (Tiny YOLOv8)
model = YOLO("yolov8n.pt") # YOLOv8 Nano (Tiny)

# Entrenar el modelo
model.train(
    data="/content/drive/MyDrive/Alda/dataset.yaml", # Archivo YAML del dataset
    epochs=25, # Más épocas para entrenar
    batch=4, # Tamaño del lote
    imgsz=640, # Tamaño de las imágenes
    augment=True # Activar aumentación
)
```

Nota. Autoría Propia

En cuanto al desempeño por clase en la Figura 58, se observa que el modelo logra una mAP promedio de 0.774 para la clase *bad leaf*, lo que indica una buena capacidad para identificar correctamente las hojas en mal estado.

Para la clase *good leaf*, se obtuvo una mAP de 0.744, lo cual refleja una alta precisión en la detección de hojas saludables. Finalmente, la clase *person*, añadida para controlar posibles interferencias humanas en las imágenes, alcanzó una mAP de 0.552, un valor moderado que

puede explicarse por la menor representación de esta clase en el conjunto de datos. A pesar de ello, el modelo logra distinguir adecuadamente la presencia de personas cuando aparecen, evitando falsas detecciones sobre las hojas. En conjunto, estos resultados evidencian que el modelo es capaz de generalizar correctamente entre diferentes clases de objetos relevantes para el caso de estudio.

Figura 58

Resultados de Evaluación de Desempeño

```
fitness: np.float64(0.7023011125474873)
keys: ['metrics/precision(B)', 'metrics/recall(B)', 'metrics/mAP50(B)', 'metrics/mAP50-95(B)']
maps: array([ 0.77436, 0.74464, 0.552421])
names: {0: 'bad leaf', 1: 'good leaf', 2: 'person'}
plot: True
results_dict: {'metrics/precision(B)': np.float64(0.7034177430468702), 'metrics/recall(B)': np.float64(0.7893409589016284), 'metrics/mAP50(B)': np.float64(0.7023011125474873), 'metrics/mAP50-95(B)': np.float64(0.6893409589016284)}
save_dir: PosixPath('runs/detect/train2')
speed: {'preprocess': 0.3505943907782142, 'inference': 10.506292521845308, 'loss': 0.00159874272063921, 'postprocess': 1.6767818519396827}
task: 'detect'
```

Nota. Autoría Propia

3.6.5.2 Pruebas del Entrenamiento con 50 Épocas.

Una vez finalizado el entrenamiento con 25 épocas, y tras analizar los resultados obtenidos, se decidió realizar un nuevo entrenamiento, esta vez aumentando la cantidad de épocas a 50 como se muestra en la Figura 59. El objetivo de esta prueba fue evaluar si una mayor cantidad de ciclos de entrenamiento permitiría al modelo alcanzar un mejor ajuste a los datos, incrementando así su capacidad de generalización y reduciendo posibles errores en la detección y clasificación de las hojas. Este nuevo experimento se llevó a cabo utilizando el mismo conjunto de datos robustecido de 1865 imágenes, manteniendo constantes los demás parámetros, con el fin de que la variable evaluada fuera exclusivamente la cantidad de épocas.

Figura 59*Entrenamiento con 50 Épocas*

```

from ultralytics import YOLO

# Cargar modelo preentrenado (Tiny YOLOv8)
model = YOLO("yolov8n.pt") # YOLOv8 Nano (Tiny)

# Entrenar el modelo
model.train(
    data="/content/drive/MyDrive/Alda/dataset.yaml", # Archivo YAML del dataset
    epochs=50, # Más épocas para entrenar
    batch=4, # Tamaño del lote
    imgsz=640, # Tamaño de las imágenes
    augment=True # Activar aumentación
)

```

Nota. Autoría Propia

Al finalizar el entrenamiento en la Figura 60 al entrenar con 50 épocas, las métricas mostraron una mejora en todos los aspectos: la precisión aumentó a 0.747, la recuperación a 0.799, el mAP@0.5 se elevó a 0.839, y el mAP@0.5:0.95 alcanzó un valor de 0.729. Asimismo, el fitness general del modelo mejoró hasta 0.740, confirmando que un mayor número de épocas permitió una mejor adaptación del modelo a los datos de entrenamiento.

Figura 60*Resultados de Evaluación de Desempeño*

```

fitness: np.float64(0.7401583663192112)
keys: ['metrics/precision(B)', 'metrics/recall(B)', 'metrics/mAP50(B)', 'metrics/mAP50-95(B)']
maps: array([ 0.80419,  0.75786,  0.62522])
names: {0: 'bad leaf', 1: 'good leaf', 2: 'person'}
plot: True
results_dict: {'metrics/precision(B)': np.float64(0.7472388841845073), 'metrics/recall(B)': np.float64(0.7994097655650422), 'metrics/mAP50(B)': np.float64(0.8390000000000001), 'metrics/mAP50-95(B)': np.float64(0.7290000000000001)}
save_dir: PosixPath('runs/detect/train3')
speed: {'preprocess': 0.3755675970883777, 'inference': 9.568048368930373, 'loss': 0.0014832427213441759, 'postprocess': 1.4888733179587397}
task: 'detect'

```

Nota. Autoría Propia

3.6.5.3 Pruebas del Entrenamiento con 75 Épocas.

Posteriormente en la Figura 61, se llevó a cabo un tercer entrenamiento del modelo, esta vez incrementando el número de épocas a 75 con el objetivo de evaluar si el desempeño del modelo seguía mejorando con entrenamientos más prolongados. Este proceso se realizó manteniendo las mismas configuraciones previas en cuanto al tamaño del lote, el tamaño de imagen y los datos de entrada, permitiendo así una comparación justa entre las distintas sesiones de entrenamiento.

Figura 61

Entrenamiento con 75 Épocas

```

▶ from ultralytics import YOLO

# Cargar modelo preentrenado (Tiny YOLOv8)
model = YOLO("yolov8n.pt") # YOLOv8 Nano (Tiny)

# Entrenar el modelo
model.train(
    data="/content/drive/MyDrive/Alda/dataset.yaml", # Archivo YAML del dataset
    epochs=75, # Más épocas para entrenar
    batch=4, # Tamaño del lote
    imgsz=640, # Tamaño de las imágenes
    augment=True # Activar aumentación
)

```

Nota. Autoría Propia

Como puede observarse en la Figura 62, el modelo mejora ligeramente en la mayoría de los indicadores al aumentar el número de épocas. El valor de fitness más alto se obtiene con 75 épocas (0.7534), lo cual sugiere que este entrenamiento proporciona el mejor balance entre precisión y capacidad de generalización en el conjunto de validación.

Sin embargo, se aprecia que el incremento de rendimiento comienza a estabilizarse entre las 50 y 75 épocas, lo cual puede indicar que se está alcanzando un punto óptimo de aprendizaje, más allá del cual los beneficios marginales podrían no justificar el costo computacional adicional.

Figura 62*Resultados de Evaluación de Desempeño*

```

    L    0.91121,    0.91121,    0.90010, ...,    0,    0,    0.11, CONFIDENCE, RECALL JJ
fitness: np.float64(0.7534204657163743)
keys: ['metrics/precision(B)', 'metrics/recall(B)', 'metrics/mAP50(B)', 'metrics/mAP50-95(B)']
maps: array([[ 0.81274,  0.77466,  0.6433]])
names: {0: 'bad leaf', 1: 'good leaf', 2: 'person'}
plot: True
results_dict: {'metrics/precision(B)': np.float64(0.7730186603260788), 'metrics/recall(B)': np.float64(0.7870515174559806), 'metrics/mAP50(B)':
save_dir: PosixPath('runs/detect/train4')
speed: {'preprocess': 0.3062269999636821, 'inference': 10.551006752415807, 'loss': 0.0016803179560774526, 'postprocess': 2.1840844611794186}
task: 'detect'

```

Nota. Autoría Propia

3.6.5.4 Pruebas del Entrenamiento con 100 Épocas.

Como se observa en la Figura 63, se emplearon 100 épocas de entrenamiento con el objetivo de analizar el comportamiento del modelo, específicamente para determinar la presencia de sobreajuste y evaluar si el sistema podría seguir mejorando su desempeño al incrementar el número de épocas.

Figura 63*Entrenamiento con 100 Épocas*

```

▶ from ultralytics import YOLO

# Cargar modelo preentrenado (Tiny YOLOv8)
model = YOLO("yolov8n.pt") # YOLOv8 Nano (Tiny)

# Entrenar el modelo
model.train(
    data="/content/drive/MyDrive/Alda2/data.yaml", # Archivo YAML del dataset
    epochs=100, # Más épocas para entrenar
    batch=4, # Tamaño del lote
    imgsz=640, # Tamaño de las imágenes
    #device=0, # Usar GPU
    augment=False
)

```

Nota. Autoría Propia

Como puede observarse en la Figura 64, el modelo muestra una mejora progresiva en la mayoría de los indicadores al incrementar el número de épocas desde 25 hasta 75. El valor más alto de *fitness* se obtiene con 75 épocas (0.7534), lo que sugiere que este entrenamiento proporciona el mejor equilibrio entre precisión y capacidad de generalización en el conjunto de validación. No obstante, al extender el entrenamiento hasta 100 épocas, aunque la precisión aumenta, se evidencia una disminución en métricas relevantes como el *recall*, el mAP@0.5 y el mAP@0.5:0.95, así como una reducción del *fitness* (0.7300).

Este comportamiento indica que el modelo comienza a perder capacidad de generalización, lo que podría asociarse a un proceso de sobreajuste. Por tanto, se concluye que el punto óptimo de aprendizaje se alcanza en 75 épocas, ya que entrenamientos más prolongados no aportan mejoras significativas y pueden incluso degradar el desempeño global del modelo.

Figura 64

Resultados de Evaluación de Desempeño

