



**UNIVERSIDAD NACIONAL DE CHIMBORAZO**  
**FACULTAD DE INGENIERÍA**  
**CARRERA DE INGENIERÍA EN TELECOMUNICACIONES**

Desarrollo de un sistema de clasificación basado en visión artificial  
para detección temprana de plagas de especies predominantes de  
mosquitos de la fruta en Agrocalidad.

**Trabajo de Titulación para optar al título de Ingeniero en  
Telecomunicaciones**

**Autores:**

Muyulema Taco, Cristian Alexander  
Mariño Barriga, Gonzalo Darío

**Tutor:**

PhD. Ing. Leonardo Fabian Rentería Bustamante

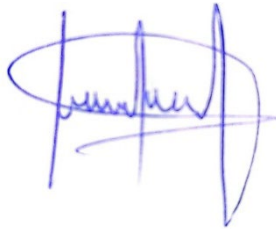
**Riobamba, Ecuador. 2026**

## DECLARATORIA DE AUTORÍA

Nosotros, Muyulema Taco Cristian Alexander, con cédula de ciudadanía 0302725171 y Mariño Barriga Gonzalo Dario, con cédula de ciudadanía 0605066356 autores del trabajo de investigación titulado: **Desarrollo de un Sistema de clasificación basado en Visión Artificial para detección temprana de plagas de Especies predominantes de Mosquitos de la Fruta en Agrocalidad**, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 05 de enero de 2026.



---

Muyulema Taco Cristian Alexander  
C.I: 0302725171



---

Nombres y Apellidos del estudiante  
C.I: 0605066356

## **DICTAMEN FAVORABLE DEL PROFESOR TUTOR**

Quien suscribe, Leonardo Fabian Rentería Bustamante catedrático adscrito a la Facultad de Ingeniería, por medio del presente documento certifico haber asesorado y revisado el desarrollo del trabajo de investigación Desarrollo de un Sistema de clasificación basado en Visión Artificial para detección temprana de plagas de Especies predominantes de Mosquitos de la Fruta en Agrocalidad, bajo la autoría de Cristian Alexander Muyulema Taco y Gonzalo Darío Mariño Barriga; por lo que se autoriza ejecutar los trámites legales para su sustentación.

Es todo cuanto informar en honor a la verdad; en Riobamba, a los cuatro días del mes de Diciembre de 2025.



Leonardo Fabian Rentería Bustamante

C.I: 1104064132

## CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación Desarrollo de un sistema de clasificación basado en visión artificial para detección temprana de plagas de especies predominantes de mosquitos de la fruta en agrocalidad, presentado por Gonzalo Darío Mariño Barriga, con cédula de identidad número 0605066356 y Cristian Alexander Muyulema Taco, con cédula de identidad número 0302725171, bajo la tutoría de PhD. Leonardo Fabián Rentería Bustamante; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 5 de Enero del 2026.

PhD. Luis Patricio Tello Oquendo.  
**PRESIDENTE DEL TRIBUNAL DE GRADO**



PhD. Daniel Antonio Santillan Haro.  
**MIEMBRO DEL TRIBUNAL DE GRADO**



PhD. Manuel Antonio Meneses Freire.  
**MIEMBRO DEL TRIBUNAL DE GRADO**





# CERTIFICACIÓN

Que, **Muyulema Taco Cristian Alexander** con CC: **0302725171** y **Mariño Barriga Gonzalo Dario** con CC: **0605066356**, estudiantes de la Carrera **Telecomunicaciones**, Facultad de **Ingeniería**; han trabajado bajo mi tutoría el trabajo de investigación titulado " **DESARROLLO DE UN SISTEMA DE CLASIFICACIÓN BASADO EN VISIÓN ARTIFICIAL PARA DETECCIÓN TEMPRANA DE PLAGAS DE ESPECIES PREDOMINANTES DE MOSQUITOS DE LA FRUTA EN AGROCALIDAD**", cumple con el 3 % de similitud y 10 % de Inteligencia Artificial, de acuerdo con el reporte del sistema Anti plagio **Compilatio**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 15 de diciembre de 2025



---

PhD. Leonardo Fabian Rentería  
**TUTOR(A)**

## DEDICATORIA

A mis padres, Mirian Barriga y Gonzalo Mariño, cuya entrega, cariño y apoyo han sido la base sobre la cual he construido cada meta en mi vida. Su confianza en mí, incluso cuando las cosas parecían difíciles, me dio la fuerza para seguir adelante. Gracias por mostrarme, con su ejemplo, que la constancia y el esfuerzo siempre encuentran recompensa. Este logro refleja todo lo que han sembrado en mí.

A mis tías, Adriana Mariño y Narcisa Barriga, por su presencia incondicional y por brindarme siempre un espacio de apoyo y comprensión. Sus palabras, su cercanía y su cariño hicieron que este camino fuese más ligero y significativo. Gracias por acompañarme con tanta generosidad y por ser parte esencial de mi crecimiento.

A ustedes, con profundo agradecimiento, dedico este trabajo que representa un paso importante en mi vida y en el futuro que estoy construyendo.

*Gonzalo Mariño*

A mis padres, Martha y Juan, porque ustedes fueron el inicio de todo. Gracias por creer en mí incluso antes de que yo entendiera mi propio camino. Gracias por cada sacrificio, por cada palabra de aliento, y por enseñarme a levantarme siempre. Todo esto es fruto del amor inmenso que me han dado.

A mis hermanos Andrés y Daniela, y a mi sobrino David, por llenar mi vida con alegría y recordarme, en cada momento, que siempre hay un motivo para sonreír y seguir adelante.

A mi tío Carlos, mi ejemplo, mi guía. Gracias por enseñarme que no importa cuán difícil sea el camino, siempre vale la pena continuar. Este logro nace también de tus enseñanzas y de la luz que siempre has puesto en mi vida.

A mis abuelos María, Jesús y Bolívar, porque en su amor encontré raíces firmes y en su sabiduría encontré dirección. Sus abrazos y consejos me han acompañado en cada etapa.

A mi novia Ingrid, por ser mi compañía fiel en este camino lleno de retos. Gracias por tu paciencia, tu comprensión y por sostenerme en los momentos en los que sentí que no podía más.

A la memoria de mi tía Ana, este logro también es tuyo, porque lo construí con el amor que dejaste en mi vida.

A cada uno de ustedes, que me han dado motivos para luchar y sueños para perseguir, les dedico este trabajo con el corazón.

*Cristian Muyulema*

## AGRADECIMIENTO

Agradezco primeramente a mis padres, Mirian Barriga y Gonzalo Mariño, quienes han sido mi mayor inspiración y la base de todo lo que soy. Gracias por su amor incondicional, por cada sacrificio y por enseñarme que la perseverancia y la humildad abren caminos incluso en los momentos más difíciles. Su apoyo constante, sus palabras de aliento y su ejemplo de vida han sido fundamentales para alcanzar este logro, que también les pertenece.

A mis hermanos, por su compañía, su cariño y por recordarme siempre la importancia de mantenerse firme y seguir adelante.

Extiendo un agradecimiento muy especial a mis tías Adriana Mariño y Narcisa Barriga, cuyo apoyo, afecto y confianza han sido un pilar en mi vida. Gracias por estar presentes en cada etapa de este camino y por brindarme siempre un impulso adicional para continuar.

A mis amigos Marco Parada, Cristian Muyulema, Gabriel Pazmiño, quienes me acompañaron con su amistad sincera, sus palabras motivadoras y su apoyo en los momentos más intensos del proceso. Gracias por hacer de este recorrido una experiencia más ligera y significativa.

*Gonzalo Mariño*

Quiero expresar mi más profundo agradecimiento a todas las personas que, de una u otra manera, hicieron posible este trabajo.

A mi familia, por ser el sostén emocional y espiritual que me mantuvo firme en cada desafío. Gracias por celebrar mis avances y alentarme cuando el camino se hacía cuesta arriba. A mis profesores y mentores, por compartir su conocimiento con generosidad y por guiarme con paciencia durante este proceso.

A mis amigos Marco, Yagui, Gonzalo, Miguel, por cada conversación, por cada gesto de apoyo y por recordarme que las metas se construyen mejor acompañado.

Y a todas las personas que, de alguna manera, aportaron luz a mis días durante este recorrido, mi gratitud sincera. Cada uno dejó una huella que también está en estas páginas.

*Cristian Muyulema*

## ÍNDICE GENERAL

DECLARATORIA DE AUTORÍA	
DICTAMEN FAVORABLE DEL PROFESOR TUTOR	
CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL	
CERTIFICADO ANTIPLAGIO	
DEDICATORIA	
AGRADECIMIENTO	
ÍNDICE GENERAL	
ÍNDICE DE TABLAS	
ÍNDICE DE FIGURAS	
RESUMEN	
ABSTRACT	
CAPITULO I. INTRODUCCIÓN	17
1.1 PLANTEAMIENTO DEL PROBLEMA.	18
1.2 JUSTIFICACIÓN	19
1.3 OBJETIVOS	20
1.3.1 General	20
1.3.2 Específicos	20
CAPÍTULO II. MARCO TEÓRICO	21
2.1 Entomología Económica	21
2.1.1 Moscas De La Fruta	21
2.1.2 Especies De Importancia Cuarentenaria en el Ecuador	22
2.1.3 Morfología Clave De Las Especies Objetivo	23
2.1.4 Métodos Tradicionales de Detección de Plagas	24
2.1.5 Inteligencia Artificial	25
2.1.6 Tecnologías Y Algoritmos De Visión Artificial	25
2.1.7 Visión Artificial En La Agricultura	26
2.1.8 Detección Temprana De Plagas En La Agricultura De Precisión	26
2.1.9 Arquitectura Yolo Para La Detección Y Clasificación De Plagas En Tiempo Real	27
2.1.10 Uso De Dispositivos Embebidos e IOT en Agricultura	28
2.2 Estudio Del Estado Del Arte	28
CAPÍTULO III. METODOLOGÍA	33
3.1 Tipo de Investigación	33
3.1.1 Diseño de la Investigación	33
3.1.2 Técnicas de Recolección de Datos	33
3.1.3 Población de estudio	34



3.1.4	Operacionalización de las variables .....	35
3.1.5	Fase Uno .....	35
3.1.6	Fase Dos .....	38
3.1.7	Fase Tres.....	39
3.1.8	Fase Cuatro .....	55
<b>CAPITULO IV. RESULTADOS Y DISCUSIÓN .....</b>		<b>62</b>
4.1	RESULTADOS .....	62
4.1.1	Prueba de normalidad.....	62
4.1.2	Prueba U de Mann-Whitney para muestras independientes. ....	62
4.1.3	Resultados y Análisis Comparativo del Tiempo según el Nivel de Conocimiento. .....	63
<b>CAPITULO V CONCLUSIONES Y RECOMENDACIONES .....</b>		<b>65</b>
5.1	CONCLUSIONES .....	65
5.2	RECOMENDACIONES .....	66
<b>BIBLIOGRAFIA .....</b>		<b>67</b>
<b>ANEXOS .....</b>		<b>71</b>