```
[1, ..., 0, 0, 0]], 'Confidence', 'Recall']]
fitness: np.float64(0.8893441176161214)
keys: ['metrics/precision(B)', 'metrics/recall(B)', 'metrics/mAP50(B)', 'metrics/mAP50-95(B)']
maps: array([ 0.8935, 0.88519])
names: {0: 'bad leaf', 1: 'good leaf'}
nt_per_class: array([ 180, 181])
nt_per_image: array([ 33, 140])
results_dict: {'metrics/precision(B)': 0.8974252745563462, 'metrics/recall(B)': 0.9117288054886661, 'metrics/mAP50(B)': 0.9615446538469994, 'metrics/mAP50-95(B)': 0.8893441176161214, 'fitness': 0.8893441176161214}
save_dir: PosixPath('/content/runs/detect/train')
speed: {'preprocess': 0.4923267514473108, 'inference': 5.033025901737693, 'loss': 0.001621820812070348, 'postprocess': 4.6548973179197395}
stats: {'tp': [], 'conf': [], 'pred_cls': [], 'target_cls': [], 'target_img': []}
task: 'detect'
```

Nota. Autoría Propia

3.6.5.5 Comparativa de Épocas: Criterios Para Determinar la Mejor Opción

Una vez fortalecido el dataset con un mayor número de imágenes, se procedió a realizar un conjunto de entrenamientos variando la cantidad de épocas con el objetivo de identificar el punto óptimo de aprendizaje del modelo de detección.

Se realizaron cuatro entrenamientos con 25, 50, 75 y 100 épocas, manteniendo constantes los demás parámetros (modelo YOLOv8n, tamaño de imagen de 640 px, lote de 4 imágenes y aumentación activada). Al finalizar cada entrenamiento, se evaluaron los indicadores de desempeño que se presentan en la Tabla 15.

Tabla 15

Comparativa de Épocas

Comparativa de épocas					
N.º de Épocas	Precisión	Recall	mAP@0.5	mAP@0.5:0.95	Fitness
25	0.7034	0.7893	0.8088	0.6905	0.7023
50	0.7472	0.7994	0.8389	0.7291	0.7402
75	0.7730	0.7870	0.8421	0.7436	0.7534
100	0.8120	0.7200	0.8200	0.7100	0.7300

Como se puede observar en la figura 65, al incrementar el número de épocas desde 25 hasta 75, las métricas muestran una mejora progresiva. La precisión aumenta de 0.7034 a 0.7730, mientras que el mAP@0.5 pasa de 0.8088 a 0.8421, evidenciando una mejor capacidad del modelo para detectar correctamente las hojas afectadas por la enfermedad.

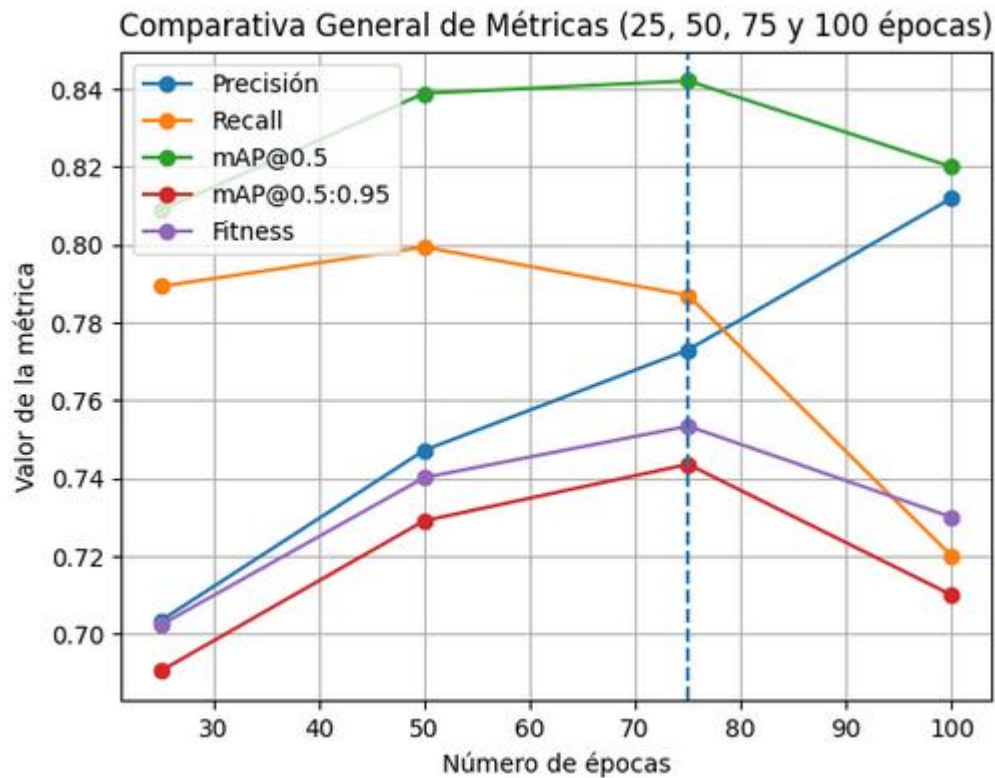
Asimismo, la métrica mAP@0.5:0.95 presenta un crecimiento sostenido hasta las 75 épocas, alcanzando un valor de 0.7436. Este indicador es especialmente relevante, ya que evalúa el rendimiento del modelo bajo distintos umbrales de IoU, reflejando su capacidad de generalización.

La métrica fitness, que resume globalmente el desempeño del modelo, alcanza su valor máximo en 75 épocas (0.7534), lo que sugiere que en este punto el modelo logra el mejor equilibrio entre precisión y detección.

Sin embargo, al extender el entrenamiento hasta 100 épocas, se observa un comportamiento diferente. Aunque la precisión aumenta hasta 0.8120, el recall disminuye considerablemente a 0.7200, lo que indica que el modelo comienza a omitir objetos reales. De igual forma, tanto el $mAP@0.5$ como el $mAP@0.5:0.95$ presentan una reducción respecto a las 75 épocas, y la métrica fitness desciende a 0.7300.

Figura 65

Comparativa General de Métricas Para la Selección del Número Optimo de Épocas



Nota. Autoría Propia

Otro de los factores como se ve en la figura 64 se puede ver que el modelo de 50 épocas tiene:

- El modelo de 50 épocas presenta un recall ligeramente superior (0.7994 frente a 0.7870), lo que indica una mayor capacidad para detectar objetos reales dentro del conjunto de validación. Asimismo, muestra un comportamiento competitivo en la métrica mAP@0.5 (0.8389), evidenciando una adecuada calidad de detección.
- Sin embargo, el modelo entrenado con 75 épocas supera al de 50 épocas en métricas clave de precisión (0.7730 frente a 0.7472), lo que implica una menor cantidad de falsos positivos. Además, alcanza el valor más alto de mAP@0.5 (0.8421) y mAP@0.5:0.95 (0.7436), lo que demuestra un mejor desempeño bajo distintos umbrales de IoU y, por tanto, una mayor capacidad de generalización.
- Adicionalmente, la métrica fitness también alcanza su valor máximo en 75 épocas (0.7534), lo que confirma que este modelo ofrece el mejor equilibrio global entre precisión y capacidad de detección.
- En consecuencia, aunque el modelo de 50 épocas presenta un recall ligeramente superior, el modelo de 75 épocas muestra un desempeño más equilibrado y robusto, consolidándose como la mejor configuración de entrenamiento dentro del presente estudio.

3.6.5.6 Validación del Modelo: Gráficas de Precisión y Pérdida Del Modelo Entrenado

Posteriormente, se realizan las gráficas de pérdida y precisión con el fin de analizar la evolución del entrenamiento de la red, a su vez se realiza una matriz de confusión para conocer el porcentaje de precisión del modelo entrenado.

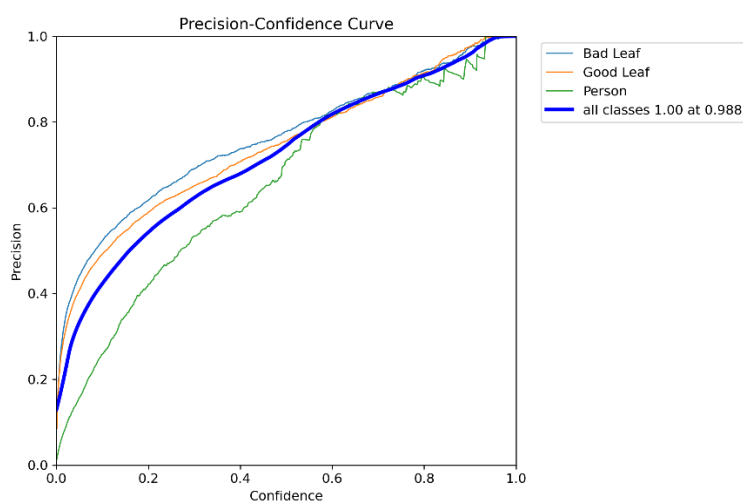
Como se observa en la Figura 66, la curva Precisión–Confianza evidencia un comportamiento creciente en todas las clases evaluadas, lo que indica que el modelo YOLOv8n mejora la confiabilidad de sus predicciones a medida que se incrementa el umbral de confianza.

La clase correspondiente a hojas enfermas (Bad Leaf), asociada a la presencia de *Botrytis cinerea*, mantiene valores de precisión elevados incluso en niveles intermedios de confianza, lo cual resulta fundamental para garantizar la efectividad del sistema de detección temprana. De manera similar, la clase Good Leaf presenta un comportamiento estable, mientras que la clase Person muestra mayor variabilidad en umbrales bajos, sugiriendo mayor incertidumbre inicial en sus predicciones.

En términos globales, la curva promedio (all classes) alcanza una precisión cercana a 1.00 con un umbral de 0.988, evidenciando una alta capacidad de discriminación y una reducción significativa de falsos positivos cuando se emplean configuraciones estrictas, lo que demuestra la robustez del modelo en la tarea de clasificación y detección.

Figura 66

Gráfica de Precisión



Nota. Autoría Propia

En la figura 67 se muestra la gráfica *val/cls_loss* representa la pérdida de clasificación en el conjunto de validación es decir esta pérdida mide que tanto se equivoca el modelo al reconocer y clasificar los objetos detectados, en el grafico se observa una disminución continua desde los 2.0; lo cual significa que el modelo cometía muchos errores hasta los 0.8; que indica

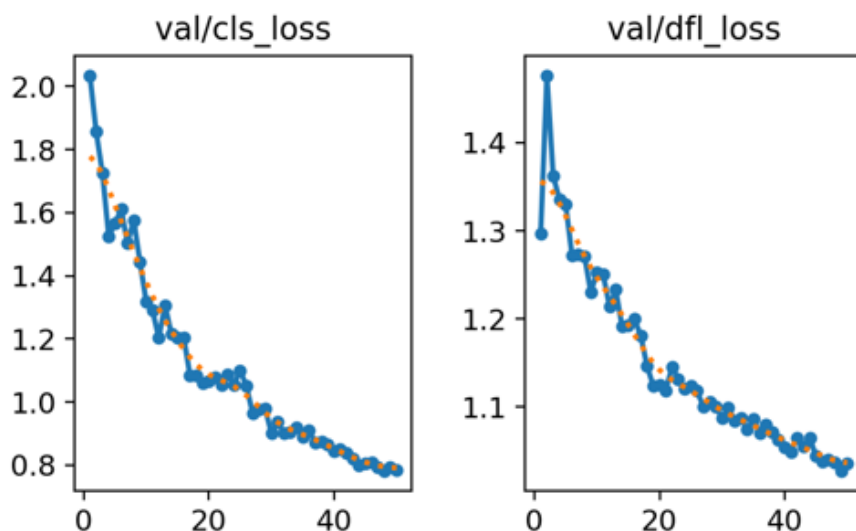
que el modelo va aprendiendo y comete menos errores, lo cual es una señal positiva de aprendizaje progresivo.

Para la gráfica *val/df_l_loss* que representa la pérdida de distribución de distancia es muy común para el modelo que se está ejecutando, como lo es Yolov8 especialmente cuando se usan representaciones discretas; es decir mide que tan bien el modelo predice la posición de objetos dentro de la imagen, en el mismo se puede ver aproximadamente como disminuye de 1.45 a 1.0, lo que indica que el modelo cada vez ubica mejor los objetos en la imagen.

Cabe mencionar que estos valores no representan porcentajes, sino que corresponden a medidas de error. Mientras más bajos sean, mejor desempeño indica el modelo. Estas métricas se utilizan para evaluar si el modelo está aprendiendo de manera adecuada durante el entrenamiento

Figura 67

Gráfica de Pérdida



Nota. Autoría Propia

De las dos gráficas anteriores se observa que la precisión aumenta y la pérdida disminuye en función de los ciclos realizados, tanto para el conjunto de entrenamiento como para el de validación, lo que significa que el modelo cometió pocos errores en algunos datos. Además, al utilizar la arquitectura de red neuronal Inception V3 se logra reducir Overfitting o sobreajuste del modelo, ya que en la gráfica se observa que la curva es similar para los datos de entrenamiento y validación. En caso de que la pérdida disminuyera en los datos de entrenamiento y no en los de validación el modelo estaría aprendiendo demasiado de los datos de entrenamiento.

3.6.5.7 Matriz de Confusión

En inteligencia artificial y aprendizaje automático, se propone el uso de matrices de confusión como herramienta de visualización del rendimiento del algoritmo utilizado, de forma que se puedan conocer la cantidad de éxitos y fracasos del modelo planteado a medida que avanza en el proceso de aprendizaje. Esta matriz consta de filas que representan los valores reales y las columnas representan los valores de predicciones.

En este caso, al tratar con dos etiquetas “Good leaf” y “Bad leaf”, se trata de una matriz binaria y surgen cuatro casos:

- Verdadero positivo (VP): número de casos en donde la pertenece a la clase Good leaf y el modelo lo clasificó como Good leaf.
- Verdadero negativo (VN): número de casos en donde la hoja pertenece a la clase Bad leaf y el modelo lo clasificó como no Bad leaf.
- Falso negativo (FN): número de casos en donde la hoja pertenece a la clase Good leaf y el modelo lo clasificó como Bad leaf.
- Falso positivo (FP): número de casos en donde la hoja

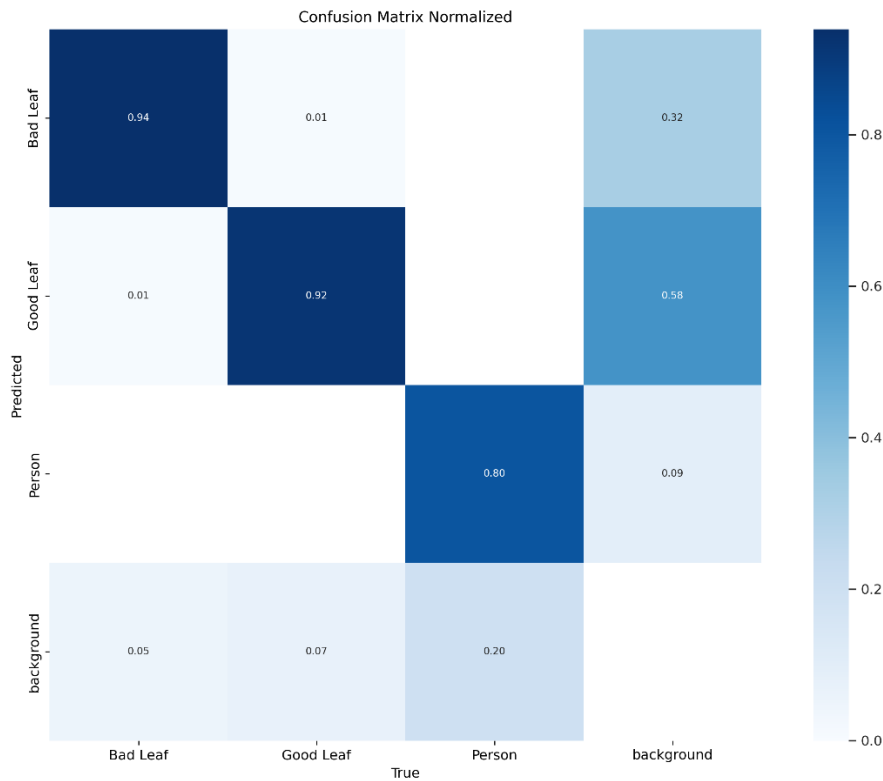
pertenece a la clase Bad leaf y el modelo lo clasificó como Good leaf.

En la figura 68 se establece la matriz de confusión, la cual consta de etiquetas reales y etiquetas predichas. El conjunto de datos puestos a prueba es de 871 muestras de las cuales 558 pertenecen a la etiqueta Good leaf, mientras que 313 pertenecen a la etiqueta Bad leaf. En la matriz de confusión se observa que para la etiqueta aptas existen 546 muestras acertadas, mientras que para la etiqueta no apta existen 300 muestras acertadas.

En la matriz de confusión se puede observar la inclusión de la clase "Person", la cual fue agregada con el objetivo de minimizar errores de detección, como falsos positivos o falsos negativos, cuando una persona aparece frente a la cámara. Esta estrategia mejora la eficiencia y confiabilidad del sistema, al reducir interferencias no deseadas durante el proceso de reconocimiento.

Figura 68

Matriz de Confusión del Modelo Entrenado



Nota. Autoría Propia

En la figura 69 se muestra un reporte de clasificación como métrica para evaluar el rendimiento del modelo entrenado. Los parámetros evaluados son:

- **Precisión:** métrica para conocer el porcentaje de casos positivos detectados. Para ello se realiza el cálculo de verdaderos positivos dividido entre los resultados positivos tomando en cuenta verdaderos positivos y falsos positivos. El valor de precisión del modelo entrenado es de 97%.

$$P = \frac{VP}{VP + FP} \times 100$$

- **Recuperación(recall):** es la relación entre los verdaderos positivos y la suma de verdaderos positivos y falsos negativos.

$$R = \frac{VP}{VP + FN} \times 100$$

Figura 69

Valores Finales de Precisión, Recall y Fitness del Modelo.

```
results_dict: {'metrics/precision(B)': 0.8974252745563462, 'metrics/recall(B)': 0.9117288054886661, 0.8893441176161214, 'fitness': 0.8893441176161214}
```

3.6.7 Diseño del Backend

En la figura 70 se puede ver como en este fragmento de código se crea una clase que utiliza ffmpeg como subproceso para capturar video desde una cámara RTSP. El flujo es decodificado y transformado a un formato crudo (rawvideo) en tiempo real, permitiendo su posterior procesamiento y transmisión. Esta etapa es fundamental, ya que actúa como puente

entre la cámara IP y el sistema WebRTC, facilitando la integración de hardware convencional con tecnologías de transmisión modernas.

Figura 70

Captura y Transmisión de Video Mediante WebRTC y Ffmpeg

```
class RTSPVideoTrack(VideoStreamTrack):
    def __init__(self, rtsp_url, width=640, height=360):
        super().__init__()
        self.width = width
        self.height = height
        self.frame_size = self.width * self.height * 3 # bgr24
        self.rtsp_url = rtsp_url
        self.pipe = subprocess.Popen([
            'ffmpeg',
            '-rtsp_transport', 'udp',
            '-i', self.rtsp_url,
            '-loglevel', 'quiet',
            '-an',
            '-f', 'rawvideo',
            '-pix_fmt', 'bgr24',
            '-vcodec', 'rawvideo',
            '-'
        ], stdout=subprocess.PIPE, stderr=subprocess.DEVNULL, bufsize=10**8)
```

Nota. Autoría Propia

En el método `recv`, se recupera cada fotograma del subprocesso de FFmpeg, se convierte en un arreglo NumPy y posteriormente en un VideoFrame, que es el formato aceptado por aiortc. Este paso es crucial para la conversión eficiente de datos de video, asegurando una transmisión fluida y con baja latencia hacia el cliente WebRTC como se indica en la figura 71.

Figura 71*Transformación de Cada Frame para ser Enviado por WebRTC*

```

async def recv(self):
    pts, time_base = await self.next_timestamp()

    raw_frame = await asyncio.get_event_loop().run_in_executor(
        None,
        self.pipe.stdout.read,
        self.frame_size
    )

    if len(raw_frame) != self.frame_size:
        raise Exception("Error leyendo frame RTSP")

    frame = np.frombuffer(raw_frame, np.uint8).reshape((self.height, self.width, 3))
    video_frame = VideoFrame.from_ndarray(frame, format='bgr24')
    video_frame.pts = pts
    video_frame.time_base = time_base
    return video_frame

```

Nota. Autoría Propia

En la figura 72 se puede ver como el cliente AsyncRTSPClient establece una conexión TCP con la cámara IP mediante el protocolo RTSP. Posteriormente, envía comandos como OPTIONS para verificar la compatibilidad y disponibilidad del dispositivo. Esta clase facilita la comunicación directa con la cámara y permite la automatización de tareas de control y mantenimiento de la conexión.

Figura 72*Conexión del Cliente AsyncRTSPClient*

```

class AsyncRTSPClient:
    global ip_cam
    def __init__(self, ip_cam, server_port=554):
        self.server_ip = ip_cam
        self.server_port = server_port
        self.reader = None
        self.writer = None
        self.cseq = 1
        self.keep_alive = True

    async def connect(self):
        try:
            self.reader, self.writer = await asyncio.wait_for(
                asyncio.open_connection(self.server_ip, self.server_port),
                timeout=3.0
            )
            print(f"Conectado a {self.server_ip}:{self.server_port}")
            return True
        except asyncio.TimeoutError:
            print("[RSTP] ❌ Tiempo de conexión agotado.")
        except Exception as e:
            print(f"[RSTP] ❌ Error de conexión: {e}")
            return False

```

Nota. Autoría Propia

3.6.7.1 Aplicación WebRTC con Procesamiento de Video y Detección de Objetos

Se importan las bibliotecas necesarias para la ejecución del servidor: comunicación WebRTC, inferencia con YOLO, transmisión de video, servidor web con aiohttp y operaciones con OpenCV como se muestra en la figura 73. Además, se inicializa el modelo de detección de objetos con una red neuronal previamente entrenada (best.pt). El uso de ultralytics. YOLO permite inferencias en tiempo real con alta precisión en objetos definidos por el usuario "hoja buena", "hoja mala".

Figura 73

Carga de Dependencias, Configuración y Variables Globales

```
import asyncio
import os
from aiohttp import web
from aiortc import RTCConfiguration, RTCPeerConnection,
import aiohttp cors
import cv2
from ultralytics import YOLO
import time
from av import VideoFrame
import uuid
```

Nota. Autoría Propia

En la figura 74 se muestra una clase que representa una pista de video que será enviada a los clientes WebRTC. Su propósito es transmitir el resultado del procesamiento de IA a los usuarios conectados. A través de la herencia de VideoStreamTrack, se pueden construir flujos de video personalizados para cada cliente.

Figura 74*Clase De Track De Video Procesado*

```

class ProcessedVideoTrack(VideoStreamTrack):
    kind = "video"

    def __init__(self):
        super().__init__()

    async def recv(self):
        global frame_to_show

        pts, time_base = await self.next_timestamp()

        while frame_to_show is None:
            await asyncio.sleep(0.01)

        frame_rgb = cv2.cvtColor(frame_to_show, cv2.COLOR_BGR2RGB)
        video_frame = VideoFrame.from_ndarray(frame_rgb, format="rgb24")
        video_frame.pts = pts
        video_frame.time_base = time_base

        return video_frame

```

Nota. Autoría Propia

En la figura 75 se muestra una función en la cual se ejecuta la inferencia del modelo YOLO sobre los frames capturados. Los resultados se traducen en coordenadas de cajas (boxes) y etiquetas (class_ids). Se aplican filtros por clase detectada, asignando colores distintos según el tipo de objeto (persona, hoja buena, hoja mala). El uso de `asyncio.to_thread()` permite no bloquear el evento loop principal.

Figura 75*Procesamiento de IA sobre los Frames Recibidos*

```

async def processFrame():
    global latest_frame, latest_boxes, ia_activada, draw_person_boxes

    while True:
        if latest_frame is not None and ia_activada:
            start_time = time.perf_counter()

            frame_copy = latest_frame.copy()

            # Ejecuta la inferencia de forma asincrona
            results = await asyncio.to_thread(model.predict, frame_copy)
            boxes = results[0].boxes.xyxy.cpu().numpy()
            class_ids = results[0].boxes.cls.cpu().numpy()
            names = model.names

            new_boxes = []

            for box, cls_id in zip(boxes, class_ids):
                x1, y1, x2, y2 = box

                cls_name = names[int(cls_id)].lower()
                if cls_name == 'bad leaf':
                    color = (0, 0, 255)
                elif cls_name == 'good leaf':
                    color = (0, 255, 0)
                elif cls_name == 'person' and draw_person_boxes:
                    color = (255, 255, 0)
                else:
                    continue

                new_boxes.append(((x1, y1, x2, y2), cls_name, color))

            latest_boxes = new_boxes

```

Nota. Autoría Propia

En main(), se levantan las tareas principales del sistema: la visualización, el procesamiento con IA y el servidor HTTP como se muestra en la figura 76. Estas corren de forma concurrente usando asyncio. El sistema queda en espera indefinida, asegurando disponibilidad continua.

Figura 76*Inicialización del Servidor Web y Ciclo Principal*

```
async def main():
    # Inicia el proceso paralelo
    asyncio.create_task(viewFrame())
    asyncio.create_task(processFrame())

    sessions["processed_track"] = ProcessedVideoTrack()

    runner = web.AppRunner(app)
    await runner.setup()
    site = web.TCPSite(runner, port=80)
    await site.start()

    print("\nServidor iniciado en puerto 80")