## ÍNDICE DE TABLAS

Tabla 1. Morfología de las especies objetivo. ....	23
Tabla 2. Características YOLOv11.....	27
Tabla 3. Parámetros y Métodos de Evaluación. ....	35
Tabla 4. Comparativa de plataformas de procesamiento .....	36
Tabla 5. Comparativa de modelos de clasificación .....	37
Tabla 6. Composición Cuantitativa del Dataset .....	38
Tabla 7. Rendimiento del Modelo YOLOv11 en sus diferentes variantes. ....	46
Tabla 8. Parámetros de Entrenamiento. ....	48
Tabla 9 Comparación entre el método tradicional y el sistema basado en visión artificial YOLOv11. ....	57
Tabla 10. Prueba de Normalidad Shapiro-Wilk N=36.....	62
Tabla 11. Prueba U de Mann-Whitney para muestras independientes N=36. ....	63
Tabla 12. Prueba de Kruskal-Wallis para muestras independientes N=32. ....	64

## ÍNDICE DE FIGURAS

Figura 1. Ceratitis capitata (Mosca de la fruta del Mediterráneo).....	23
Figura 2. Pipeline del Sistema. ....	29
Figura 3. Componentes electrónicos de la trampa electrónica. (A) Trampa electrónica con panel solar, pantalla Stevenson para proteger el sensor de temperatura y humedad relativa. (B) Cámara colocada frente a una trampa Rimi®. (C) Batería y Electrónica. ....	29
Figura 4. Imagen del prototipo de trampa electrónica McPhail colgado de un árbol de cítricos. ....	30
Figura 5. Manejo de moscas.....	31
Figura 6. Comparativa de la tasa de aciertos de los modelos.....	31
Figura 7. (A, B) representan dos conjuntos diferentes de muestras y, después de la segmentación semántica y la segmentación del umbral, se puede extraer una posición y forma más claras del líquido regurgitado.....	32
Figura 8. Etapas del Desarrollo del Prototipo. ....	33
Figura 9. Proceso de etiquetado en RoboFlow.....	39
Figura 10. Diagrama de Bloques del Dispositivo.....	40
Figura 11. Tarjeta de Red Externa.....	42
Figura 12. Estuche acrílico.....	42
Figura 13. Jetson Nano sobre la base inferior del estuche acrílico.....	43
Figura 14. Sistema de ventilación.....	43
Figura 15. Conexión botones de encendido y reinicio.....	44
Figura 16. Cámara IMX219-77.....	44
Figura 17. Pantalla LCD.....	45
Figura 18. Nvidia SDK Manager.....	45
Figura 19. Conexión de la alimentación de la Nvidia Jetson Nano.....	46
Figura 20. Instalación de dependencias. ....	48
Figura 21. importación del dataset a Google Colab.....	49
Figura 22. Proceso de entrenamiento.....	50
Figura 23. Como descargar el modelo final desde Google Colab. ....	51
Figura 24. Conjunto de Entrenamiento.....	52
Figura 25. Conjunto de Validación.....	52
Figura 26. Pantalla inicial.....	53
Figura 27. Interfaz de captura de datos y generación de reportes. ....	54
Figura 28. Interfaz de visualización de resultados y re etiquetado de imágenes.....	54
Figura 29. Pruebas. ....	55
Figura 30. Prueba Manual. ....	56
Figura 31. Prueba Automática.....	56
Figura 32. Curva Precisión–Recall. ....	58
Figura 33. Recall vs. umbral de confianza.....	58
Figura 34. Matriz de confusión.....	59
Figura 35. Matriz de confusión normalizada.....	59
Figura 36. Conjunto de datos y distribución de cajas. ....	60

Figura 37. Curvas de entrenamiento y validación. ....	60
Figura 38. F1 vs. umbral de confianza.....	61
Figura 39. Precisión vs. umbral de confianza.....	61
Figura 40. Diagrama de caja del Tiempo por Nivel_Conocimiento.....	64

## RESUMEN

Las moscas de la fruta (Diptera: Tephritidae) se encuentran entre las principales amenazas fitosanitarias para la agricultura a nivel mundial, causando pérdidas económicas considerables en el Ecuador. La identificación temprana y precisa de las especies predominantes es esencial para la ejecución efectiva de programas de control y cuarentena liderados por organismos como Agrocalidad. Los métodos tradicionales de vigilancia son laboriosos, dependen de la pericia humana y pueden retrasar la toma oportuna de decisiones.

Esta tesis presenta el desarrollo de un sistema de clasificación basado en visión artificial para la detección temprana de las principales especies de moscas de la fruta relevantes en las operaciones de Agrocalidad. El objetivo es proporcionar una herramienta automatizada y eficiente que optimice los procesos de inspección. Se empleó un enfoque de aprendizaje profundo mediante una Red Neuronal Convolucional (CNN), entrenada y validada con un conjunto de datos etiquetados de alta calidad de *Ceratitis capitata*, *Anastrepha serpentina* y *Anastrepha striata*. El sistema también fue optimizado para su implementación en una NVIDIA Jetson Nano, garantizando eficiencia computacional para uso en campo o laboratorio.

Los resultados experimentales mostraron un desempeño sólido, alcanzando una precisión del 99.0% y una velocidad de inferencia de 7.3450 segundos por imagen, adecuada para monitoreo en tiempo casi real. Estos hallazgos demuestran el potencial del sistema para fortalecer la detección de plagas y los programas fitosanitarios en el Ecuador.

**Palabras Clave:** Visión Artificial, Detección de Plagas, Mosca de la Fruta, Red Neuronal Convolucional (CNN), Aprendizaje Profundo, Agrocalidad.

## ABSTRACT

Fruit flies (Diptera: Tephritidae) are among the most significant phytosanitary threats to global agriculture, causing considerable economic losses in Ecuador. Early and accurate identification of predominant species is essential for the effective execution of control and quarantine programs led by agencies such as Agrocalidad. Traditional surveillance methods are labor-intensive, depend on human expertise, and may delay timely decision-making.

This thesis presents the development of a computer-vision-based classification system for the early detection of the main fruit fly species relevant to Agrocalidad's operations. The objective is to provide an automated and efficient tool that optimizes inspection processes. A Deep Learning approach was applied, using a Convolutional Neural Network (CNN) trained and validated on a high-quality labeled dataset of *Ceratitis capitata*, *Anastrepha serpentina*, and *Anastrepha striata*. The system was also optimized for deployment on an NVIDIA Jetson Nano to ensure computational efficiency for field or laboratory use.

Experimental results showed strong performance, achieving 99.0% classification accuracy and an inference time of 7.3450 seconds per image, suitable for near-real-time monitoring. These findings demonstrate the potential of the proposed system to improve pest detection and strengthen phytosanitary programs in Ecuador.

**Keywords:** Computer Vision, Pest Detection, Fruit Fly, Convolutional Neural Network (CNN), Deep Learning, Agrocalidad.



Firma de aceptación/validación por:  
**SOFIA  
FERNANDA  
FREIRE  
CARRILLO**  
validar documento con QRcode

Reviewed by:

Mgs. Sofia Freire Carrillo

**ENGLISH PROFESSOR**

C.C. 0604257881

## CAPITULO I. INTRODUCCIÓN

En el Ecuador, las moscas de la fruta representan un problema constante para los cultivos frutales, ya que dañan los frutos y reducen su calidad. Entre las especies de interés económico más comunes se encuentran *Anastrepha Striata*, *Anastrepha Serpentina* y *Ceratitis Capitata* [1]. Estas especies se consideran de alto impacto económico por su facilidad para infestar diferentes hospederos y por los daños que ocasionan en cultivos de importancia comercial, de esta manera no solo afectan la producción, sino que también pueden impedir la exportación de fruta debido a las normas fitosanitarias que deben cumplir los productores. Los documentos oficiales emitidos por AGROCALIDAD explican que el monitoreo permanente es esencial para mantener bajo control a esta plaga y evitar pérdidas económicas [2].

El sistema de monitoreo oficial se basa en trampas como las McPhail y Jackson, que permiten obtener datos para calcular el índice MTD (Mosca/Trampa/Día), el cual indica si la población de moscas se encuentra dentro de parámetros aceptables. No obstante, este proceso depende de inspecciones manuales y del análisis visual de los especímenes, lo que implica un esfuerzo considerable y tiempos prolongados antes de tomar decisiones. En zonas donde la producción es intensiva o donde las condiciones ambientales favorecen el desarrollo de la plaga, los métodos tradicionales pueden resultar insuficientes para detectar incrementos poblacionales de manera rápida. [3].

En este escenario, tecnologías como la visión artificial y el machine learning ofrecen nuevas oportunidades para automatizar y mejorar el proceso de identificación de insectos. Estas herramientas permiten analizar imágenes y reconocer especies según sus características visuales, logrando una clasificación rápida y consistente. Además, la incorporación de dispositivos, como el Jetson Nano, permite el desarrollo de sistemas portátiles que pueden utilizarse directamente en campo. La presente investigación propone el diseño de un sistema de detección automática de moscas de la fruta basado en imágenes, con el objetivo de apoyar las labores de monitoreo, reducir el tiempo de respuesta y contribuir a un manejo fitosanitario más eficiente [4].

## **1.1 PLANTEAMIENTO DEL PROBLEMA.**

El monitoreo tradicional de moscas de la fruta enfrenta varias limitaciones que afectan su eficiencia y rapidez. En la práctica, los técnicos deben visitar periódicamente los sitios de producción, revisar manualmente las trampas instaladas y clasificar cada uno de los insectos encontrados. Este trabajo demanda tiempo, recursos económicos y personal capacitado, además de estar expuesto a errores humanos por fatiga, condiciones del clima o exceso de muestras. A esto se suma que las muestras suelen necesitar confirmación en laboratorio, lo que extiende aún más el tiempo entre la captura del insecto y la obtención del resultado final [2].

En cultivos de exportación como el mango, pequeñas variaciones en el índice MTD pueden provocar sanciones inmediatas, como la suspensión de la cosecha de un lote o la obligación de aplicar planes de manejo intensivos. Cuando la identificación no se realiza a tiempo, la plaga puede seguir creciendo sin ser detectada, lo cual incrementa el riesgo de que los niveles superen los límites permitidos por la normativa. Además, factores como la presencia de frutos caídos, el manejo insuficiente del predio o la existencia de hospederos cercanos pueden aumentar la presión de la plaga de manera inesperada, lo que hace aún más necesario contar con herramientas que brinden información inmediata [3], [5].

A pesar de los avances en tecnología agrícola, actualmente no existe un sistema accesible que pueda identificar de forma automática y precisa las especies de moscas de la fruta a partir de imágenes tomadas en campo. Los métodos basados en machine learning aún no se han implementado en los programas oficiales de monitoreo, a pesar de su potencial para reducir errores, acelerar el análisis y proporcionar datos en tiempo real. Esta ausencia tecnológica limita la capacidad de respuesta de los productores y técnicos, genera demoras en la toma de decisiones y puede ocasionar pérdidas económicas considerables. Por ello, es necesario desarrollar un sistema inteligente que ofrezca una alternativa práctica para el reconocimiento rápido de moscas de la fruta y que apoye las actividades del manejo integrado de plagas [4], [6].



## 1.2 JUSTIFICACIÓN

La creación de un sistema que utilice machine learning para identificar moscas de la fruta es importante porque permite mejorar la forma en que se realiza el monitoreo fitosanitario. Al automatizar la clasificación de las especies mediante imágenes, es posible obtener resultados en poco tiempo y detectar problemas antes de que la plaga alcance niveles que afecten la producción. Esto ayuda a los técnicos y productores a actuar con rapidez y evitar que la plaga supere los valores permitidos que pueden detener la cosecha o impedir la exportación [4], [6].

El uso de tecnologías como cámaras y dispositivos de bajo costo ofrece una ventaja adicional, ya que puede implementarse en zonas donde no hay laboratorios disponibles. De esta manera, la identificación se vuelve más accesible y menos dependiente del análisis manual. Esto contribuye a que el monitoreo sea más preciso, uniforme y rápido, mejorando así la eficiencia del manejo integrado de moscas de la fruta.

Además, este tipo de solución impulsa el desarrollo de nuevas herramientas tecnológicas para el sector agrícola, lo que fortalece los programas de vigilancia y permite avanzar hacia una agricultura más moderna y basada en datos. El sistema complementa los métodos tradicionales y ofrece información que puede evitar pérdidas económicas y mejorar la planificación de las actividades de campo. Su aplicación también puede extenderse a otros cultivos y plagas, lo que abre oportunidades para futuros desarrollos en el área.

### **1.3 OBJETIVOS**

#### **1.3.1 General**

Desarrollo de un Sistema de clasificación basado en Visión Artificial para detección temprana de plagas de Especies predominantes de Mosquitos de la Fruta en Agrocalidad.

#### **1.3.2 Específicos**

- Investigar y seleccionar hardware y software óptimo para implementar un sistema de clasificación de especies de mosquitos de la fruta basado en machine learning.
- Investigar y seleccionar las técnicas de machine learning más adecuadas para la clasificación de especies de mosquitos de la fruta.
- Desarrollar e implementar un sistema de clasificación de especies de mosquitos de la fruta utilizando la técnica de machine learning seleccionada.
- Evaluar el funcionamiento del sistema con pruebas de campo.

## **CAPÍTULO II. MARCO TEÓRICO**

### **2.1 Entomología Económica**

La entomología económica es la rama de la entomología que estudia los insectos que causan efectos directos o indirectos en las actividades humanas, especialmente en la agricultura, la industria alimentaria, la salud pública y los sistemas de almacenamiento. Según los textos de formación entomológica [7], esta disciplina está orientada a comprender la biología, ecología, comportamiento y mecanismos de daño de las especies consideradas plagas, con el fin de desarrollar estrategias de manejo integrado de plagas (MIP) que reduzcan las pérdidas económicas y aseguren la sostenibilidad productiva.

Entre los elementos fundamentales analizados por la entomología económica se encuentran los umbrales económicos, el impacto en el rendimiento y la calidad de los productos agrícolas, así como los costos derivados del monitoreo, control químico, control biológico y cumplimiento de las normativas fitosanitarias. Según la guía técnica del OIEA [8], esta disciplina integra conocimientos de biología, climatología, economía agrícola y tecnologías aplicadas para diseñar sistemas de prevención y manejo que mitiguen el daño a los cultivos comerciales y reduzcan las pérdidas postcosecha.

La entomología económica, en el contexto ecuatoriano, es importante porque el país tiene diversidad agrícola y existen plagas cuarentenarias de alto impacto (*Ceratitis capitata*, *Anastrepha striata* y *Anastrepha serpentina*). Estas especies afectan los cultivos destinados a la exportación y al consumo interno, causan restricciones fitosanitarias, aumentan los costos de producción y representan una amenaza continua para los pequeños y medianos productores, sugiere Agrocalidad [9]. Por lo tanto, el país implementa programas de vigilancia, control integrado y regulación fitosanitaria destinados a minimizar su impacto.

#### **2.1.1 Moscas De La Fruta**

##### **Ciclo De Vida**

Los factores en estos ecosistemas, incluyendo la temperatura, la humedad, la vegetación nativa, la fruta para la oviposición, el sustrato de pupación y la disponibilidad de alimentos, regulan estrictamente el ciclo de vida [9].

El ciclo ocurre de la siguiente manera: una hembra fertilizada inserta su ovipositor en una fruta que está madurando y pone huevos. Una vez que las larvas salen, se alimentan de la pulpa de la fruta hasta completar tres etapas larvarias. Cuando maduran, caen al suelo y se convierten en pupas. El adulto crece y, después de un tiempo, emerge de la pupa y comienza un nuevo ciclo. Los géneros presentes son *Anastrepha*, *Ceratitis* y *Toxotrypana*, y son multivoltinos, es decir, tienen varias generaciones por año. También son generalistas en cuanto al número de plantas hospedadoras que atacan y su biología gira en torno a dos o más especies de frutas [9].



### 2.1.2 Especies De Importancia Cuarentenaria en el Ecuador

AGROCALIDAD ha logrado localizar 34 especies del género *Anastrepha* en Ecuador mediante el PNMMF hasta 2015. A estas moscas se suma la especie introducida *Ceratitis capitata*, más comúnmente conocida como la mosca de la fruta del Mediterráneo, y *Toxotrypana recurcauda*. Las especies más significativas y comunes, considerando aspectos de distribución, importancia económica, rango de hospedadores y el daño que causan, son: *Anastrepha fraterculus*, *A. striata*, *A. serpentina*, *A. obliqua* y *Ceratitis capitata*, de las cuales se han seleccionado las siguientes tres especies [1], [9].

- ***Ceratitis Capitata*.** - Se considera una de las moscas de la fruta más invasivas y polífagas a nivel mundial. Su amplia distribución en las áreas costeras y de la sierra de Ecuador se debe a su gran adaptabilidad, fertilidad y capacidad para colonizar diversos microclimas. Esta especie ataca a más de 200 hospedadores, incluidos cultivos de alto valor como el durazno, mango, tomate de árbol y cítricos, lo que la convierte en una plaga de importancia económica y cuarentenaria [9].
- ***Anastrepha Striata*.** - Es una de las moscas de la fruta más importantes en Ecuador, especialmente debido a su fuerte asociación con la guayaba, un cultivo de importancia económica para los pequeños productores rurales. Esta especie está ampliamente distribuida en las regiones tropicales y subtropicales del país, donde las temperaturas cálidas favorecen su reproducción y supervivencia. También puede atacar otros hospedadores como el babaco y algunos cítricos [9].
- ***Anastrepha Serpentina*.** - Es una mosca de la fruta de importancia cuarentenaria en Ecuador debido a su capacidad para dañar árboles frutales de alta demanda como el aguacate, zapote, cítricos y otras especies tropicales. Su presencia se ha registrado principalmente en áreas cálidas y húmedas, donde encuentra condiciones ideales para la reproducción. Esta especie tiene un ciclo biológico eficiente que facilita brotes recurrentes en huertos poco mecanizados [9].

### 2.1.3 Morfología Clave De Las Especies Objetivo

*Tabla 1. Morfología de las especies objetivo.*

Especie	Rasgos Morfológicos Distintivos
<p><b>Ceratitis capitata</b></p>  <p><i>Figura 1. Ceratitis capitata (Mosca de la fruta del Mediterráneo)</i></p>	<ul style="list-style-type: none"> <li>• Alas: Poseen un patrón de manchas y bandas oscuras muy característico, con bandas transversales de color amarillo/pardo claro y una mancha oscura en el ápice alar.</li> <li>• Tórax: Presenta un dibujo bien definido en blanco y negro (mármol).</li> <li>• Abdomen: Color pardo-amarillento, a veces con bandas oscuras.</li> </ul>
<p><b>Anastrepha Serpentina</b></p>  <p><i>Figura 2. Anastrepha Serpentina.</i></p>	<ul style="list-style-type: none"> <li>• Alas: Se caracterizan por tener un patrón de bandas amarillentas o marrones con forma de "S" o de "V" invertida, con la vena R4+5 presentando una mancha oscura que se extiende hacia el margen.</li> <li>• Ovopositor: Largo y delgado, lo que le permite alcanzar la pulpa de frutos grandes.</li> </ul>

---

### Anastrepha Striata

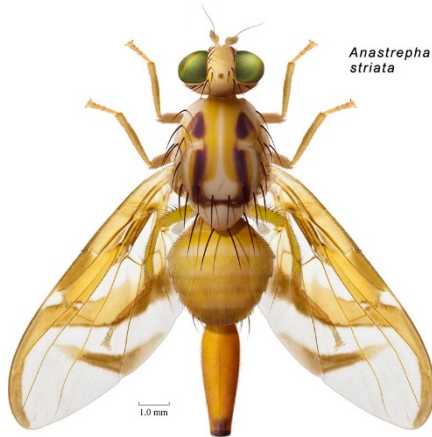


Figura 3. *Anastrepha striata* (Mosca de la guayaba).

- Alas: Su patrón es similar al de otras *Anastrepha*, pero la banda costal (a lo largo del borde del ala) es más ancha y la banda S (en forma de 'S') es más marcada y continua que en otras especies.
- Cuerpo: Generalmente de coloración más clara (amarillo-rojiza) que otras especies.
- Ovipositor: Es una de las características clave para distinguir las especies de *Anastrepha*.

---

Fuente:[1], [10], [11].

#### 2.1.4 Métodos Tradicionales de Detección de Plagas

Durante décadas, los métodos comunes de detección de plagas han sido el pilar de la agricultura (inspecciones visuales y trampas adhesivas y todo eso), incluyendo la detección de insectos mediante inspecciones visuales. Las inspecciones visuales son procesos laboriosos y subjetivos en los que los agricultores o técnicos inspeccionan los cultivos en busca de signos visibles de infestación de plagas. Este trabajo puede ser inconsistente y falible, dependiendo en gran medida de la experiencia del inspector, de ahí la eficiencia de tales inspecciones. Además, estas inspecciones son poco prácticas en grandes áreas de cultivo por ser laboriosas y consumir mucho tiempo[12].