    # Mantener el servidor corriendo
    while True:
        await asyncio.sleep(3600)

if __name__ == "__main__":
    asyncio.run(main())
```

Nota. Autoría Propia

3.7 Integración del Sistema

Se desplegó una instancia de máquina virtual (VM) en Google Cloud Platform, ubicada en la zona us-central1-a. La instancia, identificada con el nombre automático instance-20250204-100818, se encuentra actualmente en estado activo, lo que indica que está operativa y lista para recibir conexiones.

La máquina dispone de una IP interna como se muestra en la figura 81 10.128.0.7 utilizada para la comunicación dentro de la red privada del proyecto, y una IP externa pública 34.27.253.244, que permite el acceso remoto desde el exterior. Para la conexión remota y gestión de la instancia, se habilitó el acceso mediante SSH directamente desde la consola de Google Cloud como se muestra en la figura 77. Esta configuración permite desplegar y ejecutar servicios o aplicaciones de manera remota sobre la infraestructura en la nube, facilitando la escalabilidad, disponibilidad y administración del sistema.

Figura 77

Instancia de Máquina Virtual del Servidor Web

Instancias de VM

Crear instancia

Importar VM

Actualizar

Más información

Instancias

Observabilidad

Programas de las instancias

Instancias de VM

Filtro

Escribir el nombre o valor de la propiedad

☐	Estado	Nombre ↑	Zona	Recomendaciones	En uso por	IP interna	IP externa	Conectar
☒		instance-20250204-100818	us-central1-a			10.128.0.7 (nic0)	34.27.253.244 (nic0)	SSH

Acciones relacionadas

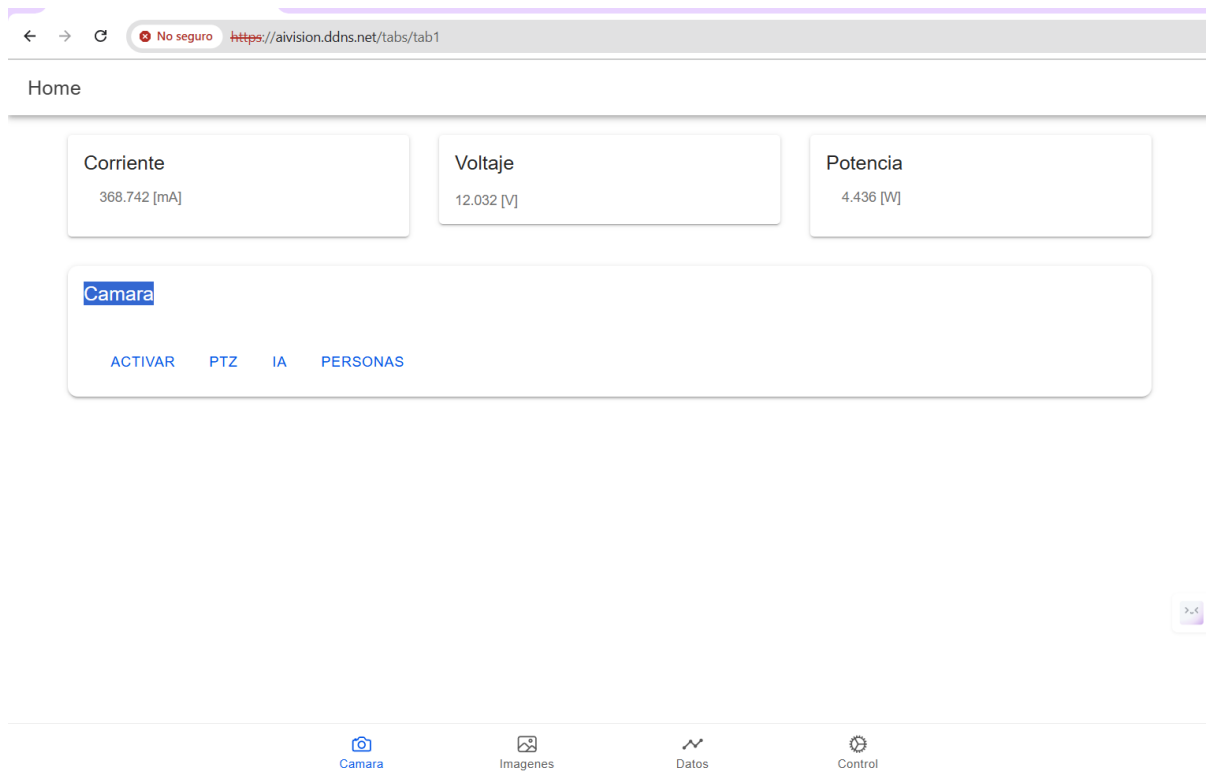
Hide

Nota. Autoría Propia

El sistema dispuso de una interfaz web como se muestra en la figura 78 compuesta por cuatro secciones principales en la parte inferior de la página. En la primera sección se integró un módulo de visualización de la cámara, el cual permitía su activación y desactivación. Asimismo, se implementó un control PTZ (Pan-Tilt-Zoom) que facilitaba el movimiento direccional de la cámara. La segunda sección incluyó un componente de inteligencia artificial, diseñado para su activación mediante un botón específico, el cual ponía en marcha los algoritmos previamente programados. Finalmente, se añadió un botón de detección de personas que permitía excluir del procesamiento a los individuos que abandonaban el campo de visión de la cámara.

Figura 78

Interfaz Gráfica



Nota. Autoría Propia

En la siguiente sección de la página web como se muestra en la figura 79 se incorporó un botón destinado a la visualización de imágenes, el cual permitía acceder a capturas que contenían hojas defectuosas. Además, se implementó un control de umbral ajustable, que permitía modificar el nivel de sensibilidad en la detección de imperfecciones, determinando así qué imágenes debían almacenarse en la base de datos de Firebase.

Figura 79*Interfaz Visualización de Detecciones*

Nota. Autoría Propia

En la siguiente sección como muestra la figura 80 de la página web se implementó un módulo de monitoreo del estado de la batería, el cual permitía al usuario seleccionar un rango de fechas definido por una fecha de inicio y una de fin con el objetivo de consultar el historial de consumo energético del sistema. Este historial se genera automáticamente cuando se detecta variaciones iguales o superiores a 0.1 voltios en el voltaje de la batería. La funcionalidad ofrece una visualización detallada de los eventos registrados, lo que facilita el análisis del comportamiento energético a lo largo del tiempo y permite identificar posibles anomalías o patrones de descarga inusuales. Los datos recopilados son almacenados para su posterior análisis y trazabilidad.

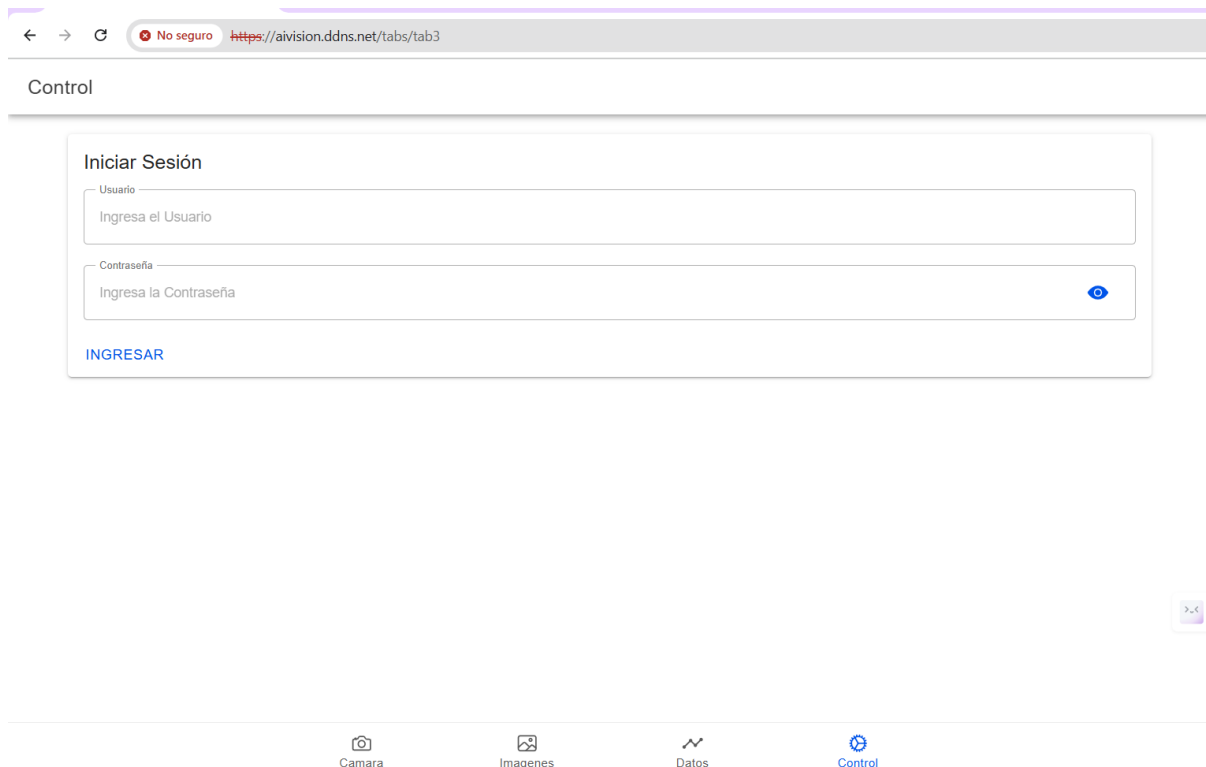
Figura 80*Datos de Control de la Batería*

Nota. Autoría Propia

En la cuarta sección de la interfaz web como se muestra en la figura 81 se encuentra un botón de control que permite el acceso a un panel administrativo. A través de este módulo, los usuarios con privilegios autorizados pueden modificar parámetros clave del sistema, ajustando el comportamiento de variables operativas según las necesidades del entorno o del funcionamiento esperado. Esta funcionalidad es esencial para la configuración dinámica del sistema, ya que facilita la calibración y personalización de distintos componentes sin necesidad de intervenir directamente en el código fuente.

Figura 81

Ingreso al Control



The screenshot shows a web browser window with the address bar displaying "No seguro https://aivision.ddns.net/tabs/tab3". The page title is "Control". The main content area features a login form titled "Iniciar Sesión". This form contains two input fields: "Usuario" with the placeholder text "Ingresa el Usuario" and "Contraseña" with the placeholder text "Ingresa la Contraseña". A blue eye icon is visible next to the password field. Below the fields is a blue button labeled "INGRESAR". At the bottom of the page, there is a navigation bar with four icons: a camera icon labeled "Camara", a picture icon labeled "Imagenes", a line graph icon labeled "Datos", and a gear icon labeled "Control", which is currently selected.

Nota. Autoría Propia

Dentro del módulo de control de cámara como muestra la figura 82 se encuentra una sección de configuración basada en el protocolo MQTT, la cual incluye campos para definir el *Hostname*, el puerto de comunicación, el *path* y el *topic* correspondiente. Esta configuración permite establecer la conexión con el servidor de mensajería para la transmisión de comandos. Posteriormente, el usuario puede seleccionar entre encender o apagar la cámara mediante un control específico. Finalmente, se incorpora una opción de *sleep*, que permite definir el intervalo de tiempo durante el cual la cámara permanece apagada antes de reactivarse automáticamente.

Figura 82

Configuración de Control

The screenshot shows a web browser at the URL <https://aivision.ddns.net/tabs/tab3>. The page title is 'Control'. It features three main configuration sections:

- Configuración MQTT:** Contains four input fields: 'Broker Hostname' (broker.emqx.io), 'Port' (8084), 'Path' (/mqtt), and 'Tema' (jcoder/#). Below these are two buttons: 'CONECTAR' and 'GUARDAR CONFIGURACIÓN'.
- Camara:** Contains a single button labeled 'ON/OF'.
- Sleep:** Contains a 'Tiempo [S]' input field with the value 60, and an 'ENVIAR' button below it.

At the bottom of the page is a navigation bar with four icons and labels: 'Camara', 'Imagenes', 'Datos', and 'Control' (which is highlighted in blue).

Nota. Autoría Propia

En la Figura 83 se observa el sistema completamente integrado, compuesto por una cámara instalada en un poste de madera y un cajón metálico que alberga los componentes principales. En su interior se encuentran la Raspberry Pi y el nodo central, el cual incluye un panel solar y una batería de motocicleta utilizados para el desarrollo de este trabajo de tesis.

Figura 83

Nodo Integrado en la Parcela



Nota. Autoría Propia

En la Figura 84 se muestra la división estratégica empleada para la evaluación comparativa de la incidencia de *Botrytis cinerea* bajo diferentes condiciones de exposición solar y microclima. A continuación, se presenta la justificación de la selección de la parcela utilizada para la implementación de este trabajo de grado:

Justificación de la ubicación

- ✓ **Condiciones agroclimáticas óptimas:** La zona presenta rangos de temperatura y humedad adecuados tanto para el cultivo de granadilla como para el desarrollo de *Botrytis cinerea*.
- ✓ **Prevalencia endémica:** Se registra una incidencia histórica de la enfermedad en el sector, lo que permite un estudio representativo.
- ✓ **Representatividad regional:** Corresponde a una zona productora característica de la región andina ecuatoriana.

- ✓ **Accesibilidad logística:** El área cuenta con energía eléctrica y acceso vehicular, facilitando la instalación y el monitoreo del sistema.
- ✓ **Infraestructura existente:** El sistema de emparrado del cultivo ofrece condiciones adecuadas para la instalación de las cámaras y demás equipos requeridos.

Figura 84

División del Área Experimental



Nota. Autoría Propia

3.7.1 Foto Aérea de los Tres Bloques a Analizar

En la Figura 85 se observa el bloque compacto correspondiente a la división del área experimental en tres bloques. El primer bloque se dejó con una mayor presencia de hojas, con el objetivo de establecer una diferencia visible dentro del estudio comparativo y analizar cómo esta condición influye en los resultados obtenidos. Por su parte, los dos bloques restantes se

utilizaron para evaluar el efecto de la Botrytis, ya que presentan una mayor carga de la enfermedad, evidenciándose un índice más elevado de hojas afectadas. Esta disposición permite comparar el comportamiento del sistema bajo diferentes condiciones de severidad de la enfermedad.

Figura 85

Foto Aérea de los 3 Bloques



Nota. Autoría Propia

CAPÍTULO IV: Pruebas de Funcionamiento

En el cuarto capítulo se lleva a cabo la implementación del sistema propuesto considerando los dos escenarios planteados exponiendo así la cuarta fase de la metodología Bottom-Up, que es centrada en la verificación integral del sistema desarrollado. Además, se realizan pruebas de funcionamiento del sistema de detección temprana de *Botrytis cinerea* en hojas de granadilla para verificar el cumplimiento de los requerimientos definidos en el proyecto. Asimismo, se presentan las conclusiones basadas en los resultados obtenidos y se proponen recomendaciones que pueden ser útiles para futuras investigaciones o mejoras del sistema.

4.1 Cronograma de Pruebas de Funcionamiento del Sistema

El sistema comienza con la adquisición de imágenes de las hojas de granadilla. Posteriormente, se ejecuta el modelo preentrenado en la placa embebida Raspberry. Este modelo preentrenado se obtiene mediante el software Google Colab, ya que, aunque la Raspberry Pi 4 puede ejecutar códigos relacionados con visión artificial, no es adecuada para entrenar la red debido a sus limitaciones de hardware. En esta etapa, se aprovechan los mapas de características, las operaciones de entrenamiento del modelo preentrenado y las clases previamente establecidas.

Finalmente, una vez procesada la información, el usuario puede acceder a los resultados de la clasificación de las hojas (buenas, malas o mixtas) mediante una interfaz intuitiva y amigable.

4.2 Pruebas funcionales

En esta sección se presentan los resultados obtenidos tras la implementación del sistema inteligente para la detección temprana de Botrytis cinerea en hojas de granadilla, empleando visión artificial. A continuación, se detallan las pruebas realizadas en este bloque:

- **Prueba de clasificación de lotes:** usando imágenes de escenarios completos, como los lotes, se procesan las imágenes para identificar y cuantificar las hojas categorizadas en los estados mencionados anteriormente, brindando una visión global del estado del cultivo como lo muestra la Figura 86.

Figura 86

Prueba de clasificación individual – Bad Leaf en un Umbral de 1



Nota. Autoría Propia

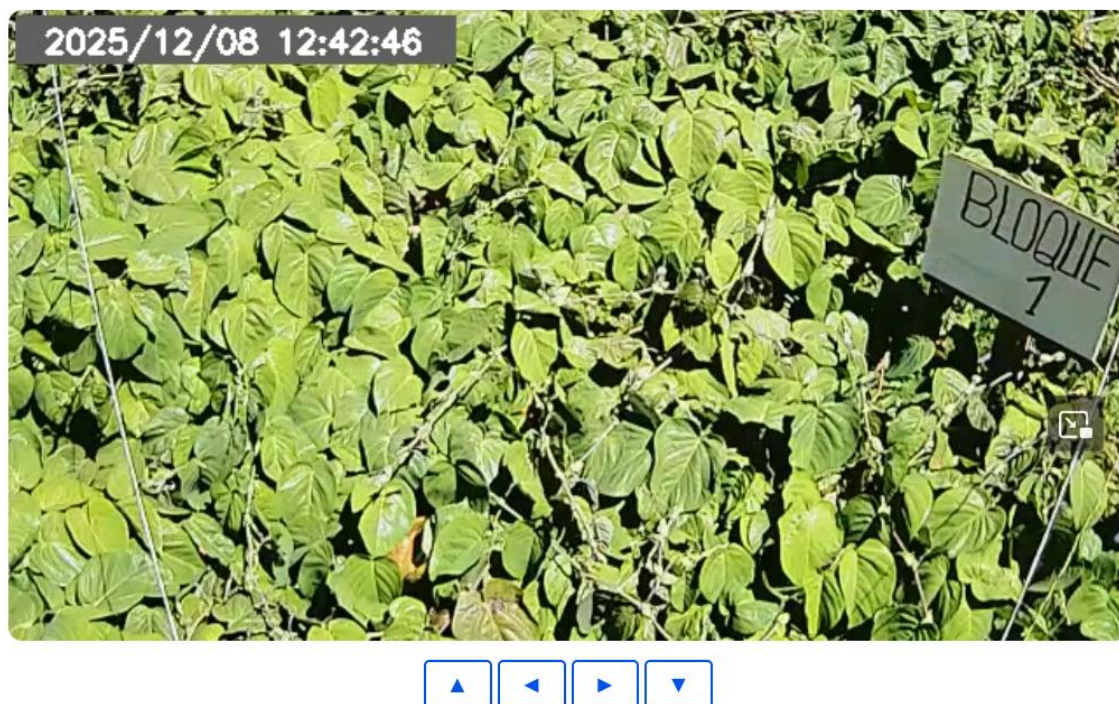
4.2.2. Prueba de clasificación de bloques

Para visualizar los resultados de las pruebas, se utiliza una pantalla. La visualización se realiza a través de video en tiempo real.

El primer bloque se caracterizó por presentar dimensiones considerablemente amplias, lo que permitió definirlo como la sección inicial y más representativa del espacio estudiado. Este bloque tuvo un largo de 5,92 metros y un ancho de 5,79 metros, medidas que evidenciaron una geometría casi cuadrada y una distribución espacial equilibrada. Durante el proceso de análisis, se observó que estas proporciones ofrecían una base sólida para la organización general del proyecto, ya que su tamaño facilitaba la disposición de elementos estructurales y funcionales. Además, dichas dimensiones permitieron evaluar con precisión la relación entre superficie útil y límites perimetrales, lo que resultó fundamental para comprender el comportamiento global del área construida. En consecuencia, el primer bloque se interpretó como la unidad fundamental sobre la cual se articularon las demás secciones, aportando estabilidad y coherencia al conjunto como se muestra en la Figura 87.

Figura 87

Bloque 1

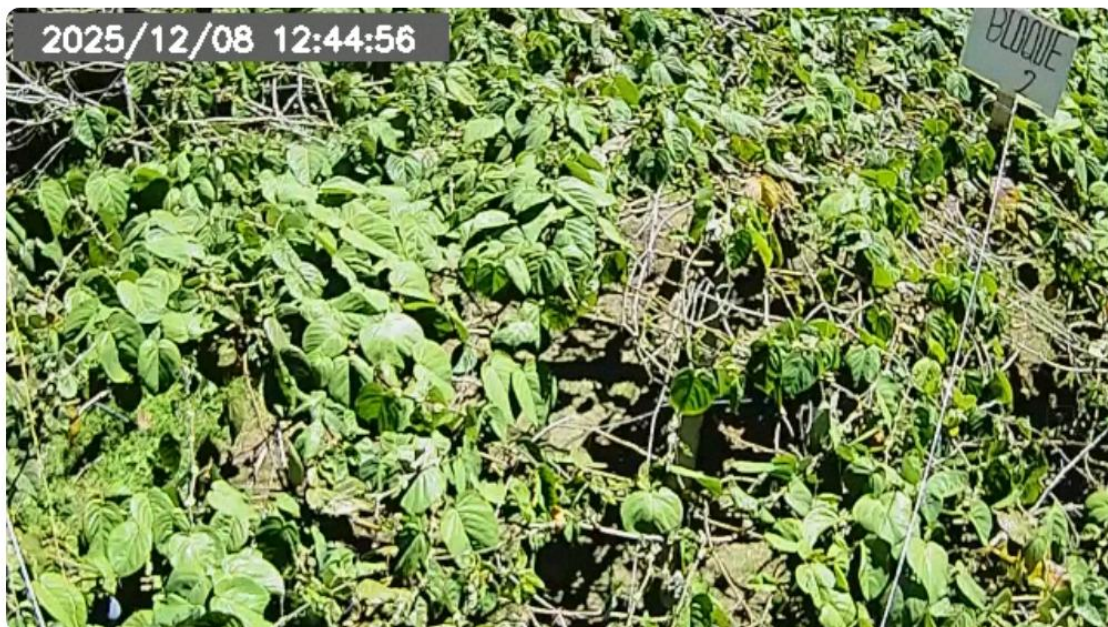


Nota. Autoría Propia

El segundo bloque como se muestra en la Figura 88 presentó variaciones importantes en relación con el primero, particularmente en términos de proporción y distribución espacial. Este segmento registró un largo de 5,85 metros y un ancho de 4,50 metros, lo que evidenció un diseño más alargado que el bloque inicial. Durante la etapa de evaluación, se determinó que esta configuración respondía a necesidades específicas del proyecto, tanto funcionales como estructurales. Las dimensiones del segundo bloque influyeron directamente en la manera en que se integraron los accesos, la circulación interna y la conectividad con las otras áreas. Asimismo, su forma más estrecha permitió analizar cómo los cambios en la morfología de los espacios modificaban la dinámica interna del conjunto arquitectónico. De esta forma, el segundo bloque se interpretó como una zona de transición que articulaba los diferentes sectores sin comprometer la continuidad espacial del diseño.

Figura 88

Bloque 2

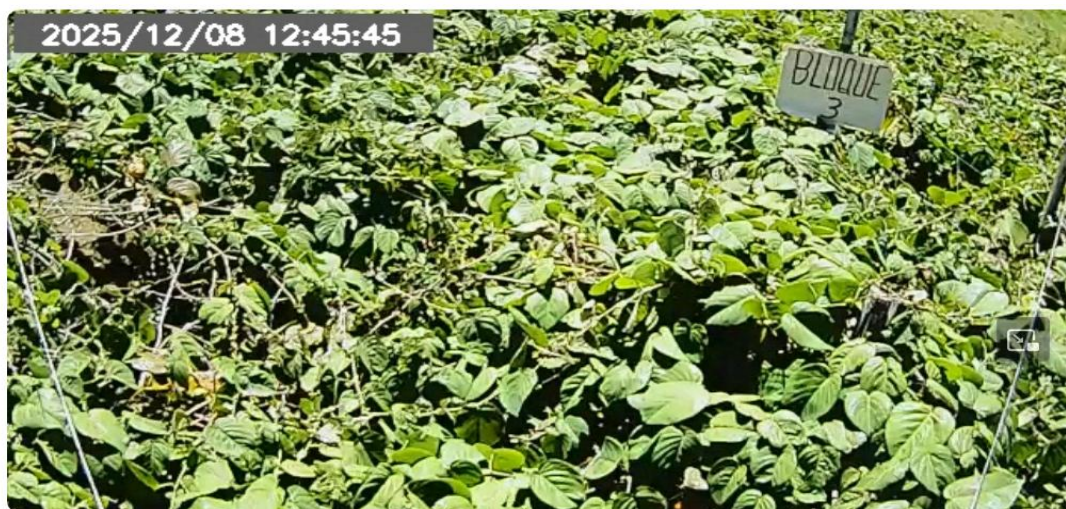


Nota. Autoría Propia

El tercer bloque, el de menores dimensiones dentro del conjunto, cumplió un papel complementario pero esencial en la configuración total del proyecto. Este bloque presentó un largo de 4,12 metros y un ancho de 3,95 metros, medidas que lo definieron como un espacio compacto y de uso más específico como lo muestra la Figura 89. A lo largo del análisis, se identificó que sus proporciones permitían una organización eficiente de actividades o elementos que requerían un área reducida pero bien delimitada. La presencia de este bloque también contribuyó a equilibrar el diseño general, proporcionando variación volumétrica y funcional dentro del conjunto arquitectónico. Su tamaño más contenido ofreció la oportunidad de estudiar la adaptación de recursos y materiales en espacios pequeños, lo que generó conclusiones valiosas respecto a la optimización del espacio y el rendimiento constructivo. En síntesis, el tercer bloque funcionó como una unidad estratégica que reforzó la funcionalidad global del proyecto.

Figura 89

Bloque 3



Nota. Autoría Propia

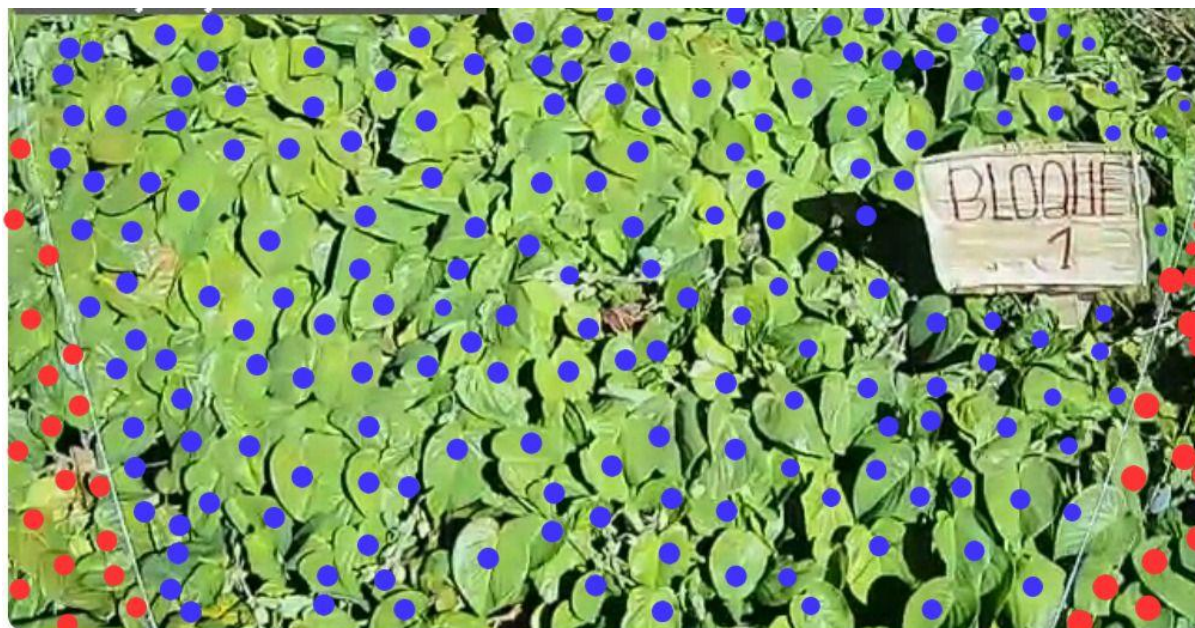
4.2.3 Conteo Manual Técnico Realizado Para el Estudio

De manera global, y a partir de un conteo manual realizado hoja por hoja, considerando los tres bloques evaluados, se registró un total de 726 hojas. De este conjunto, 555 hojas fueron seleccionadas manualmente para el análisis del estudio, mientras que 171 hojas, aunque presentes en el área de evaluación, no fueron consideradas. En términos porcentuales, el 76,45 % de las hojas contabilizadas manualmente fueron incluidas en el análisis, mientras que el 23,55 % fueron excluidas. Estos resultados permiten establecer una base de datos sólida y representativa, adecuada para el desarrollo del análisis comparativo propuesto.