En contraste, las trampas adhesivas son instrumentos pasivos que capturan insectos voladores, dando una pista de la actividad de plagas. Sin embargo, estas trampas requieren mantenimiento constante y no son conocidas por ser efectivas para la detección temprana, ya que monitorean las plagas solo una vez que ya han comenzado a colonizar el área. Cuando estos métodos tradicionales se comparan con métodos automáticos basados en visión artificial, los sistemas automáticos, según estudios, son más rápidos y precisos para la identificación de plagas, proporcionando monitoreo continuo y alertas tempranas de infestación [13].

### 2.1.5 Inteligencia Artificial

La inteligencia artificial (IA) es una tecnología que permite a las computadoras y máquinas imitar la capacidad humana de aprender, comprender, resolver problemas, tomar decisiones, ser creativos y ser autónomos [14].

La IA ha surgido como un método poderoso para la agricultura con un monitoreo preciso de cultivos y detección temprana de plagas, lo que ayuda a reducir pérdidas económicas y minimizar el uso excesivo de pesticidas. Las tecnologías de inteligencia artificial (IA) como sensores y drones permiten a los agricultores tomar decisiones informadas basadas en datos en tiempo real, ayudando a usar los recursos de manera eficiente y aumentando la productividad [15].

- **Machine learning.**

El aprendizaje automático es un subcampo de la inteligencia artificial que entrena algoritmos para predecir o hacer algunas predicciones sobre datos sin un guion o programación explícita. Hay varios enfoques, por ejemplo, árboles de decisión, regresión, bosques aleatorios, SVM, KNN y agrupamiento para diferentes problemas. Uno de los más utilizados son las redes neuronales artificiales que imitan el cerebro humano y son útiles para el reconocimiento de patrones en grandes cantidades de datos. El más simple es el aprendizaje supervisado; aquí, el modelo aprende de datos etiquetados para identificar las relaciones entre la entrada y la salida de un modelo, obteniendo así una clasificación y/o predicción precisa de nuevos casos [14].

- **Deep learning.**

El aprendizaje profundo es un tipo de aprendizaje automático basado en múltiples capas de redes neuronales profundas que son capaces de aprender sin supervisión directa a partir de grandes cantidades de material no estructurado. Esta tecnología es la base de la mayoría de las aplicaciones modernas de inteligencia artificial, como la visión por computadora y el procesamiento del lenguaje natural [14].

### 2.1.6 Tecnologías Y Algoritmos De Visión Artificial

La creación de un sistema de clasificación basado en visión artificial para la detección de plagas es muy tecnológico y orientado al aprendizaje automático. Las redes neuronales convolucionales (CNNs) son ideales para el análisis y clasificación de imágenes agrícolas, ya que las arquitecturas pueden aprender y comprender patrones complejos y las características. Tales redes consisten en varias capas compuestas por la extracción de características de las imágenes, que pueden usarse para clasificarlas de manera efectiva [16]. Además, el método de segmentación de imágenes es un método utilizado para separar ciertas áreas de una imagen para identificar plagas en un entorno complejo [17].

La implementación práctica real de estos sistemas en el campo debe estar respaldada por el hardware adecuado y el software más reciente. Cámaras de alta resolución y sensores multiespectrales de drones pueden capturar una instantánea extendida de un cultivo real, que se procesa utilizando el algoritmo de visión artificial, con el potencial de detectar plagas en

tiempo real [18]. Herramientas de software como TensorFlow, PyTorch se utilizan para desarrollar modelos de visión artificial y entrenarlos, mientras que las plataformas de monitoreo agrícola ponen a trabajar los datos almacenados para obtener una imagen definitiva sobre la salud del cultivo. Un enfoque combinado de estas tecnologías permite un monitoreo eficiente y una respuesta rápida a las infestaciones de plagas, mejorando la productividad y sostenibilidad de la agricultura [19].

### **2.1.7 Visión Artificial En La Agricultura**

Para los agricultores que se enfrentan a una gran cantidad de trabajo repetitivo que generalmente requiere esfuerzo humano para completarse, la visión por computadora es una tecnología revolucionaria en la agricultura moderna. La visión por computadora, al aplicar cámaras y algoritmos avanzados de procesamiento de imágenes, puede clasificar automáticamente plagas, enfermedades y otras causas de preocupaciones sobre la salud de los cultivos. Esta tecnología se puede utilizar para analizar grandes cantidades de datos visuales en tiempo real, ayudando así a tomar decisiones de manera más efectiva. Por ejemplo, las redes neuronales convolucionales (CNN) han demostrado gran precisión en la detección de enfermedades en hojas de tomate con una precisión superior al 90% [16], [20].

La visión por computadora juega un papel importante en el campo de la agricultura, donde aumenta la precisión y eficiencia de la detección de plagas y disminuye la dependencia de métodos manuales, que son más propensos a errores humanos y consumen más tiempo. Técnicas como la segmentación de imágenes se utilizan para aislar y analizar ciertas partes de una planta o una pieza de fruta, e identificar la presencia de plagas con mucha más precisión [21]. Además, el aprendizaje por transferencia apoya la modificación de modelos preentrenados para nuevas tareas con muestras más pequeñas de datos, lo que se realiza más fácilmente en entornos agrícolas menos explorados donde los datos etiquetados son escasos [22].

### **2.1.8 Detección Temprana De Plagas En La Agricultura De Precisión**

Con el fin de mejorar la eficiencia de la producción agrícola y el uso de recursos, se ha implementado un modelo de toma de decisiones con tecnología de análisis basada en precisión e información, para optimizar las decisiones inteligentes en la gestión de insumos y para monitorear y predecir la producción de cultivos utilizando tecnología avanzada. En este contexto, la detección temprana de plagas se considera uno de los componentes centrales, ya que desempeña un papel primordial en la reducción de pérdidas económicas y la minimización del consumo excesivo de pesticidas [23], [24], lo cual es crucial para la agricultura sostenible. Los métodos de monitoreo convencionales, que incluyen inspecciones visuales frecuentes y trampas adhesivas, han demostrado tener tanto una eficiencia limitada como una pobre oportunidad de respuesta. Por otro lado, los sistemas de agricultura de precisión utilizan sensores, redes inalámbricas, dispositivos de Internet de las Cosas (IoT) y algoritmos de inteligencia artificial para rastrear parámetros biológicos y ambientales relacionados con las actividades de las plagas a medida que ocurren [25].



Uno de los componentes cruciales para la detección temprana es la automatización del proceso de identificación mediante visión por computadora, lo que permite la adquisición, procesamiento e interpretación de imágenes de alta resolución obtenidas de trampas inteligentes o cámaras que se han desplegado en el campo [26]. Este método ayuda a acelerar y mejorar la precisión del diagnóstico y permite una respuesta rápida a las infestaciones de plagas antes de que alcancen niveles críticos. Además, los modelos predictivos analizan datos históricos y actuales para anticipar la propagación de plagas basándose en fenómenos ambientales (temperatura, humedad, estacionalidad). Así, instrumentos como las redes neuronales recurrentes (RNN) y los modelos híbridos de aprendizaje profundo pueden predecir la probabilidad de infestación basándose en datos climáticos y biológicos, reforzando aún más el sistema de alerta temprana [27].

### 2.1.9 Arquitectura Yolo Para La Detección Y Clasificación De Plagas En Tiempo Real

YOLO (You Only Look Once) es líder mundial en arquitectura de detección de objetos en tiempo real, caracterizada por su alta velocidad/efectividad mientras logra un buen equilibrio de alta precisión. Su novedad esencial es detectar en una sola pasada a través de la red. También hemos dividido nuestra imagen de entrada en una cuadrícula en una capa, donde cada celda estima al mismo tiempo la probabilidad del objeto y sus coordenadas. A diferencia de los métodos convencionales de múltiples etapas (por ejemplo, Faster-RCNN), este paradigma tiene la ventaja de procesar muy rápidamente. Por razones como esta, YOLO es lo suficientemente potente como para servir en aplicaciones de monitoreo de campo y es adecuado para marcos de optimización de inferencia como PyTorch y TensorRT [28].

La ventaja clave de YOLO es trabajar en dispositivos integrados y con pequeños recursos de computación, lo que permite la construcción de sistemas autónomos que no requieren acceso continuo a la nube, minimizando así la latencia y los costos. La arquitectura ha sido sustancialmente mejorada con las versiones más recientes (por ejemplo, YOLOv10 y YOLOv11) que incorporan detección sin anclas, modelos reparametrizables y mejor atención visual. Estas mejoras conducen a una mayor precisión (mAP) y a un tiempo de inferencia reducido, preservando a YOLO como la opción para ofertas rápidas y modulares de visión por computadora [29].

*Tabla 2. Características YOLOv11.*

Característica Clave de YOLOv11	Justificación
<b>Arquitectura <i>Anchor-Free</i></b>	A diferencia de versiones anteriores, YOLOv11 elimina las "cajas ancla" (anchor boxes). Esto simplifica el entrenamiento y es más efectivo para objetos con formas y tamaños muy variables (como insectos en distintas orientaciones y estados).

<b>Modelos Reparametrizables (<i>RepVGG-style blocks</i>)</b>	Permiten aumentar significativamente la velocidad de inferencia sin perder precisión. Es crucial para el despliegue en la Jetson Nano, ya que optimiza el uso de la memoria y la GPU Maxwell en un entorno de baja potencia.
<b>Eficiencia Computacional (FLOPS/Parámetros)</b>	El modelo YOLOv11 Large (25.3M de parámetros, 86.9 GFLOPs) ofrece un balance óptimo de recursos/precisión <sup>15</sup> , garantizando que se mantenga una alta velocidad de inferencia (< 3 segundos por imagen) mientras se mantiene la precisión del 99%.
<b>Visión Global y Localización Integrada</b>	YOLO procesa toda la imagen en una sola pasada, minimizando los errores de contexto que sufren las redes de dos etapas (como Faster R-CNN). Esto es vital para distinguir moscas muy cercanas en la trampa y evitar contarlas como una sola instancia.

**Fuente:** [28], [30].

#### 2.1.10 Uso De Dispositivos Embebidos e IOT en Agricultura

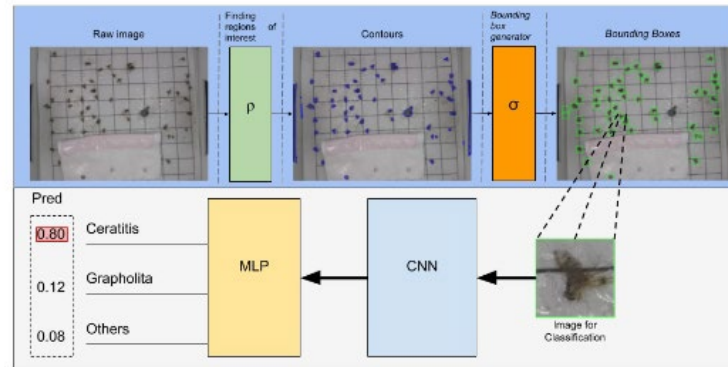
La combinación de dispositivos integrados y tecnología IoT (Internet de las Cosas) ha cambiado el rostro de la gestión agrícola al mecanizar y digitalizar el monitoreo de cultivos. Todas estas herramientas con sensores, cámaras y software de computación permiten la captura y procesamiento de datos en tiempo real. Las plataformas integradas juegan un papel clave en la gestión fitosanitaria, donde el despliegue de modelos de aprendizaje profundo (visión por computadora) directamente en el campo es vital. Esto tiene la ventaja de contar con instalaciones de procesamiento local para aumentar la eficiencia, reducir la latencia en la toma de decisiones y eliminar la dependencia de una conexión permanente a la nube, preservando así la privacidad de los datos [31].

Esta arquitectura IoT integrada es muy escalable, de esta manera, se pueden desplegar varios nodos de detección distribuidos geográficamente. La operación de los modelos de visión por computadora se realiza en estos mismos nodos y la información fluye a través de protocolos inalámbricos (LoRa, Wi-Fi u otros) hacia un servidor central compartido. Esta configuración no solo reduce los costos operativos y mejora la gestión fitosanitaria, sino que también ayuda a evitar la pérdida de tiempo, con la detección y registro en tiempo real de plagas como las moscas de la fruta. Permite la notificación automática de alertas automáticas y aumenta la resiliencia dentro de los sistemas agrícolas con más toma de decisiones [31].

## 2.2 Estudio Del Estado Del Arte

En [32], se observa el procedimiento para identificar al mosquito de la fruta consiste en utilizar trampas inteligentes que toman imágenes de los insectos que quedan atrapados.

Utiliza una Red Neuronal Artificial para clasificar, observar figura 2, una red neuronal convolucional (CNN) para aprender las características de los insectos y un procesamiento digital de imágenes para identificar áreas afectadas por plagas. Mientras que SqueezeNet presenta una precisión parecida pero con un tiempo de inferencia más corto, el modelo obtiene un 93.55% de precisión para *Ceratitis capitata* y un 91.28% para *Grapholita molesta* mediante ResNet18.



*Figura 2. Pipeline del Sistema.*

**Fuente:** [32]

En [33], se analiza la identificación y clasificación de la mosca del olivo con los algoritmos SVM (Support Vector Machine) y RF (Random Forest), que han sido incorporados en una trampa electrónica basada en Raspberry Pi B+. La combinación de ambas técnicas permite detectar la mosca del olivo con una precisión del 89,1%. Este porcentaje se eleva al 94,5% en el caso de SVM y al 91,9% en el de RF si comparamos todas las especies de moscas con otros insectos. El empleo de dispositivos IoT para clasificar imágenes crea nuevas oportunidades para optimizar recursos y proteger la privacidad, mientras que la aplicación efectiva de machine learning en este sistema de trampa electrónica brinda información importante para el control de plagas, observar figura 3.



*Figura 3. Componentes electrónicos de la trampa electrónica. (A) Trampa electrónica con panel solar, pantalla Stevenson para proteger el sensor de temperatura y humedad relativa. (B) Cámara colocada frente a una trampa Rimi®. (C) Batería y Electrónica.*

**Fuente:** [33].

En [34], se presentan métodos automáticos para monitorear plagas clave en manzanas, como *Cydia pomonella* y moscas de la fruta (Tephritidae y Drosophilidae), destacando el impacto del cambio climático en su comportamiento. Se evalúan sistemas basados en análisis de imágenes, trampas inteligentes y sensores que, combinados con IA y IoT, permiten una detección temprana y precisa, optimizando el manejo integrado de plagas. Estos sistemas automatizados reducen el uso de pesticidas y mejoran la eficiencia en el monitoreo, contribuyendo a una agricultura sostenible y adaptada a condiciones climáticas cambiantes.

En [35], sugieren usar Faster R CNN con la estructura de red ResNet50 para detectar y clasificar plagas, utilizando la trampa McPhail, observar figura 4 el conjunto de datos FF studio, que se especializa en insectos relevantes para la agricultura. La evaluación del modelo se realizó a través de métricas estándar, como el recall, la precisión, el IoU y la precisión media (AP), poniendo de manifiesto un rendimiento elevado en la detección de especies que presentan variaciones en forma, tamaño y condiciones de luz. Al emplear ResNet50 como extractor de características, se logró perfeccionar la detección de objetos diminutos y optimizar la habilidad general del sistema para el aprendizaje profundo. Asimismo, se destaca el beneficio de Faster R CNN por su estructura de dos fases, que fusiona propuestas regionales con una clasificación exacta, obteniendo así resultados más altos en comparación con los métodos tradicionales de visión computacional. La investigación subraya el potencial de integrar este tipo de modelos en sistemas de monitoreo automatizados de plagas para reducir costos y tiempos en comparación con la inspección manual. Asimismo, se enfatiza la utilidad de estas técnicas en el marco de la agricultura de precisión, donde la detección temprana de plagas es clave para minimizar daños en cultivos. Finalmente, los autores sugieren que la ampliación del dataset y el uso de técnicas de data augmentation podrían incrementar aún más la robustez del modelo en escenarios reales.

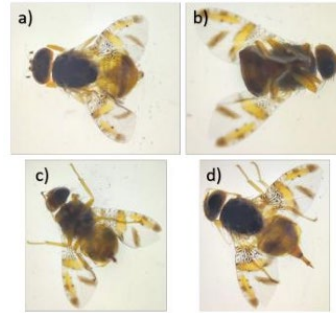


*Figura 4. Imagen del prototipo de trampa electrónica McPhail colgado de un árbol de cítricos.*

**Fuente:** [35].

En [36], muestran un método de aprendizaje por transferencia para clasificar imágenes de dos especies de moscas de la fruta (*Anastrepha fraterculus* y *Ceratitis capitata*), observar figura 5, que fueron tomadas con una cámara de teléfono móvil montada en un

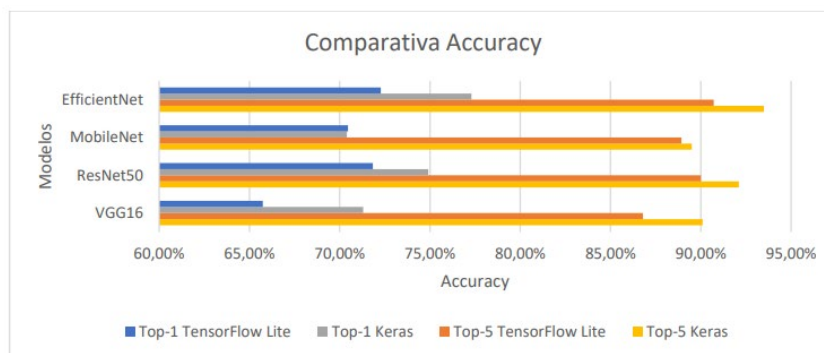
estereomicroscopio y segmentadas para separar áreas morfológicas importantes. Con puntuaciones F1, se examinaron tres modelos preentrenados (VGG16, VGG19 e Inception v3), logrando 82 % con los dos primeros y hasta 93 % con el tercero; este último sobresalió también en ambientes no controlados. Se empleó la técnica Grad CAM para observar las zonas fundamentales que afectan la clasificación, lo cual corroboró que el modelo tiene la capacidad de captar rasgos morfológicos importantes. Este trabajo demuestra que, con un dataset bien curado y un pipeline basado en aprendizaje por transferencia, se puede lograr una clasificación precisa y replicable de especies de moscas de la fruta.



*Figura 5. Manejo de moscas*

**Fuente:** [36].

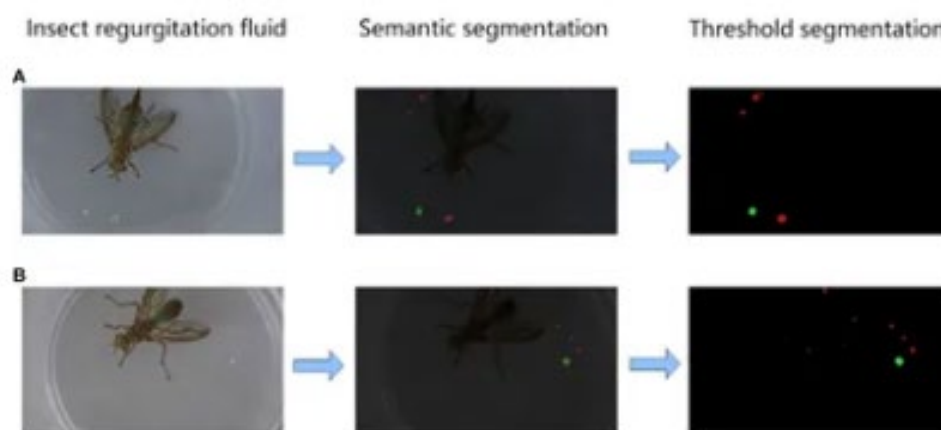
En [37], se estudia el reconocimiento de objetos mediante redes neuronales convolucionales con la finalidad de analizar su efectividad en las tareas de clasificación de imágenes. Este análisis tiene lugar en dispositivos móviles que poseen Android. Usando TensorFlow Lite, una herramienta diseñada para optimizar la inferencia en el límite de red, el proyecto examina modelos preentrenados como VGG16, ResNet50, MobileNet y EfficientNetB0, observar figura 6. Se pone en marcha una aplicación que, mediante la cámara del aparato, detecta elementos en tiempo real y contrasta su eficiencia energética, su precisión y los tiempos de ejecución. Aunque no se enfoca específicamente en la detección de insectos, el trabajo demuestra cómo la visión artificial y el aprendizaje profundo pueden aplicarse eficazmente en dispositivos embebidos, sentando las bases para futuras aplicaciones en entornos agrícolas o de monitoreo ambiental donde la clasificación automatizada y local es crucial.



*Figura 6. Comparativa de la tasa de aciertos de los modelos.*

**Fuente:** [37].

En [38], se muestra una variedad de métodos que utilizan técnicas de aprendizaje profundo y visión por computadora para analizar la regurgitación en las moscas de la fruta (Diptera: Tephritidae). Se sugieren tres técnicas diferentes, observar figura 7: el reconocimiento de regurgitación con I3D, la segmentación semántica utilizando U Net optimizado por el sistema de atención CBAM, y el seguimiento de trayectoria mediante la fusión de Yolov5 y DeepSort. La detección y el rastreo de insectos logran una exactitud del 99,8 %; la segmentación a través de U Net+CBAM logra un índice medio de unión (mIoU) del 90,96 %; y el modelo I3D obtuvo un Top 1 Accuracy del 96,3 %, por encima de C3D y X3D. Se empleó OpenCV y segmentación por umbral para cuantificar el líquido que fue regurgitado, lo cual permitió medir con exactitud el área. El uso conjunto de reconocimiento, segmentación y tracking posiciona esta metodología como una herramienta integral para analizar la regurgitación, aportando datos cuantitativos útiles para acciones fitosanitarias específicas en cultivos.