En el caso del primer bloque, el conteo manual permitió registrar un total de 201 hojas. De estas, 165 hojas fueron consideradas aptas para el análisis de imagen, mientras que las 36 hojas restantes, si bien fueron visualmente identificadas durante la inspección manual, no cumplieron con los criterios establecidos para su inclusión en el estudio. En términos porcentuales, las hojas seleccionadas manualmente representan el 82,05 % del total, mientras que el 17,91 % corresponde a hojas descartadas. Este proceso de selección manual se ilustra en la Figura 90 y fue realizado con el objetivo de garantizar la calidad, consistencia y confiabilidad de los datos evaluados.

Figura 90

Conteo de hojas en el Bloque 1



Nota. Autoría Propia

Para este bloque, tal como se muestra en la Figura 91, se realizó un conteo manual de un total de 229 hojas. De este conjunto, 179 hojas fueron seleccionadas manualmente para ser incluidas en el análisis, mientras que las 50 hojas restantes, aunque identificadas durante la inspección visual manual, no se incorporaron al estudio. En términos porcentuales, las hojas consideradas representan aproximadamente el 78,17 % del total contabilizado manualmente, mientras que el 21,83 % corresponde a aquellas que fueron excluidas del análisis. Esta diferenciación, obtenida mediante un proceso de selección manual, permite delimitar de manera precisa y confiable el conjunto de datos finalmente evaluados.

Figura 91

Conteo de hojas en el Bloque 2



Nota. Autoría Propia

En el bloque 3, tal como se muestra en la Figura 92, se llevó a cabo un conteo manual que permitió registrar un total de 296 hojas. De estas, 211 hojas fueron seleccionadas manualmente para su inclusión en el análisis, mientras que 85 hojas fueron descartadas al no cumplir con las condiciones necesarias para su evaluación. En términos porcentuales, las hojas incluidas representan el 71,28 % del total contabilizado manualmente, mientras que el 28,72 % corresponde a hojas no consideradas. Esta proporción evidencia una mayor presencia de hojas no aptas en comparación con los bloques evaluados anteriormente, lo que sugiere un mayor nivel de afectación en este bloque.

Figura 92*Conteo de hojas en el Bloque 3*

Nota. Autoría Propia

4.2.4 Comparación de los Bloques Estudiados

A partir de la información presentada en la Tabla 16, se evidencia una variación progresiva en la cantidad y proporción de hojas consideradas para el análisis entre los tres bloques evaluados. Cabe señalar que este conteo fue realizado de manera manual, mediante una revisión visual de cada hoja, lo que permitió discriminar aquellas aptas para el estudio según los criterios establecidos.

El bloque 1 presenta el mayor porcentaje de hojas incluidas, con un 82,05 %, lo que indica una condición más favorable para el análisis y una menor presencia de hojas descartadas. En el bloque 2, el porcentaje de hojas consideradas disminuye a 78,17 %, lo que refleja un incremento moderado en el número de hojas excluidas. Finalmente, el bloque 3 registra el menor porcentaje de hojas analizadas, con un 71,28 %, evidenciando una mayor incidencia de

hojas no consideradas, posiblemente asociada a un mayor nivel de afectación o deterioro fitosanitario.

En términos generales, de un total de 726 hojas evaluadas manualmente, el 76,45 % fue utilizado en el estudio. Este resultado garantiza la conformación de una base de datos representativa para el análisis comparativo, al mismo tiempo que pone de manifiesto diferencias claras entre los bloques evaluados en función de su estado fitosanitario.

Tabla 16

Conteo de Hojas que Serán Tomadas en Cuenta por Bloques

Bloque	Hojas totales (conteo manual)	Hojas consideradas	Hojas no consideradas	% consideradas	% no consideradas
Bloque 1	201	165	36	82.05%	17.91%
Bloque 2	229	179	50	78.17%	21.83%
Bloque 2	296	211	85	71.28%	28.72%
Total	726	555	171	76.45%	23.55%

4.3 Detección Temprana de Hojas Malas en los bloques

En la imagen correspondiente al Bloque 1 de la Figura 93 se observa un alto nivel de detección de hojas sanas, identificadas y delimitadas mediante cajas delimitadoras de color verde con la etiqueta “good leaf”, lo que evidencia un desempeño adecuado del sistema de visión artificial en un entorno de alta densidad foliar. Visualmente, la mayoría de las hojas presentes en la escena han sido correctamente reconocidas por el modelo, lo cual es coherente con el elevado porcentaje de hojas consideradas en este bloque, mientras que un número reducido de hojas no presenta detección, principalmente debido a superposición, variaciones

de tamaño o posiciones desfavorables dentro de la imagen. Las hojas detectadas muestran características homogéneas de sanidad, como coloración verde uniforme y ausencia de síntomas visibles, lo que refuerza la confiabilidad del sistema para establecer una referencia de hojas sanas. En conjunto, la imagen refleja una cobertura mayoritaria del cultivo por parte del modelo, validando su aplicabilidad para el monitoreo automatizado y como base para la detección temprana de enfermedades como *Botrytis cinerea*.

Figura 93

Detección Temprana Vista en el Bloque 1



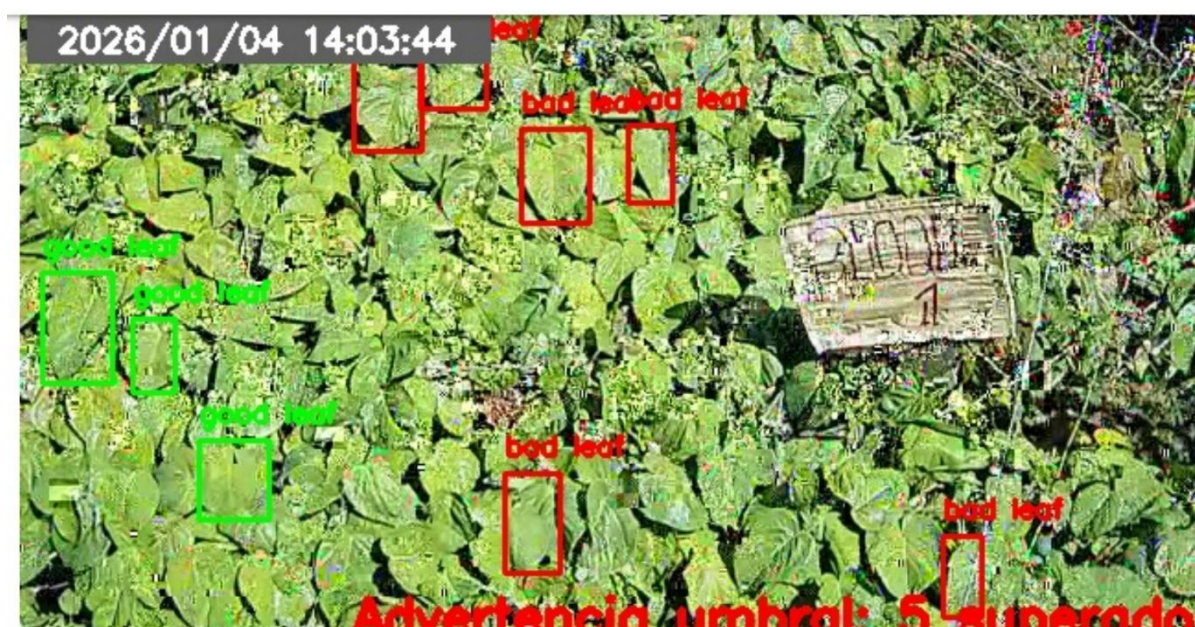
Nota. Autoría Propia

En la imagen correspondiente a la figura 94 de la semana siguiente del Bloque 1 se observa que, aunque la mayoría de las hojas continúan siendo clasificadas como sanas (“good leaf”), aparece un porcentaje claramente identificable de hojas afectadas (“bad leaf”) delimitadas en color rojo, concentradas principalmente en la zona central y derecha de la escena. De acuerdo con la densidad visual de las detecciones, aproximadamente entre un 15 % y 20 % de las hojas detectadas presentan condición de “bad leaf”, mientras que el 80 % a 85 %

restante corresponde a hojas sanas, lo que representa un incremento evidente de hojas en mal estado respecto a la semana anterior, donde predominaban casi en su totalidad las hojas sanas. Este cambio visual respalda la activación de la advertencia por superación de umbral mostrada en la imagen y evidencia una evolución progresiva del estrés o enfermedad en el cultivo, confirmando la utilidad del sistema para el seguimiento temporal y la detección temprana de afecciones como *Botrytis cinerea*.

Figura 94

Enfermedad Detectada en el Bloque 1



Nota. Autoría Propia

En la imagen correspondiente al Bloque 2 de la Figura 95 se observa una presencia más marcada de hojas clasificadas como “bad leaf”, delimitadas con cajas rojas y distribuidas principalmente en la zona central e inferior de la escena, lo que evidencia un deterioro más evidente del estado fitosanitario del cultivo en comparación con el Bloque 1.

Figura 95*Detección Temprana Vista en el Bloque 2*

Nota. Autoría Propia

A partir de la densidad visual de las detecciones, como se observa en la Figura 96, se estima que aproximadamente entre el 25 % y el 30 % de los elementos detectados fueron clasificados por el sistema como “bad leaf”, mientras que el 70 % a 75 % restante corresponde a hojas identificadas como sanas, lo que confirma la superación del umbral de advertencia indicado en la imagen. No obstante, durante el análisis detallado de los resultados se identificó que una de las detecciones clasificadas como hoja afectada corresponde en realidad al cartel de identificación del bloque, el cual presenta características visuales (forma rectangular, coloración contrastante y textura) que el modelo interpreta erróneamente como una hoja con daño. Esta confusión evidencia una limitación puntual del sistema asociada a la ausencia de elementos no vegetales en el conjunto de entrenamiento, mas no a una falla general del modelo. Aun considerando esta detección errónea, el incremento de hojas afectadas reales sugiere una progresión más avanzada del estrés o enfermedad, posiblemente relacionada con la expansión

de *Botrytis cinerea*, y demuestra la capacidad del sistema de visión artificial para identificar variaciones espaciales y temporales en el estado fitosanitario del cultivo, permitiendo un monitoreo diferenciado por bloques y apoyando la toma de decisiones agronómicas oportunas.

Figura 96

Enfermedad Detectada en el Bloque 2



Nota. Autoría Propia

En la imagen correspondiente al Bloque 3 de la figura 97 se observa una presencia claramente identificable de hojas clasificadas como “bad leaf”, delimitadas mediante cajas rojas y distribuidas de forma dispersa en la zona central y lateral de la escena. A partir del análisis visual de las detecciones, se estima que aproximadamente entre un 20 % y 25 % de las hojas detectadas presentan condición de “bad leaf”, mientras que el 75 % a 80 % restante corresponde a hojas que aún conservan un estado visualmente sano.

Figura 97*Detección Temprana Vista en el Bloque 3*

Nota. Autoría Propia

La superación del umbral de advertencia indicada en la imagen confirma que el nivel de afectación en el Bloque 3 es significativo, aunque ligeramente inferior al observado en el Bloque 2. Sin embargo, al realizar un análisis detallado de las detecciones mostradas en la Figura 98, se identifica que una de las clasificaciones correspondientes a “bad leaf” se asocia en realidad al cartel de identificación del bloque, el cual es interpretado erróneamente por el sistema como una hoja afectada debido a similitudes visuales en color, forma y contraste con el follaje circundante. Esta situación pone en evidencia una limitación puntual del modelo, relacionada con la ausencia de elementos no vegetales dentro del conjunto de entrenamiento, sin comprometer la capacidad general del sistema para detectar hojas con síntomas reales. Considerando esta observación, el comportamiento detectado sugiere una propagación moderada del deterioro foliar, lo que demuestra la capacidad del sistema de visión artificial para identificar variaciones espaciales en el estado fitosanitario del cultivo y respaldar un

monitoreo diferenciado por bloques, contribuyendo a la detección temprana de afecciones como *Botrytis cinerea*.

Figura 98

Enfermedad Detectada en el Bloque 3



Nota. Autoría Propia

4.4 Detección Visual de Hojas Malas en los Bloques realizado por el Agricultor

En el bloque 1, la señora Marcia Robles y el señor Ermel Robles, propietarios de la finca, realizaron un conteo manual visual de las hojas afectadas por *Botrytis cinerea*. Durante la inspección del bloque, los agricultores identificaron un total de 5 hojas consideradas como malas, basándose en la presencia de manchas visibles, cambios de coloración y signos leves de necrosis. Este conteo se realizó hoja por hoja, recorriendo el bloque completo, y refleja una percepción visual de baja afectación, coherente con las condiciones fitosanitarias más favorables observadas en este sector.

Para el bloque 2, el conteo manual visual efectuado por los propietarios de la finca permitió identificar un total de 8 hojas afectadas. En este caso, los agricultores observaron una mayor presencia de manchas marrones y zonas irregulares en comparación con el bloque 1, lo

que influyó en el incremento del número de hojas clasificadas como malas. El conteo se realizó de manera directa y secuencial, considerando únicamente los síntomas visibles al momento de la evaluación, lo que evidencia una afectación intermedia del cultivo según el criterio empírico del agricultor.

En el bloque 3, el conteo manual visual registró un total de 10 hojas afectadas por *Botrytis cinerea*. Durante la inspección, se identificaron hojas con lesiones más extensas, necrosis avanzada y una mayor alteración de la textura superficial, lo que facilitó su detección visual. Este mayor número de hojas consideradas como malas concuerda con un estado fitosanitario más desfavorable en este bloque, según la apreciación directa del agricultor durante el recorrido del área evaluada.

4.5 Análisis Comparativo Entre Bloques de la Parcela

En el Bloque 1, el sistema de visión artificial detectó un total de 6 hojas afectadas, valor muy cercano al conteo manual visual realizado por el agricultor, quien identificó 5 hojas malas. Esta similitud evidencia un buen desempeño del sistema bajo condiciones de manejo favorables, donde el riego tecnificado y la fumigación contribuyen a una menor presencia de síntomas y facilitan la correcta identificación de hojas afectadas.

En el Bloque 2, el sistema registró únicamente 4 hojas clasificadas como malas, mientras que el conteo manual visual del agricultor identificó 8 hojas afectadas. Esta discrepancia se debe a una limitación del sistema, ya que durante el proceso de detección se observó que el cartel identificador del bloque fue erróneamente clasificado como una hoja mala. Esta falsa detección influyó directamente en el conteo final, reduciendo la capacidad del sistema para identificar correctamente hojas afectadas reales en este bloque.

De manera similar, en el Bloque 3, el sistema detectó 4 hojas malas, mientras que el agricultor registró 10 hojas afectadas. Al igual que en el bloque anterior, se evidenció que el

sistema confundió el cartel identificador del bloque 3 con una hoja mala, lo que constituye un error de clasificación. Esta situación resulta especialmente relevante, ya que el Bloque 3 no contó con riego tecnificado ni fumigación, por lo que se esperaba una mayor detección de hojas afectadas. La confusión del cartel como objeto de interés pone de manifiesto la necesidad de mejorar el proceso de segmentación o exclusión de elementos ajenos al follaje.

Estos resultados evidencian en la tabla 17 que, si bien el sistema de visión artificial presenta un desempeño adecuado en escenarios controlados, aún existen limitaciones asociadas a la detección de objetos externos al cultivo, como carteles identificadores. No obstante, la identificación de este tipo de errores permite proponer mejoras futuras, tales como la incorporación de máscaras de exclusión, filtrado por forma o el entrenamiento del modelo con clases adicionales, fortaleciendo así la robustez del sistema.

Tabla 17

Comparación de los Bloques Tanto para Hojas Detectadas por el Sistema y Conteo Manual

Bloque	Condición es de manejo	Total de hojas evaluadas	Total de hojas detectadas por el sistema	Hojas malas detectadas por el sistema	Hojas malas detectadas por el agricultor	Observación clave
Bloque 1	Riego tecnificado	201	181	6	5	Alta concordancia entre sistema y agricultor
Bloque 2	Riego tecnificado	229	203	4	8	Falsa detección del

							cartel	del
							bloque	
Bloque	Sin	riego	296	256	4	10	Falsa	
3	tecnificado						detección del	
							cartel	del
							bloque	

Estos resultados evidencian que, si bien el sistema de visión artificial presenta un desempeño adecuado en escenarios controlados, aún existen limitaciones asociadas a la detección de objetos externos al cultivo, como carteles identificadores. No obstante, la identificación de este tipo de errores permite proponer mejoras futuras, tales como la incorporación de máscaras de exclusión, filtrado por forma o el entrenamiento del modelo con clases adicionales, fortaleciendo así la robustez del sistema.

4.6 Análisis de efectividad del sistema de visión artificial en condiciones reales de campo

Con el fin de evaluar cuantitativamente la efectividad del sistema de visión artificial desarrollado, se realizó una comparación entre el número total de hojas evaluadas manualmente y el número de hojas detectadas automáticamente por el sistema en cada uno de los bloques analizados.

En el Bloque 1, el sistema logró detectar 181 de 201 hojas evaluadas, alcanzando una efectividad del 90,05 %, lo que evidencia un desempeño elevado bajo condiciones de riego tecnificado.

En el Bloque 2, se detectaron 203 de 229 hojas, obteniendo una efectividad del 88,65 %, mientras que en el Bloque 3, sin riego tecnificado, se identificaron 256 de 296 hojas, con una efectividad del 86,49 %. Al promediar los resultados de los tres bloques, se obtiene una