*Figura 7. (A, B) representan dos conjuntos diferentes de muestras y, después de la segmentación semántica y la segmentación del umbral, se puede extraer una posición y forma más claras del líquido regurgitado.*

**Fuente:** [38]

## CAPÍTULO III. METODOLOGÍA

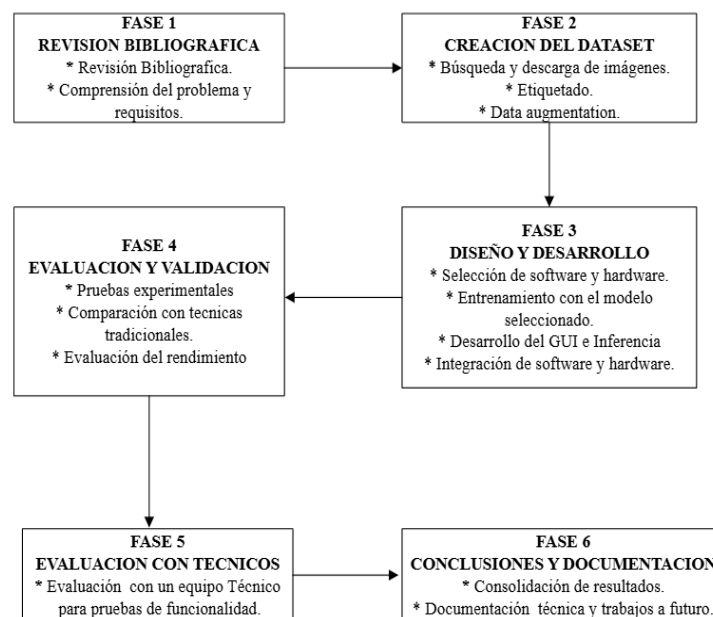
### 3.1 Tipo de Investigación

El presente trabajo corresponde a una investigación experimental aplicada, orientada al desarrollo de un sistema de clasificación de especies de moscas de la fruta mediante técnicas de visión artificial. El enfoque aplicado radica en el uso de un modelo de Deep Learning para resolver un problema concreto del ámbito agropecuario, específicamente el proceso de identificación temprana de plagas que afectan la producción frutícola nacional.

El trabajo se basa en el entrenamiento, ajuste y evaluación de un modelo de detección de objetos, comparando su desempeño con los métodos tradicionales empleados por la entidad Agrocalidad. De esta manera, se busca determinar la precisión de identificación y la reducción del tiempo de respuesta que ofrece el sistema propuesto frente al proceso manual de clasificación.

#### 3.1.1 Diseño de la Investigación

El procedimiento metodológico se estructuró en seis fases consecutivas, que permitieron organizar el desarrollo del proyecto desde la revisión teórica hasta la validación del sistema, observar figura 8.



*Figura 8. Etapas del Desarrollo del Prototipo.*

#### 3.1.2 Técnicas de Recolección de Datos

La recopilación de datos para la evaluación del sistema de clasificación se fundamentó en una combinación de técnicas que aseguraron la obtención de información directa, objetiva y continua, esenciales para validar el rendimiento del prototipo frente al método tradicional.

- **Observación**

La observación se empleó como técnica de recopilación de información directa en un entorno controlado (Laboratorio de Entomología de AGROCALIDAD) para evaluar el desempeño humano y del prototipo durante las pruebas experimentales.

Mediante la observación, se pudo registrar y analizar de forma directa:

- Se observó la interacción del personal técnico con el sistema desarrollado, incluyendo la manipulación de la interfaz gráfica (GUI) para la captura, detección, y generación de reportes.
- Se registró el Tiempo de Identificación (en segundos) que tomaban los participantes para realizar la clasificación manual de las moscas bajo un microscopio, lo cual sirvió de base para la comparación con el método automático.
- Se observó cómo el sistema realizaba la detección automática en las mismas muestras, registrando el tiempo de procesamiento y la exactitud de la clasificación, verificando su fiabilidad en un entorno de laboratorio.

- **Subsistema de Adquisición y Procesamiento de Datos**

Esta técnica une el procesamiento automatizado local con la intervención controlada del usuario. El propósito es producir resultados en tiempo real y objetivos a partir de las fotografías que se toman y cargan en el dispositivo.

El proceso comienza cuando el técnico de laboratorio o de campo emplea la cámara IMX219-77, que es la unidad sensora en este caso, para tomar una fotografía de la muestra y cargarla manualmente en la Unidad de Procesamiento Embebido usando la interfaz gráfica (GUI). El sistema actúa de manera independiente tras cargar la imagen: el modelo YOLOv11 es ejecutado por el NVIDIA Jetson Nano y produce automáticamente los datos de salida esenciales. Estos datos incluyen la Clasificación de la Especie identificada, el Nivel de Confianza asociado a cada detección, y el Tiempo de Inferencia que sería velocidad de procesamiento en milisegundos. Asegurándonos así que, aunque la entrada sea manual, la identificación, clasificación y el registro de métricas de rendimiento se ejecuten de manera automática y se almacenen en registros digitales para su trazabilidad.

### **3.1.3 Población de estudio.**

La población de estudio está compuesta por todas las mediciones posibles del tiempo de ejecución, que se logran dependiendo del método utilizado para identificar a las moscas de la fruta y el nivel de conocimiento del usuario. Estas mediciones fueron producidas durante las pruebas experimentales que se llevaron a cabo en los locales de Agrocalidad, en un ambiente controlado que posibilitó la normalización de las condiciones del sistema.

Esta población, desde una perspectiva estadística, representa el grupo teórico y sin límites de valores que se podrían alcanzar si cualquier usuario potencial empleara el sistema en



circunstancias parecidas a las del análisis. Como la cantidad de usuarios potenciales, así como las mediciones de tiempo y niveles de conocimiento, no tienen un límite preestablecido, se asumió que la población es infinita. Esta definición posibilita que se empleen métodos estadísticos adecuados para poblaciones grandes y hace más sencillo extender los resultados a un extenso conjunto de usuarios y escenarios.

### 3.1.4 Operacionalización de las variables

*Tabla 3. Parámetros y Métodos de Evaluación.*

Variable	Concepto	Tipo de variable	Técnicas e Instrumentación
<b>Independiente</b>			
Método	Forma en que se realiza la detección de las especies.	Cualitativa Nominal	Detección manual vs Detección Automática
<b>Dependiente</b>			
Tiempo de detección	Tiempo necesario para la identificación y clasificación de la muestra.	Cuantitativa Continua	Tiempo empleado por cada método de clasificación, medido en segundos.

En la tabla 3, se detalla la operacionalización de las variables objeto de estudio.

### 3.1.5 Fase Uno

La primera fase del proyecto comprende el estudio de los principios teóricos relacionados con la visión artificial y el aprendizaje profundo empleados para la identificación y clasificación de plagas agrícolas. En esta etapa, el objetivo es establecer las bases conceptuales que sustenten el desarrollo del sistema de clasificación orientado a la detección temprana de las especies de moscas de la fruta más comunes. Posteriormente, tras una reunión con el departamento de Entomología, se determinó que las especies de mayor relevancia para la investigación son *Ceratitis capitata*, *Anastrepha serpentina* y *Anastrepha striata*.

Se lleva a cabo un estudio de proyectos e investigaciones previas relacionadas con el uso de modelos de inteligencia artificial para el monitoreo y control de plagas, identificando las arquitecturas de redes neuronales, los métodos empleados y las plataformas informáticas utilizadas. Este análisis permitió establecer criterios técnicos para la selección de las herramientas más adecuadas para el sistema propuesto.

- **Comparación de plataformas de procesamiento**

Se comparan diversas plataformas de hardware con el objetivo de ejecutar modelos de visión artificial en tiempo real. Entre los factores considerados para la evaluación se incluyeron el costo, el consumo energético, la capacidad de procesamiento, la compatibilidad con frameworks de aprendizaje profundo y el soporte para aceleración mediante GPU.

*Tabla 4. Comparativa de plataformas de procesamiento*

<b>Plataforma</b>	<b>Procesador/ GPU</b>	<b>Compatibi lidad IA</b>	<b>Consumo energético</b>	<b>Costo aproximado</b>	<b>Ventajas principales</b>	<b>Desventajas principales</b>
Raspberry Pi 4	ARM Cortex-A72, sin GPU dedicada	Limitada (sin CUDA)	Bajo (5–7 W)	Bajo	Económica y accesible	Bajo rendimiento en IA
Google Coral	TPU Edge integrado	TensorFlow Lite	Bajo (5 W)	Medio	Alta velocidad en inferencias con TensorFlow	Soporte limitado a modelos específicos
NVIDIA Jetson Nano	Quad-Core ARM A57 + GPU 128 CUDA	PyTorch, TensorFlow, ONNX	Moderado (10 W)	Medio	Aceleración por GPU, soporte TensorRT, buen equilibrio entre potencia y costo	Requiere disipación térmica adecuada

**Fuente:** [39], [40].

La plataforma más adecuada para el sistema, de acuerdo con la comparación presentada en la Tabla 4, fue seleccionada como la NVIDIA Jetson Nano. Esta elección se fundamenta en su capacidad para ejecutar modelos complejos de detección en tiempo real mediante aceleración por GPU, así como en la posibilidad de utilizar librerías optimizadas como TensorRT. En conjunto, esta plataforma garantiza un equilibrio entre rendimiento, eficiencia energética y viabilidad económica.

- **Comparación de arquitecturas de redes neuronales convolucionales**

Se analizan diversas arquitecturas de redes neuronales convolucionales (CNN) comúnmente empleadas en la clasificación de imágenes, evaluándolas según su exactitud, velocidad de inferencia, requerimientos computacionales y facilidad de implementación en dispositivos embebidos.

*Tabla 5. Comparativa de modelos de clasificación*

<b>Modelo CNN</b>	<b>Precisión (mAP)</b>	<b>Velocidad de inferencia</b>	<b>Requerimientos computacionales</b>	<b>Adecuado para tiempo real</b>	<b>Observaciones</b>
Inception V3	Alta (~80%)	Media	Alto	Parcialmente	Buena precisión, pero elevada demanda de recursos
EfficientNet-B0	Alta (~83%)	Media	Medio	Parcialmente	Buen balance entre rendimiento y eficiencia
YOLO v11	Muy alta (>85%)	Alta	Moderado	Sí	Excelente rendimiento en detección en tiempo real

**Fuente:** [29][41], [42].

A partir del análisis comparativo presentado en la Tabla 5, se seleccionó el modelo YOLOv11 en su versión l, debido a su alta precisión, capacidad de operación en tiempo real y adecuado acoplamiento con la plataforma NVIDIA Jetson Nano. Esta combinación permite una detección eficiente y precisa en condiciones variables de iluminación y entorno.

### 3.1.6 Fase Dos

Se lleva a cabo la elaboración del conjunto de datos (dataset) necesario para el entrenamiento y la validación del modelo de clasificación. Esta fase resulta fundamental, ya que la calidad y diversidad de las imágenes influyen directamente en el desempeño y la capacidad de generalización del sistema.

El procedimiento inicia con la identificación y recopilación de imágenes correspondientes a las tres especies seleccionadas: *Anastrepha striata*, *Anastrepha serpentina* y *Ceratitis capitata*. Las imágenes fueron obtenidas a partir de bases de datos abiertas, archivos científicos y capturas realizadas en campo, con el propósito de cubrir distintas condiciones de fondo, iluminación y orientación.

Posteriormente, se efectúa el proceso de etiquetado de las imágenes mediante el uso de herramientas especializadas como Roboflow, las cuales permitieron delimitar las áreas de interés asociadas a cada insecto. Este proceso posibilitó la generación de los archivos de anotación requeridos por el modelo YOLOv11 para su entrenamiento supervisado.

Finalmente, se aplican técnicas de aumento de datos (data augmentation) con el fin de incrementar la diversidad y robustez del dataset, considerando que en su etapa inicial este contaba únicamente con 1 498 imágenes.

*Tabla 6. Composición Cuantitativa del Dataset*

Conjunto	<i>Ceratitis capitata</i>	<i>Anastrepha Serpentina</i>	<i>Anastrepha Striata</i>	Total de Instancias	Total de Imágenes	Porcentaje
<b>Entrenamiento (70%)</b>	1074	1035	1035	3144	X	70%
<b>Validación (20%)</b>	96	98	105	299	Y	20%
<b>Prueba (Test) (10%)</b>	46	54	49	149	Z	10%
<b>Total General</b>	1216	1187	1189	3592	(X+Y+Z)	100%

Para generar el conjunto de datos final de 3592 instancias, se aplica un pipeline de Data Augmentation en la plataforma Roboflow para incrementar la diversidad y prevenir el sobreajuste, observar tabla 6.

Las transformaciones incluyeron:

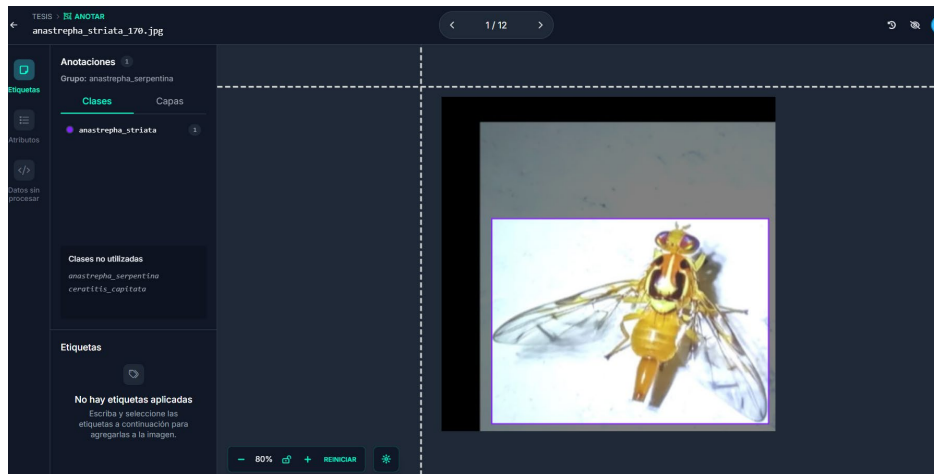
Rotación de 90°: en sentido horario, antihorario, boca abajo.

Rotación: Entre -13° y +13°.

Saturación: Entre -28% y +28%.

Brillo: Entre -24% y +24%.

Ruido: hasta el 0,97% de los píxeles.



*Figura 9. Proceso de etiquetado en RoboFlow.*

En la figura 9, se observa el procedimiento de etiquetado de imágenes mediante bounding box.

### 3.1.7 Fase Tres

Esta fase se enfoca en el diseño e implementación de un sistema de categorización basado en visión artificial, cuyo propósito fue lograr la detección temprana de moscas de la fruta, específicamente de las especies *Anastrepha serpentina*, *Anastrepha striata* y *Ceratitis capitata*. El objetivo principal consistió en integrar componentes de hardware y software que permitieran un procesamiento óptimo en tiempo real, proporcionando una solución portátil, accesible y eficiente para apoyar la vigilancia agrícola y la prevención de plagas.

El diagrama de bloques presentado en la figura 10 ilustra la arquitectura de hardware y software del sistema propuesto. El bloque de alimentación es el encargado de suministrar los 5 V/4 A necesarios para el funcionamiento estable de todos los componentes del sistema, entre ellos la cámara, la pantalla táctil de 7 pulgadas y la plataforma NVIDIA Jetson Nano, garantizando una operación sin fluctuaciones de voltaje.

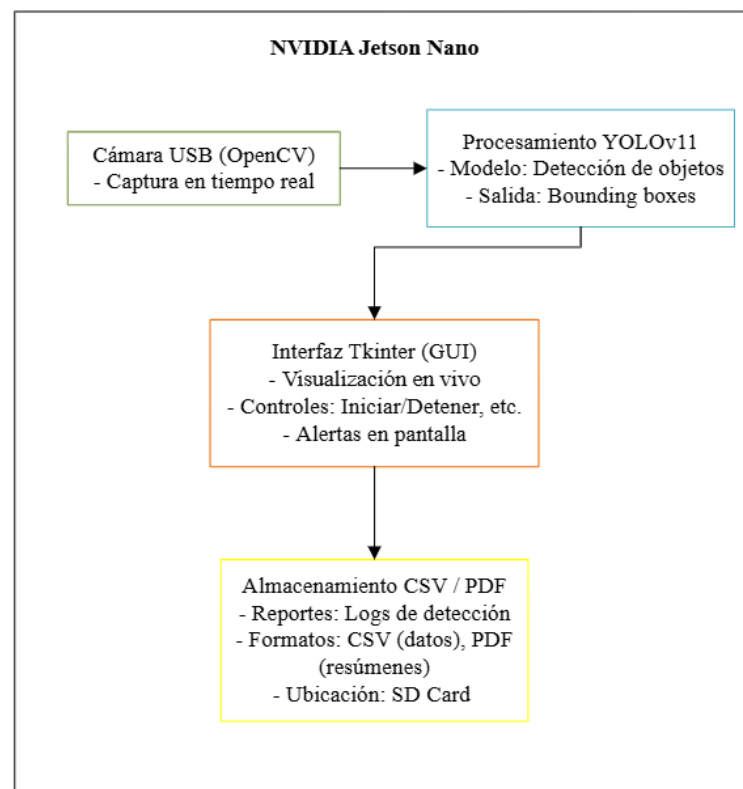
En el bloque de adquisición de datos se encuentra la cámara, cuya función es capturar imágenes del entorno en tiempo real. Estas imágenes son enviadas directamente al módulo de procesamiento, ubicado en la Jetson Nano, que constituye el núcleo central del sistema.

En el bloque de procesamiento, la Jetson Nano ejecuta el modelo YOLOv11, previamente entrenado mediante técnicas de transfer learning, realizando la detección e inferencia en tiempo real para identificar las especies presentes en las imágenes capturadas. Para este proceso se emplean bibliotecas como OpenCV, Ultralytics YOLO, Tkinter y NumPy, las cuales permiten el procesamiento visual, la gestión de datos y la implementación de la interfaz gráfica.

El bloque de visualización presenta los resultados de la clasificación a través de una interfaz gráfica desarrollada en Tkinter y desplegada en una pantalla táctil de 7 pulgadas. Esta interfaz muestra las detecciones en tiempo real, incluyendo la especie identificada, el nivel de confianza y un conteo acumulativo de las muestras detectadas. Adicionalmente, el sistema permite la generación de reportes en formatos CSV y PDF que contienen la información procesada y las estadísticas de detección, facilitando su posterior análisis.

Finalmente, el sistema incorpora un bloque de almacenamiento y gestión de resultados, en el cual se registran las detecciones junto con la especie identificada, la fecha y la hora correspondientes. Desde la interfaz es posible consultar o exportar dichos registros, lo que permite mantener un historial digital de las detecciones y simplificar el seguimiento del monitoreo.

En conjunto, la arquitectura propuesta integra los módulos de captura, procesamiento, visualización y almacenamiento, con el objetivo de ofrecer una herramienta autónoma, portátil y de bajo costo orientada a respaldar las labores de detección temprana de plagas en el ámbito agrícola.



*Figura 10. Diagrama de Bloques del Dispositivo*

- **Ensamblaje e integración del sistema de clasificación basado en jetson nano.**

El presente apartado detalla el proceso de ensamblaje y puesta en funcionamiento del kit de desarrollo NVIDIA Jetson Nano de Makeronics, el cual constituye la plataforma fundamental para la implementación del sistema de detección y clasificación de especies de moscas de la fruta mediante visión artificial.

El conjunto de componentes incluidos en el kit se detalla a continuación:

**A. Estuche de acrílico Jetson Nano**

1 funda protectora acrílica.

1 juego de tornillos y separadores.

1 ventilador de refrigeración PWM.

1 destornillador de montaje.

1 botón de encendido y 1 botón de reinicio.

Manual de instrucciones.

**B. Cámara IMX219-77**

1 módulo de cámara IMX219-77.

1 funda de protección para la cámara.

Manual de montaje.

**C. Tarjeta microSD de 64 GB**

1 tarjeta TF de 64 GB

1 lector de tarjetas USB.

**D. Kit de pantalla táctil LCD de 7 pulgadas**

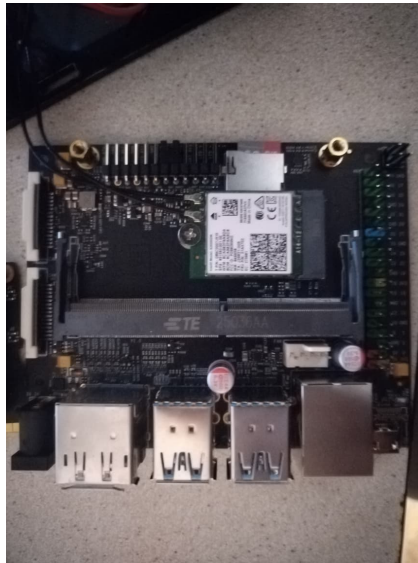
Pantalla táctil capacitiva de 7" (resolución 1024×600 IPS).