efectividad global del sistema del 88,40 %, lo que demuestra una alta capacidad de detección de hojas en condiciones reales de campo.

Las variaciones observadas entre bloques se asocian principalmente a factores como la superposición de hojas, variaciones de iluminación y la presencia de elementos externos al cultivo, sin comprometer el desempeño general del sistema como muestra la tabla 18.

Tabla 18

Efectividad del Sistema De Visión Artificial por Bloque

Bloque	Total	hojas	Hojas	detectadas	Efectividad (%)
	evaluadas		por el sistema		
Bloque 1	201		181		90,05 %
Bloque 2	229		203		88,65 %
Bloque 3	296		256		86,49 %

$$Promedio\ general = \frac{90.05 + 88.65 + 86.49}{3}$$

$$Promedio\ general = 88.40\%$$

El análisis de los resultados evidencia que el Bloque 1 presenta el mayor nivel de efectividad del sistema, lo cual se asocia a condiciones de manejo más controladas, como la presencia de riego tecnificado y un menor grado de deterioro foliar. En contraste, se observa una ligera disminución de la efectividad en los bloques con mayor afectación del cultivo, lo que sugiere que el estado fitosanitario influye en el desempeño del sistema. Esta reducción se relaciona principalmente con factores propios del entorno agrícola real, tales como la superposición de hojas, variaciones en las condiciones de iluminación y la presencia de objetos externos al cultivo, como los carteles de identificación de los bloques, que generan detecciones erróneas. No obstante, aun considerando estas limitaciones, el sistema alcanza una efectividad

cercana al 90 % en condiciones reales de campo, lo cual representa un desempeño altamente satisfactorio para aplicaciones agrícolas de detección temprana.

4.7 Análisis Costo/Beneficio

La Tabla 19 presenta el desglose de los costos asociados al hardware necesario para la implementación del sistema propuesto. En ella se detallan la cantidad de cada componente, el elemento utilizado, su costo unitario y el costo total correspondiente. El hardware considerado incluye una placa embebida Raspberry Pi 4, que actúa como unidad de procesamiento principal, y una cámara, empleada para la captura de imágenes o datos visuales. El costo total de los componentes de hardware asciende a 165 dólares, lo cual refleja una solución económicamente accesible y adecuada para el desarrollo del proyecto.

Tabla 19

Costos de hardware

Cantidad	Elemento	Costo Unitario	Costo Total
1	Placa embebida Raspberry Pi4	\$ 120	\$ 120
1	Cámara	\$ 35	\$ 35
1	Batería de moto	\$ 20	\$ 20
1	Panel solar	\$ 10	\$ 10
1	Circuito electrónico	\$ 34.10	\$ 34.10
TOTAL			\$ 219.10

La Tabla 20 presenta el detalle de los costos asociados al software utilizado en el desarrollo e implementación del sistema propuesto. Se incluyen herramientas y plataformas de uso libre o gratuito, como Google Drive, Google Colab, RoboFlow, Python y Firebase, las cuales no representan un costo económico directo. Asimismo, se considera el uso de Google

Cloud, cuyo costo se calcula en función del consumo de servicios, generando un costo total de 23.46 dólares. En conjunto, estos valores reflejan que la mayor parte del software empleado corresponde a soluciones de acceso gratuito, reduciendo significativamente los costos del proyecto.

Tabla 20

Costos de software

Cantidad	Elemento	Costo Unitario	Costo Total
1	Google Drive	0	0
1	Google Colab	0	0
1	Roboflow	0	0
1	Python	0	0
1	Firebase	0	0
1	Google Cloud	2.9	23.46
TOTAL			\$ 23.46

La Tabla 21 muestra los costos correspondientes a la infraestructura necesaria para la implementación física del sistema. En ella se detallan los elementos requeridos, entre los que se incluyen una estructura metálica, cableado y un adaptador nano USB con tecnología Bluetooth, junto con sus respectivos costos unitarios y totales. El costo total de la infraestructura asciende a 37.5 dólares, lo que evidencia que los requerimientos físicos del sistema pueden ser cubiertos con una inversión moderada.

Tabla 21

Costos de infraestructura

Cantidad	Elemento	Costo Unitario	Costo Total
1	Estructura metálica	\$ 25	\$ 25
30m	Cable de timbre	\$ 0.20	\$ 6
1	Adaptador nano usb bluetooth	\$ 6.5	\$ 6.5
TOTAL			\$ 37.5

La estimación de costos presentada en la tabla 22 se fundamenta en valores reales de remuneración profesional en el contexto ecuatoriano y corresponde a las actividades necesarias para el desarrollo y validación del sistema de visión artificial propuesto. El estudio de campo y documentación se realizó en una sola jornada de trabajo, equivalente a un día, e incluyó la inspección del cultivo, identificación visual de síntomas asociados a *Botrytis cinerea*, recopilación de imágenes y registro de información técnica, para lo cual se consideró la asesoría de un ingeniero agrónomo con un costo estimado de 80 USD por día. (Computrabajo Ecuador (o DGNET LTD como empresa propietaria), 2025)

El diseño de hardware y la programación de los códigos de entrenamiento y clasificación fueron desarrollados por un ingeniero en telecomunicaciones, considerando aproximadamente 50 horas de trabajo especializado destinadas a la integración de componentes, configuración de la cámara, desarrollo del algoritmo de visión artificial, entrenamiento del modelo y validación de resultados. Para la valoración económica de estas actividades se tomó como referencia la remuneración promedio establecida para un ingeniero en telecomunicaciones en Ecuador, la cual, según datos oficiales del Ministerio del Trabajo, equivale a un salario mensual aproximado de 496,57 USD, correspondiente a un valor de 3,08 USD por hora. (Silvia Guadalupe Barba Proaño, 2025)

Finalmente, la implementación de la infraestructura requirió únicamente 5 horas de trabajo técnico, destinadas a la instalación física del sistema, conexiones eléctricas y pruebas

básicas de funcionamiento, aplicando el mismo valor horario. En conjunto, estos costos reflejan una estimación realista y coherente, demostrando que el sistema propuesto es económicamente viable y puede ser implementado como una solución de bajo costo para el monitoreo fitosanitario y la detección temprana de enfermedades en cultivos agrícolas.

Tabla 22

Costos de ingeniería

Actividades	Costo Unitario	Costo Total
Estudio de campo y 1 día (8 h)	\$ 80	\$ 80
documentación		
Diseño de hardware,	45 horas 15 / hora	\$ 675
programación de códigos de		
entrenamiento y clasificación		
Implementación de	\$ 3.08	\$ 15.4
infraestructura		
TOTAL		\$ 770.4

La Tabla 23 muestra el costo general del sistema, resultado de la integración de los costos de hardware, software, infraestructura e ingeniería previamente detallados. Esta tabla permite visualizar de manera consolidada la inversión total requerida para la implementación del sistema propuesto. El costo total asciende a 1050,46 dólares, lo que proporciona una referencia clara del presupuesto global del proyecto y facilita su evaluación económica.

Tabla 23

Costo general del sistema

Descripción	Costo Unitario
--------------------	-----------------------

Costos de hardware	\$ 219.1
Costos de software	\$ 23.46
Costos de Infraestructura	\$ 37.5
Costos de Ingeniería	\$ 770.4
TOTAL	\$ 1050.46

4.7.1 Beneficios

A continuación, se presentan los principales beneficios derivados de la implementación del sistema inteligente basado en visión artificial para la detección temprana de Botrytis cinerea en hojas de granadilla:

- La integración de tecnologías modernas como visión artificial, procesamiento digital de imágenes y aprendizaje automático permite mejorar la gestión del cultivo, facilitando el monitoreo continuo del estado fitosanitario de las plantas y optimizando la toma de decisiones agrícolas.
- El sistema propuesto permite identificar de manera temprana la presencia de hojas afectadas por Botrytis cinerea, lo que posibilita la aplicación oportuna de medidas correctivas, reduciendo la propagación de la enfermedad y minimizando pérdidas en la producción.
- La detección temprana contribuye a mejorar la calidad del producto final, lo que se traduce en un mayor valor de comercialización de la granadilla, beneficiando directamente a los productores mediante un incremento en los ingresos económicos.
- La automatización del proceso de clasificación de hojas reduce la dependencia de la evaluación visual humana, disminuyendo errores asociados al cansancio, la fatiga visual o la subjetividad, y garantizando resultados más consistentes y confiables.

- El sistema presenta una solución de bajo costo y fácil implementación, lo que lo hace accesible para pequeños y medianos productores, promoviendo el uso de tecnologías inteligentes en el sector agrícola.

4.7.2 Evaluación de la rentabilidad e impacto del sistema de detección temprana en el cultivo de granadilla

4.7.2.1 Análisis de Recuperación de la Inversión Y Rentabilidad del Sistema Propuesto

Para determinar la viabilidad económica del sistema de detección temprana de Botrytis cinerea en hojas de granadilla, se realizó un análisis de recuperación de la inversión considerando únicamente los costos reales necesarios para la implementación del sistema en campo. En este análisis no se incluyen los costos de ingeniería, ya que estos corresponden al desarrollo académico del proyecto y no representan un gasto directo para el productor al momento de adoptar la tecnología.

4.7.2.2 Inversión Total Real del Sistema

De acuerdo con la Tabla 17 a la Tabla 19 correspondiente a los costos generales del sistema, la inversión necesaria para la implementación del prototipo de detección temprana se compone de los siguientes rubros:

- **Costos de hardware:** 219,10 dólares
- **Costos de software:** 23,46 dólares
- **Costos de infraestructura:** 37,50 dólares
- **Costos de ingeniería:** 770.40 dólares

La suma de estos valores da como resultado una inversión total real inicial de 1050.46 dólares, la cual contempla tanto los elementos físicos del sistema como el desarrollo, diseño, integración y validación técnica del prototipo.

Es importante señalar que los costos de ingeniería se consideran únicamente en la primera implementación, debido a que corresponden al desarrollo inicial del sistema. En futuras implementaciones o réplicas del proyecto, este rubro no será necesario, reduciendo la inversión a 280,06 dólares, correspondientes únicamente a hardware, software e infraestructura.

4.7.2.3 Incremento en el Precio de Comercialización de la Granadilla

La implementación del sistema tiene un impacto directo en la mejora del estado fitosanitario del cultivo, lo cual se traduce en un aumento del precio de comercialización de la granadilla. Durante el periodo de evaluación se observó que:

- Antes de la implementación, el precio promedio de la caja de granadilla fue de 13,50 dólares.
- Después de la implementación, el precio promedio se incrementó a 15,50 dólares por caja.

Este incremento de 2,00 dólares por caja representa un aumento aproximado del 14,8 % en el valor de comercialización, atribuible a la mejora de la calidad del producto, la reducción de pérdidas por enfermedades y una mayor aceptación en el mercado.

4.7.2.4 Escenario de Producción Considerado

Para el análisis económico se consideró un escenario conservador, acorde con la realidad de pequeños productores, con una producción mensual estimada de 40 cajas de granadilla. Adicionalmente, se incorporaron costos operativos y de insumos agrícolas reales asociados a la producción y comercialización del cultivo, los cuales corresponden a la parcela completa y no únicamente a los tres bloques evaluados en el estudio. En cuanto a los costos

operativos mensuales, se contempla la mano de obra, correspondiente a dos personas con un costo de 15 dólares por persona, equivalente a 30 dólares mensuales; el uso de cajas vacías, con un costo unitario de 1,20 dólares por caja, lo que representa un gasto mensual de 48 dólares; y el transporte desde la parcela hasta la ciudad de Ibarra, con un costo aproximado de 1 dólar por caja, equivalente a 40 dólares mensuales.

A estos valores se suman los costos de insumos agrícolas, donde la fumigación se realiza cada 15 días, con un gasto mensual aproximado de 120 dólares, mientras que la abonada se efectúa cada dos meses, con un costo de 300 dólares, lo que equivale a un gasto mensual promedio de 150 dólares. De esta manera, el costo operativo mensual total, incluyendo labores, logística e insumos, asciende a 388 dólares. Cabe destacar que estos gastos se mantienen tanto antes como después de la implementación del sistema propuesto, por lo que no influyen directamente en el incremento de ingresos, pero permiten evaluar de forma realista el margen económico del proyecto y el impacto potencial del sistema de detección temprana en la optimización de las prácticas agrícolas.

4.7.2.5 Ingreso Adicional Mensual Generado por el Sistema

Con base en el incremento promedio de 2,00 dólares por caja y una producción mensual estimada de 40 cajas de granadilla, el ingreso adicional generado por la mejora en la calidad del producto se calcula multiplicando ambos valores, lo que resulta en un ingreso adicional mensual de 80 dólares. Este incremento económico se atribuye principalmente a la detección temprana de hojas afectadas por *Botrytis cinerea*, lo que permite una gestión más oportuna del cultivo, reduciendo pérdidas por deterioro del fruto y mejorando su presentación comercial. Es importante destacar que este beneficio se obtiene sin modificar los costos operativos habituales de producción y comercialización, ya que gastos como mano de obra, cajas, transporte e insumos agrícolas forman parte del proceso productivo regular de la parcela y se mantienen constantes antes y después de la implementación del sistema.

Al comparar este ingreso adicional con el costo operativo mensual total, que incluye tanto los costos logísticos como los insumos agrícolas (fumigación y abonada), se observa que el incremento generado representa aproximadamente un 20,6 % del costo operativo mensual, lo que evidencia un impacto económico positivo y sostenible. Si se considera únicamente el costo operativo directo asociado a comercialización y mano de obra, el incremento equivale a un porcentaje aún mayor, lo que demuestra que el sistema propuesto contribuye de manera significativa a mejorar la rentabilidad del productor sin generar un aumento en los gastos de operación, consolidándose como una herramienta económicamente viable para el monitoreo fitosanitario del cultivo.

4.7.2.6 Tiempo de Recuperación de la Inversión

El tiempo de recuperación de la inversión se determinó a partir de la relación entre la inversión total realizada y el ingreso adicional mensual generado por la mejora en la calidad del producto. Para la primera implementación del sistema, que incluye los costos asociados al desarrollo de ingeniería, la inversión total asciende a 1 050,46 dólares. Considerando un ingreso adicional mensual estimado de 80 dólares, el tiempo de recuperación de la inversión es de aproximadamente 13,1 meses. Este valor refleja el carácter pionero del proyecto, en el cual se concentran los costos de diseño, programación y validación del sistema.

En el caso de implementaciones futuras, donde no se consideran los costos de ingeniería debido a la reutilización del sistema desarrollado, la inversión se reduce a 280,06 dólares, manteniéndose el mismo ingreso adicional mensual. Bajo estas condiciones, el tiempo de recuperación de la inversión es de aproximadamente 3,5 meses, lo que evidencia una recuperación significativamente más rápida. Estos resultados confirman que, si bien la inversión inicial es mayor en la primera implementación, el sistema presenta una alta viabilidad económica y un retorno atractivo, especialmente cuando se considera su replicación en otras parcelas o ciclos productivos.

4.8 Comparación Cuantitativa Antes y Después de la Implementación del Sistema

Las diferencias en el número de hojas malas detectadas entre el conteo manual y el sistema se deben, en parte, a detecciones erróneas de elementos no vegetales, lo cual se considera una limitación puntual del modelo.

A continuación en la tabla 24 se muestra una comparación del sistema antes y después de la implementación del sistema de monitoreo.

Tabla 24

Antes y Después de la Implementación del Sistema

Indicador	Antes del sistema	Después del sistema
Producción mensual	40 cajas	40 cajas
Precio de venta por caja	Precio base	Precio base + 2,00 USD
Incremento por caja	0,00 USD	2,00 USD
Ingreso mensual adicional	0,00 USD	80,00 USD
Fumigaciones mensuales	2	2
Costo operativo mensual total	388,00 USD	388,00 USD
Inversión inicial	0,00 USD	1 050,46 USD
Ingreso adicional / costo operativo	0 %	20,6 %
Tiempo de recuperación	No aplica	13,1 meses

4.8.1 Análisis de rentabilidad y retorno de la inversión aplicado a la totalidad del cultivo

El análisis económico desarrollado en esta investigación se realizó sobre tres bloques representativos como se muestra en la tabla 25, los cuales corresponden aproximadamente al 10 % del área total sembrada de la finca. En este escenario de estudio parcial, se determinó un ingreso adicional mensual de 80 dólares, derivado de un incremento promedio de 2,00 dólares por caja y una producción mensual de 40 cajas, atribuible a la mejora en la calidad del producto gracias a la detección temprana de *Botrytis cinerea* mediante el sistema de visión artificial.

Considerando que las condiciones productivas, fitosanitarias y de manejo del cultivo se mantienen homogéneas en el resto del sembradío, es posible realizar una proyección económica lineal para el 100 % del área cultivada.

Bajo esta premisa, al extrapolar los resultados obtenidos al total del sembradío, el ingreso adicional mensual potencial ascendería a 800 dólares, equivalente a diez veces el valor observado en el área analizada. En términos anuales, este incremento representaría un beneficio económico aproximado de 9 600 dólares, sin considerar un aumento en los costos operativos habituales, ya que actividades como mano de obra, transporte, fumigación y abonada forman parte del proceso productivo regular de la finca. Esta proyección evidencia que la implementación del sistema a escala total permitiría mejorar de manera significativa la rentabilidad del cultivo, reduciendo pérdidas asociadas a la detección tardía de la enfermedad y optimizando la toma de decisiones agronómicas.

Adicionalmente, al considerar que los costos de ingeniería corresponden principalmente a la etapa inicial de desarrollo, la replicación del sistema en el resto del sembradío implicaría una inversión marginal reducida, limitada principalmente a la adquisición e instalación de equipos adicionales. En este contexto, el tiempo de recuperación de la inversión para la finca completa sería considerablemente menor, consolidando al sistema propuesto como una solución escalable, económicamente sostenible y de alto impacto para productores de granadilla. En conjunto, estos resultados demuestran que el uso de tecnologías de visión artificial no solo mejora el monitoreo fitosanitario a pequeña escala, sino que presenta un alto potencial de rentabilidad cuando se implementa de manera integral en toda la unidad productiva.

Tabla 25

Proyección económica del sistema: 10 % del sembradío vs. 100 % de la finca

Indicador	Área estudiada (10 %)	Proyección total (100 %)
-----------	-----------------------	--------------------------

Porcentaje del sembradío	10 %	100 %
Producción mensual	40 cajas	400 cajas
Incremento por caja	2,00 USD	2,00 USD
Ingreso adicional mensual	80,00 USD	800,00 USD
Ingreso adicional anual	960,00 USD	9 600,00 USD
Costo operativo mensual	388,00 USD	388,00 USD*
Inversión inicial del sistema	1050,46 USD	2251,96 USD
Tiempo de recuperación	13,1 meses	≈ 2,8 meses

** Los costos operativos corresponden a la parcela completa y se mantienen constantes, independientemente del porcentaje del área analizada.*

4.8.1.1 Costos de Hardware Escalados a un 100% (Total de la Finca)

Para la proyección del sistema al 100 % del área cultivada como se muestra en la tabla 26 se considera que cada bloque de monitoreo opera como un nodo autónomo e independiente, compuesto por una placa embebida Raspberry Pi 4, una cámara, una batería, un panel solar y su respectivo circuito electrónico. En este sentido, al escalar el sistema desde el 10 % del sembradío (1 nodo) hasta el 100 % del área total (10 nodos), todos los componentes de hardware se incrementan proporcionalmente en un factor de diez.

Tabla 26

Costos de Hardware

Cantidad	Elemento	Costo unitario (USD)	Costo total (USD)
10	Raspberry Pi 4	120 USD	1200 USD
10	Cámara	35 USD	350 USD
10	Batería de moto	20 USD	200 USD
10	Panel Solar	10 USD	100 USD

10	Circuito electrónico	34.10 USD	341 USD
TOTAL			2 191,00 USD
HARDWARE			

Para el análisis de la inversión real requerida por el productor en la implementación del sistema de detección temprana de *Botrytis cinerea* a escala del 100 % del sembradío, se consideraron únicamente los costos directamente asociados a la adquisición e instalación del sistema en campo. En este sentido, los costos de ingeniería no fueron incluidos, ya que corresponden al desarrollo académico inicial del prototipo y no representan un gasto recurrente para el agricultor. Bajo este criterio, la inversión real está compuesta por los costos de hardware, que ascienden a 2 191,00 USD, los costos de software por un valor de 23,46 USD y los costos de infraestructura equivalentes a 37,50 USD. En conjunto, la inversión total real necesaria para la implementación del sistema en toda la finca es de 2 251,96 USD, lo que constituye la base para el análisis de recuperación de la inversión y rentabilidad económica del sistema propuesto.

4.8.1.2 Tiempo de Recuperación de la Inversión

Bajo el escenario de implementación del sistema en el 100 % del sembradío, y considerando una inversión real de 2 251,96 USD, el tiempo estimado de recuperación de la inversión es de aproximadamente 2,8 meses, lo que evidencia una recuperación rápida y una alta viabilidad económica para el productor.

$$\text{Tiempo de recuperación} = \frac{\text{Inversión}}{\text{Ingreso mensual}}$$

Para la implementación del tiempo de recuperación se obtiene los siguientes datos:

- **Inversión del sistema:** 2251.96 USD

- **Ingreso mensual:** 800 USD

$$\text{Tiempo de recuperación} = \frac{2251,96}{800} \approx 2,81 \text{ meses}$$

4.8.2 Cálculo Del ROI Anual de Toda la Finca

El retorno de la inversión obtenido indica que el sistema permite recuperar la inversión inicial más de cuatro veces en un año, aun considerando el escalamiento completo del hardware requerido para cubrir la totalidad del sembradío. Este resultado confirma que el sistema de visión artificial propuesto es económicamente rentable, escalable y sostenible para su aplicación integral en fincas productoras de granadilla.

$$ROI(\%) = \frac{\text{Beneficio anual}}{\text{Inversión inicial}} \times 100$$

Para la implementación del sistema en el 100 % del sembradío, se tiene:

- **Ingreso adicional anual:** 9 600 USD
- **Inversión inicial:** 1 050,46 USD

$$ROI(\%) = \frac{9600}{2251.96} \approx 4.26\%$$

4.9 Análisis Comparativo del Sistema Fijo de Visión Artificial y el Uso de Drones para la Detección Temprana de *Botrytis cinerea*

Con el fin de validar la pertinencia de la solución tecnológica propuesta, se realizó una evaluación comparativa entre el sistema de visión artificial basado en nodos fijos y el uso de drones como alternativa para la detección temprana de *Botrytis cinerea* en cultivos de granadilla. Este análisis consideró aspectos técnicos, operativos y económicos, enfocados en el contexto real de pequeños y medianos productores agrícolas.

Si bien el uso de drones permite obtener una visión general del cultivo y cubrir grandes áreas en periodos cortos de tiempo, su efectividad para la detección temprana de enfermedades foliares es limitada, debido a la necesidad de capturas a baja altura y alta resolución, así como a su dependencia de condiciones climáticas favorables y de personal capacitado para su

operación. Estas restricciones reducen la frecuencia del monitoreo y pueden retrasar la identificación oportuna de los primeros síntomas de la enfermedad.

En contraste, el sistema de visión artificial propuesto, al operar mediante nodos fijos instalados a nivel del follaje, permite un monitoreo continuo, automático y localizado, facilitando la detección temprana de síntomas sutiles en hojas individuales. Además, su funcionamiento autónomo mediante energía solar y su bajo requerimiento de intervención humana lo convierten en una solución altamente viable para entornos rurales.

Desde el punto de vista económico, el sistema fijo presenta una inversión inicial comparable o incluso inferior a la de un dron agrícola, pero con la ventaja de no generar costos operativos recurrentes ni depender de servicios externos. Adicionalmente, su escalabilidad modular permite una implementación progresiva, ajustada a la capacidad económica del productor.

En función de los resultados obtenidos y del análisis comparativo realizado se puede visualizar en la tabla 27 , y se determina que el sistema de visión artificial basado en nodos fijos representa una alternativa tecnológica más adecuada, eficiente y sostenible para la detección temprana de *Botrytis cinerea* en cultivos de granadilla.

Tabla 27

Comparación de soluciones tecnológicas para monitoreo de cultivo: sistema fijo vs. drones

Criterio	Sistema de visión artificial (propuesto)	Dron agrícola (nivel básico / entrada)	Dron agrícola (multisensor profesional)
Inversión inicial estimada	2 251,96 USD (nodos + software + infraestructura)	≈ 1 500 – 5 000 USD (modelo de mapeo básico)	≈ 8 000 – 15 000+ USD (multispectral/NDVI con software)
Monitoreo continuo	SI	NO	NO

Frecuencia de monitoreo potencial	Diario / automático	Según disponibilidad de vuelo	Según disponibilidad de vuelo
Capacidad de detección foliar	Alta (niveles de hoja individual)	Media (imagen aérea general)	Alta (sensores NDVI/multiespectral)
Requerimiento de operador especializado	NO	Sí (piloto/certificado)	Sí (piloto + análisis)
Dependencia climática para operación	Baja	Alta (viento/lluvia)	Alta (viento/lluvia)
Necesidad de infraestructura adicional	Energía solar y estructura ligera	Sí (zona despejada para despegue)	Sí (misma + software y planificación)

La comparación entre el sistema de visión artificial propuesto y el uso de drones agrícolas muestra que ambos enfoques tecnológicamente pueden aportar valor al monitoreo de cultivos, pero con diferencias sustanciales en costos, operación y aplicabilidad para el agricultor promedio. El sistema fijo de visión artificial requiere una inversión estimada de 2 251,96 USD para cubrir toda el área objetivo con nodos permanentes y operativos (incluyendo cámara, Raspberry Pi, baterías y paneles solares), sin necesidad de operador especializado y con monitoreo continuo directo del cultivo.

En cuanto a los drones, los precios estimados para modelos adecuados al monitoreo de cultivos varían ampliamente según sus capacidades tecnológicas. Los drones básicos con cámaras RGB apropiadas para vigilancia aérea pueden costar entre aproximadamente 1 500 y 5 000 USD, lo que representa una alternativa de entrada al uso de imágenes aéreas. Sin embargo, estos modelos no proporcionan necesariamente la resolución y consistencia requerida para la detección temprana a nivel foliar, que es un requisito central para enfermedades como *Botrytis cinerea* y puede requerir sensibilidad de sensor y vuelo bajo.

Por su parte, los drones con sensores multiespectrales o NDVI más sofisticados que ofrecen análisis de salud del cultivo y patrones nutricionales tienen costos que pueden superar los 8 000 – 15 000 USD o más, dependiendo de la marca, sensores y software adicional. Estos sistemas suelen requerir no solo la adquisición del equipo, sino también capacitación especializada del operador, mantenimiento periódico y licencias de vuelo para su uso seguro y eficiente, lo que incrementa los costos operativos totales del agricultor.

CONCLUSIONES

- El sistema de visión artificial desarrollado permitió identificar hojas afectadas por *Botrytis cinerea* de manera automática, logrando detectar en el área de estudio un total de 14 hojas clasificadas como malas frente a 23 hojas identificadas mediante conteo manual tomando esto como muestra de un bloque, evidenciando diferencias atribuibles tanto a la detección temprana como a limitaciones puntuales del modelo asociadas a la confusión de elementos no vegetales, como los carteles de identificación de los bloques.
- El conteo manual realizado por los propietarios de la finca, la señora Marcia Robles y el señor Ermel Robles, permitió registrar un total de 726 hojas, de las cuales 555 hojas (76,45 %) fueron consideradas aptas para el análisis, lo que garantizó una base de datos representativa y evidenció variaciones claras entre bloques, siendo el Bloque 1 el de mejor condición fitosanitaria (82,05 % de hojas consideradas) y el Bloque 3 el más afectado (71,28 % de hojas consideradas).
- El análisis comparativo por bloques mostró que el Bloque 3, el cual no contó con riego tecnificado ni fumigación, presentó un mayor número de hojas afectadas (10 hojas según conteo manual) en comparación con el Bloque 1 (5 hojas) y el Bloque 2 (8 hojas), los cuales sí dispusieron de riego tecnificado y fumigación, evidenciando la influencia directa de las prácticas agronómicas en el estado fitosanitario del cultivo.
- Desde el punto de vista económico, la implementación del sistema permitió un incremento promedio de 2,00 USD por caja, lo que, considerando una producción mensual de 40 cajas, generó un ingreso adicional mensual de 80 USD en el área evaluada, sin incrementar los costos operativos habituales, los cuales se mantuvieron en 388 USD mensuales, incluyendo mano de obra, transporte e insumos agrícolas.

- La evaluación de rentabilidad determinó que la inversión inicial de 1 050,46 USD presenta un tiempo de recuperación aproximado de 13,1 meses en la primera implementación; sin embargo, al proyectar el sistema al 100 % del sembradío, el ingreso adicional anual estimado asciende a 9 600 USD, alcanzando un retorno de la inversión (ROI) del 4.26 %, lo que demuestra un alto potencial económico al escalar la solución.
- En conjunto, los resultados técnicos y económicos confirman que el sistema propuesto es viable, escalable y sostenible, ya que permite mejorar la detección temprana de Botrytis cinerea, optimizar la toma de decisiones agronómicas y aumentar la rentabilidad del productor, constituyéndose como una alternativa tecnológica de bajo costo con alto impacto para pequeños y medianos productores de granadilla.

RECOMENDACIONES

- Se recomienda ampliar el conjunto de datos utilizado para el entrenamiento del modelo de visión artificial, incorporando imágenes de elementos no vegetales presentes en el entorno del cultivo, como carteles de identificación, estructuras de soporte y fondo del suelo, con el fin de reducir las detecciones erróneas observadas en los Bloques 2 y 3 y mejorar la precisión general del sistema.
- Es aconsejable implementar una clasificación adicional o una etapa de preprocesamiento que permita diferenciar entre hojas y objetos externos, lo cual contribuiría a disminuir falsas detecciones y a fortalecer la robustez del sistema en condiciones reales de campo.
- Se recomienda extender la implementación del sistema al 100 % del sembradío, considerando que el área evaluada corresponde únicamente al 10 % de la finca y que la proyección económica evidencia un alto retorno de la inversión y un incremento significativo en la rentabilidad del cultivo cuando el sistema se aplica de manera integral.
- Se sugiere evaluar la aplicación del sistema en otros cultivos susceptibles a enfermedades fúngicas, con el fin de validar su adaptabilidad y ampliar su campo de aplicación.

Se sugiere complementar el sistema de visión artificial con registros periódicos de prácticas agronómicas, como riego y fumigación, a fin de correlacionar de manera más precisa el estado fitosanitario del cultivo con las condiciones de manejo y optimizar la toma de decisiones preventivas.

- Es recomendable realizar evaluaciones temporales continuas del sistema, aplicando el monitoreo en diferentes etapas fenológicas del cultivo y bajo distintas condiciones

climáticas, para validar su desempeño a largo plazo y asegurar su adaptabilidad frente a variaciones ambientales.

- Finalmente, se recomienda considerar la integración del sistema con plataformas de monitoreo remoto o aplicaciones móviles, de modo que los productores puedan acceder de forma rápida y sencilla a alertas, reportes y resultados del análisis, facilitando la adopción de la tecnología y promoviendo una agricultura más eficiente y sostenible.

4 BIBLIOGRAFÍA

- ADAMA. (2022). *Manejo integrado del Moho gris (Botrytis cinerea) en cultivos de agroexportación*. <https://www.adama.com/peru/es/blog/manejo-integrado-del-moho-gris-botrytis-cinerea-en-cultivos-de-agroexportacion>
- Agrios, A. (1997). Control de enfermedades vegetales. *Patologías de La Planta*, 200–216.
- Antontio, T. (2006). Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot. *Test*, 1–125.
- Ausum Cloud. (2024). *Google Cloud gratis: cómo usar la nube de Google sin coste*. <https://ausum.cloud/google-cloud-gratis/>
- Baca, G. (2013). *Evaluación de proyectos* (7ma ed.). McGraw-Hill. https://www.uachatec.com.mx/wp-content/uploads/2019/05/LIBRO-Evaluación-de-proyectos-7ma-Edición-Gabriel-Baca-Urbina-FREELIBROS.ORG_.pdf
- Carabalí Méndez. (2020). *Diseño de un Sistema de medición de la calidad del agua del río Chota de la comunidad del mismo nombre, ubicada en la Parroquia de Ambuqui, en Catón Ibarra en la provincia de Imbabura*. [UNIVERSIDAD TÉCNICA DEL NORTE]. <http://repositorio.utn.edu.ec/handle/123456789/10100>
- Carlos, J., & Pérez, M. (2018). *Aprendizaje supervisado para la detección de amenazas Web*.
- Cerdas, M., & Castro, J. (2003). Manual Práctico Para La Producción , Cosecha Y Manejo Granadilla. In *Ministerio de Agricultura y Ganadería*. http://www.mag.go.cr/biblioteca_virtual_ciencia/tec-granadilla.pdf
- Cicero, I. (2018). *Utilización de redes neuronales convolucionales para la detección de tipos de imágenes*. Instituto Tecnológico de Buenos Aires.
- Computrabajo Ecuador (o DGNET LTD como empresa propietaria). (2025). *Salarios de Ingeniero en sistemas en Ecuador 2026*.

- Deusto Formación. (2022, August 22). *Datasets y Dataframes en Big Data: ¿qué son?*
<https://www.deustoformacion.com/blog/programacion-tic/que-son-datasets-dataframes-big-data>
- Enric Cervera, R. M. J. M. (2022). *Más allá de Jupyter: usando Google Colab para la Programación de Robots*.
https://ruc.udc.es/dspace/bitstream/handle/2183/31486/2022_Cervera_Enric_Mas_alla_de_Jupyter_usando_Google_Colab_para_la_programacion_de_robots.pdf?sequence=3&isAllowed=y
- Facchin, J. (2022). *Crear una página Web gratis*. <https://josefacchin.com/crear-pagina-web-gratis/>
- García, R. (2017). *Bortrytis cinerea en el cultivo de rosa híbrida en la zona florícola sur del estado* [Universidad Autónoma del Estado de México].
<http://hdl.handle.net/20.500.11799/68560>
- Gonzales, F. (2013). *Granadilla - Cultivo*. Calaméo.
<https://www.calameo.com/read/00125340480205525c0fe>
- González García, I. T. (2015). *Diseño e implementación de sistema interactivo de información de Docentes, con Raspberry PI*. <http://dspace.ups.edu.ec/handle/123456789/10408>
- Guerrero, A. (2014). *Passiflora ligularis* “granadilla.”
<http://pinzonesygorriones.blogspot.com/2014/>
- IKI (International Climate Initiative). (2023). *Mapa de distribución: Propiedad Ermel Antonio Robles Pallacho*.
- Ilustre Colegio Oficial de Geólogos. (2025, February 17). *V Curso de introducción a la programación en Python para geólogos*. <https://www.icog.es/cursos/index.php/v-curso-de-introduccion-a-la-programacion-en-python-para-geologos/>

- Institute for Quantitative Social Science. (2021). *Dataset + File Management*.
<https://guides.dataverse.org/en/latest/user/dataset-management.html>
- Jesús Martínez. (2024, February 23). *Botrytis: el hongo de la podredumbre gris de los cultivos*.
<https://www.zerimarlaboratoire.com/es/post/botrytis-el-hongo-de-la-podredumbre-gris-de-los-cultivos>
- Kouhia, E.-P. (2019). *Development of an Arduino-Based Embedded System*. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3422103>
- López, R. (2012). *Deep Learning*.
- Matute, P. (2019). *Control biológico del moho gris (Botrytis cinerea) en cultivos de fresa (Fragaria vesca L.) mediante hongos filamentosos antagonistas* [Universidad Politécnica Salesiana]. <https://dspace.ups.edu.ec/bitstream/123456789/18147/1/UPS-CT008620.pdf>
- Mekawi, E. M., Khafagi, E. Y., & Abdel-Rahman, F. A. (2019). Effect of pre-harvest application with some organic acids and plant oils on antioxidant properties and resistance to *Botrytis cinerea* in pepper fruits. *Scientia Horticulturae*, 257. <https://doi.org/10.1016/j.scienta.2019.108736>
- Mercado Libre Ecuador. (n.d.). *Cámara de seguridad Wi-Fi Smart Camera ABQ-A8 con resolución de 2MP, visión nocturna incluida, blanco*. Retrieved July 14, 2025, from <https://www.mercadolibre.com.ec/camara-de-seguridad-wi-fi-smart-camera-abq-a8-con-resolucion-de-2mp-vision-nocturna-incluida-blanco/p/MEC19043111>
- Mohammed, A. (2021, June 9). *Setting up Google Colab for Deep Learning*. <https://dev.to/sheikhazhanmohammed/setting-up-google-colab-for-deep-learning-1ion>
- Núñez, F. (2016). *Diseño de un sistema de reconocimiento automático de matrículas de vehículos mediante una red neuronal convolucional* [Universitat Oberta de Catalunya]. <https://openaccess.uoc.edu/bitstream/10609/52222/7/fnunezsTFM0616memoria.pdf>
- Okoone Ltd. (2025). *Firestore*. <https://www.okoone.com/technologies/cloud/firebase/>

- OSTEPEC. (2009). *Abastecimiento de la granadilla*.
<https://abastecimientodegranadilla.blogspot.com/>
- Pineda, C. (2019). *Diseño De Un Sistema De Reconocimiento Automático De Vehículos Mediante El Uso De Redes Neuronales Profundas (Dnn)* [Universidad Técnica del Norte].
<https://repositorio.utn.edu.ec/handle/123456789/9136>
- Ponce, D. (2019). *El cultivo de granadilla (passiflora ligularis) como una alternativa de mejoramiento de los ingresos de los pequeños agricultores de la parroquia de Cahuasqui, cantón Urcuquí* [Universidad Técnica del Norte].
https://repositorio.utn.edu.ec/bitstream/123456789/9970/2/PG_773_TRABAJO_DE_GRADO.pdf
- Quiroga, F. ; R. F. ; L. L. ; F.-B. A. (2023). *Revisiting Data Augmentation for Rotational Invariance in Convolutional Neural Networks*. <https://arxiv.org/abs/2310.08429>
- Rondón, G. (2003). *Manejo del cultivo de mora (Rubus glaucus Benth)*.
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 60. <https://doi.org/10.1186/s40537-019-0197-0>
- Silvia Guadalupe Barba Proaño. (2025). *SISTEMA DE CONTROL DE ACCESO AUTOMATIZADO CON INTELIGENCIA ARTIFICIAL PARA EL MONITOREO DE ESTUDIANTES Y DOCENTES EN LOS TALLERES TECNOLÓGICOS DE LA FISEI*. Universidad Técnica de Ambato.
- Tamayo, P. (2001). *Principales enfermedades del tomate de árbol, la mora y el lulo en Colombia*. Corporación Colombiana de Investigación Agropecuaria.
<https://repository.agrosavia.co/handle/20.500.12324/1175>
- Vina, A. (2024, January 10). *Ultralytics YOLOv8 Turns One: A Year of Breakthroughs and Innovations*. <https://www.ultralytics.com/blog/ultralytics-yolov8-turns-one-a-year-of-breakthroughs-and-innovations>

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9(4), 611–629. <https://doi.org/10.1007/s13244-018-0639-9>

Yostin Jahir Burgos Muñiz. (2024). *DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA DETECTAR PLAGAS EN LOS CULTIVOS DE ARROZ UTILIZANDO UNA RASPBERRY PI Y OPENCV* [UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL]. <http://dspace.ups.edu.ec/handle/123456789/27717>

5 ANEXOS

ANEXO A: Formato de entrevista

Entrevista

Tema de tesis: Prototipo electrónico basado en visión artificial para la detección temprana de la enfermedad Botrytis cinerea en hojas de granadilla(*Passiflora ligularis*).

Objetivo de la tesis: Diseñar un prototipo electrónico basado en visión artificial para la detección temprana de la enfermedad Botrytis cinerea en hojas de granadilla, para mejorar la gestión agrícola promoviendo prácticas sostenibles.

Objetivo de la encuesta: El objetivo de la encuesta es recopilar información fundamental para el desarrollo del prototipo electrónico basado en visión artificial para la detección temprana de la enfermedad Botrytis cinerea en hojas de granadilla. Su diseño está orientado a obtener datos relevantes sobre la incidencia de la enfermedad, las condiciones ambientales que favorecen su aparición y las necesidades de los productores para una detección más eficiente y oportuna.

Instrucciones:

1. Lea cuidadosamente cada pregunta.
2. Marque solo una opción por cada pregunta. Para hacerlo, simplemente encierre en un círculo (●) la respuesta que considere correcta o la que mejor se ajuste a su opinión o experiencia.
3. Si alguna pregunta no aplica a su situación, puede dejarla en blanco.
4. Responda con sinceridad. Sus respuestas son muy importantes para mejorar el desarrollo del prototipo de detección temprana de Botrytis cinerea en hojas de granadilla.

Nombre: _____

Cargo ejercido: _____

Fecha: _____

Preguntas para Agricultores

1. ¿Cuánto tiempo lleva cultivando granadilla?
 - a) Menos de 1 año

- b) Entre 1 y 5 años
 - c) Más de 5 años
- 2. ¿En qué temporada del año ha notado mayor incidencia de *Botrytis cinerea* en sus cultivos de granadilla?
 - a) Temporada de lluvias
 - b) Temporada seca
 - c) No he notado variaciones
- 3. ¿Cuáles han sido las principales dificultades que ha enfrentado al tratar de controlar *Botrytis cinerea* en su cultivo?
.....
.....
.....
- 4. ¿Cómo identifica actualmente la presencia de *Botrytis cinerea* en sus cultivos?
 - a) Inspección visual propia
 - b) Diagnóstico de un especialista
 - c) Uso de herramientas tecnológicas
 - d) No realizo detección
- 5. ¿Qué métodos utiliza para prevenir o tratar *Botrytis cinerea* en su cultivo?
 - a) Aplicación de fungicidas
 - b) Mejora en la ventilación del cultivo
 - c) Eliminación de hojas afectadas
 - d) Otros (especifique)
- 6. ¿Cuánto le afecta económicamente la presencia de *Botrytis cinerea* en sus cultivos?
 - a) Mucho
 - b) Algo
 - c) Poco
 - d) Nada
- 7. ¿Qué tan efectiva considera la inspección visual para detectar la enfermedad?
 - a) Muy efectiva
 - b) Moderadamente efectiva

- c) Poco efectiva
 - d) Nada efectiva
8. ¿Qué espera de un sistema basado en visión artificial para la detección de *Botrytis cinerea*?
- a) Precisión en la detección
 - b) Facilidad de uso
 - c) Costos accesibles
 - d) Alertas tempranas
9. ¿Cómo preferiría recibir alertas sobre la detección temprana de la enfermedad?
- a) Aplicación móvil
 - b) Mensajes SMS
 - c) Dispositivo con alerta visual o sonora
 - d) Otro (especifique)
10. ¿Estaría dispuesto a utilizar un prototipo tecnológico para la detección temprana de *Botrytis cinerea* en sus cultivos?
- a) Sí
 - b) No

Entrevista

Tema de tesis: Prototipo electrónico basado en visión artificial para la detección temprana de la enfermedad Botrytis cinerea en hojas de granadilla (*Passiflora ligularis*).

Objetivo de la tesis: Diseñar un prototipo electrónico basado en visión artificial para la detección temprana de la enfermedad Botrytis cinerea en hojas de granadilla, para mejorar la gestión agrícola promoviendo prácticas sostenibles.

Objetivo de la encuesta: El objetivo de la encuesta es recopilar información fundamental para el desarrollo del prototipo electrónico basado en visión artificial para la detección temprana de la enfermedad Botrytis cinerea en hojas de granadilla. Su diseño está orientado a obtener datos relevantes sobre la incidencia de la enfermedad, las condiciones ambientales que favorecen su aparición y las necesidades de los productores para una detección más eficiente y oportuna.

Instrucciones:

1. Lea cuidadosamente cada pregunta.
2. Marque solo una opción por cada pregunta. Para hacerlo, simplemente encierre en un círculo (●) la respuesta que considere correcta o la que mejor se ajuste a su opinión o experiencia.
3. Si alguna pregunta no aplica a su situación, puede dejarla en blanco.
4. Responda con sinceridad. Sus respuestas son muy importantes para mejorar el desarrollo del prototipo de detección temprana de Botrytis cinerea en hojas de granadilla.

Nombre: _____

Cargo ejercido: _____

Fecha: _____

Preguntas para Ingenieros Agrónomos

1. ¿Cuánto tiempo lleva trabajando en el sector agrícola?
 - a) Menos de 1 año
 - b) Entre 1 y 5 años

- c) Más de 5 años
2. ¿En qué etapa fenológica de la planta de granadilla suele aparecer con mayor frecuencia *Botrytis cinerea*?
 - a) Germinación
 - b) Crecimiento vegetativo
 - c) Floración
 - d) Cosecha
 3. ¿Cuáles considera que son los factores ambientales más determinantes para la proliferación de *Botrytis cinerea* en granadilla?

.....

.....

.....
 4. ¿Qué métodos recomienda actualmente para la detección temprana de *Botrytis cinerea*?

.....

.....

.....
 5. ¿Considera que la inspección visual tradicional es suficiente para la detección oportuna de *Botrytis cinerea*?
 - a) Sí, es suficiente
 - b) Sí, pero podría mejorarse
 - c) No, es poco confiable
 - d) No, es ineficiente
 6. ¿Ha trabajado con herramientas tecnológicas para la detección de enfermedades en cultivos?
 - a) Sí
 - b) No
 7. ¿Cree que un sistema basado en visión artificial podría mejorar la detección temprana de la enfermedad en comparación con los métodos actuales? ¿Por qué?

-
-
-
8. ¿Qué características considera esenciales en un prototipo de detección de enfermedades en cultivos para que sea útil en el campo?

.....

.....

.....

9. ¿Qué tan dispuesto estaría a recomendar el uso de un sistema basado en visión artificial para la detección de *Botrytis cinerea*?

- a) Muy dispuesto
- b) Algo dispuesto
- c) Poco dispuesto
- d) Nada dispuesto

10. ¿Qué otras enfermedades o problemas agrícolas cree que podrían beneficiarse del uso de visión artificial para su detección temprana?

.....

.....

.....

.....

Elaborado por:

Sr. Ermel Robles

C.I. 1003844881

Estudiante

Revisado por:

MsC. Fabián Cuzme

C.I. 1001234567

Director