Cable HDMI.

Cable micro USB para la función táctil.

Soporte de montaje.

- **Instalación de la Tarjeta de Red de la Nvidia Jetson Nano**



*Figura 11. Tarjeta de Red Externa*

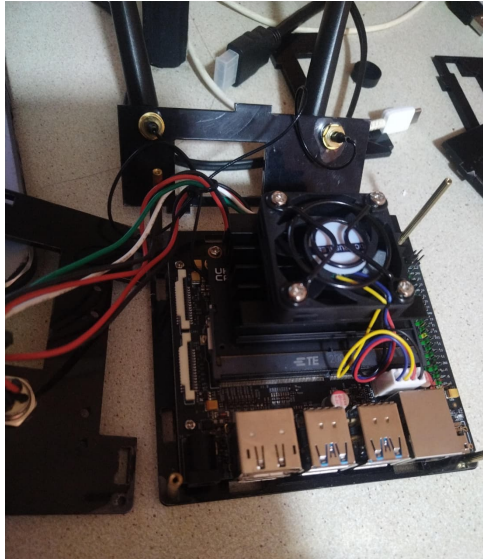
La NVIDIA Jetson Nano no cuenta con conectividad inalámbrica integrada, por lo que se incorporó una tarjeta de red externa, como se observa en la figura 11, con el propósito de habilitar la conexión a Internet requerida para la instalación de dependencias, la actualización del sistema operativo y la transferencia de archivos de entrenamiento.

- **Instalación de la NVIDIA Jetson Nano en la base acrílica**



*Figura 12. Estuche acrílico*

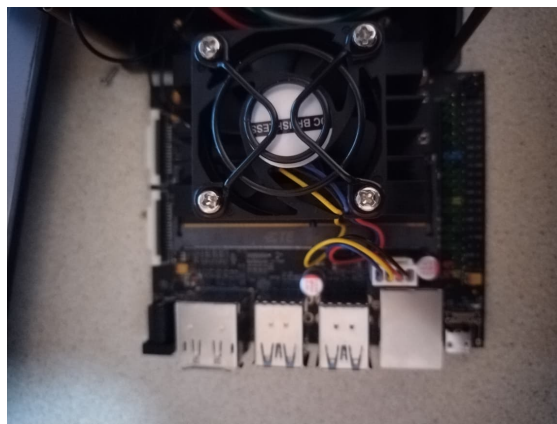




*Figura 13. Jetson Nano sobre la base inferior del estuche acrílico.*

En las Figuras 12 y 13 se observa la colocación de la placa Jetson Nano sobre la base inferior del estuche acrílico, alineando los orificios de montaje con los separadores plásticos. Posteriormente, la placa fue fijada mediante los tornillos incluidos, asegurando que quedara firme y correctamente posicionada,

- **Montaje del sistema de ventilación**

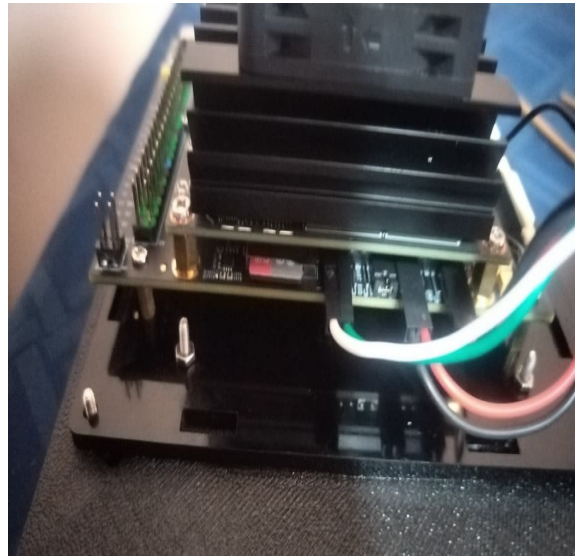


*Figura 14. Sistema de ventilación*

Sobre el disipador de la Jetson Nano se instala el ventilador de refrigeración PWM, tal como se muestra en la Figura 14, orientando el flujo de aire hacia el disipador.

El ventilador se conecta al conector GPIO de 5 V correspondiente, lo que permite la regulación automática de la velocidad y el mantenimiento de temperaturas óptimas de la tarjeta.

- **Instalación de los botones de encendido y reinicio**



*Figura 15. Conexión botones de encendido y reinicio.*

Los botones Power y Reset se colocan en los orificios del estuche acrílico, como se muestra en la Figura 15, y se conectan a los pines GPIO designados de la Jetson Nano.

Estos botones permiten encender y reiniciar el sistema sin necesidad de manipular directamente la placa.

- **Conexión de la cámara IMX219-77**

La cámara IMX219-77 figura 16, se conecta al puerto CSI (Camera Serial Interface) de la Jetson Nano mediante su cable plano, se aseguraron las trabas del conector para evitar falsos contactos.

Finalmente, se coloca la cámara dentro de su carcasa protectora y se fija al soporte frontal del estuche.



*Figura 16 Camara IMX219-77*

- **Instalación de la pantalla táctil LCD de 7 pulgadas**

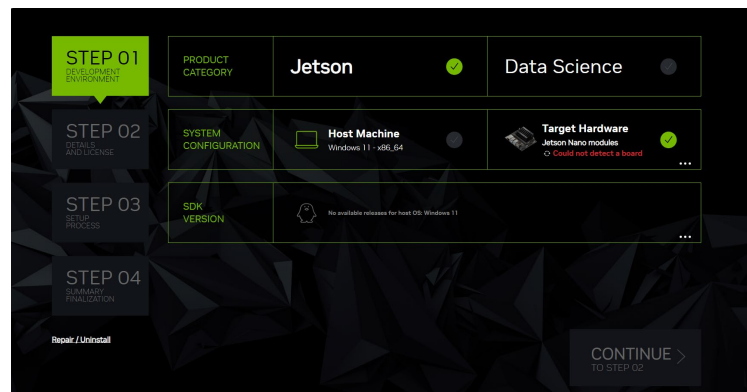
La pantalla táctil se monta en su soporte y se conecta a la Jetson Nano mediante el cable HDMI (para la señal de video) y el cable micro USB (para la función táctil). De esta forma, la pantalla actúa como el principal medio de visualización e interacción con el sistema, véase figura 17.



*Figura 17. Pantalla LCD*

- **Preparación del Sistema Operativo Nvidia Jetson Nano**

En una computadora externa, se graba la imagen del sistema operativo JetPack utilizando la herramienta Nvidia SDK Manager, véase figura 18.



*Figura 18. Nvidia SDK Manager*

- **Alimentación y verificación inicial**

Finalmente, se conecta la fuente de alimentación de 5V/4A al puerto de energía de la Jetson Nano y se presiona el botón de encendido. Tras el arranque inicial, se verifica el funcionamiento del sistema, la operación del ventilador, la detección de la cámara, el funcionamiento de la pantalla táctil y la conectividad de red, véase figura 19.



*Figura 19. Conexión de la alimentación de la Nvidia Jetson Nano.*

- **Selección del modelo base**

En este apartado se explica el procedimiento de entrenamiento del modelo de detección y categorización de especies de moscas de la fruta con la arquitectura YOLOV11.

Para el actual proyecto, se elige la versión YOLO11 (Large) de esta familia debido a que proporciona una capacidad superior para detectar objetos pequeños y un nivel de precisión más elevado, cualidades esenciales para identificar insectos como las moscas de la fruta.

A diferencia de modelos de clasificación como InceptionV3 o EfficientNet, que procesan imágenes completas, YOLO11 realiza simultáneamente la localización y clasificación de objetos, lo que lo hace ideal para realizar el sistema propuesto.

*Tabla 7. Rendimiento del Modelo YOLOv11 en sus diferentes variantes.*

<b>Modelo</b>	<b>Tamaño (Píxeles)</b>	<b>mAPval 50-95</b>	<b>Velocidad CPU ONNX (ms)</b>	<b>Velocidad T4 TensorRT10 (ms)</b>	<b>Parámetros (M)</b>	<b>FLOPS (B)</b>
<b>YOLO11n</b>	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
<b>YOLO11s</b>	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
<b>YOLO11m</b>	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
<b>YOLO11l</b>	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
<b>YOLO11x</b>	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

**Fuente:** [29].

Como se puede observar en la tabla 7, YOLO11l sobresale por su capacidad para brindar una relación equilibrada y robusta entre rendimiento y precisión, con una velocidad que permite la inferencia en tiempo real en la NVIDIA Jetson Nano.

Esto posibilita alcanzar un balance ideal entre el desempeño computacional y la precisión al clasificar especies que se parecen visualmente, como *Anastrepha striata*, *Ceratitis capitata* y *Anastrepha serpentina*.

El modelo se entrena utilizando un método de Transfer Learning, que consiste en la adaptación de un modelo preentrenado con el conjunto de datos COCO a un nuevo dataset. Este último está compuesto por imágenes de las especies más frecuentes de moscas de la fruta, que en este caso fueron *ceratitis capitata*, *anastrepha serpentina* y *anastrepha striata*.

- **Proceso de transfer learning**

Se utiliza el método de Transfer Learning para ajustar el modelo al trabajo concreto de detección de moscas de la fruta. En este método, las capas convolucionales inferiores de YOLOV11, que se encargan de la extracción de rasgos generales como las texturas, los bordes y las formas, fueron preservadas. Por su parte, las capas superiores fueron reentrenadas para adaptarse al nuevo conjunto de datos compuesto por las tres especies objetivo en la tabla 1.

De esta forma, el modelo se adapta para reconocer los patrones morfológicos distintivos de cada especie, optimizando tanto la clasificación como la localización dentro de las imágenes.

- **Conjunto de datos (dataset)**

Cada especie se organizó en carpetas independientes y etiquetada manualmente con la herramienta online RoboFlow, generando archivos en formato YOLO (.txt), donde se especifican las coordenadas de los cuadros delimitadores (bounding boxes) y la clase correspondiente.

La distribución del dataset se puede observar en la tabla 6, el conjunto final incluyó imágenes en formato .jpg y .png, con una resolución promedio de 640×640 píxeles, garantizando uniformidad en la entrada de datos durante el entrenamiento.

- **Configuración del entrenamiento**

El entrenamiento del modelo se realiza utilizando el framework Ultralytics YOLOV11, implementado en Python 3.9 y ejecutado en el entorno Google Colab, aprovechando los recursos de GPU (NVIDIA Tesla T4).

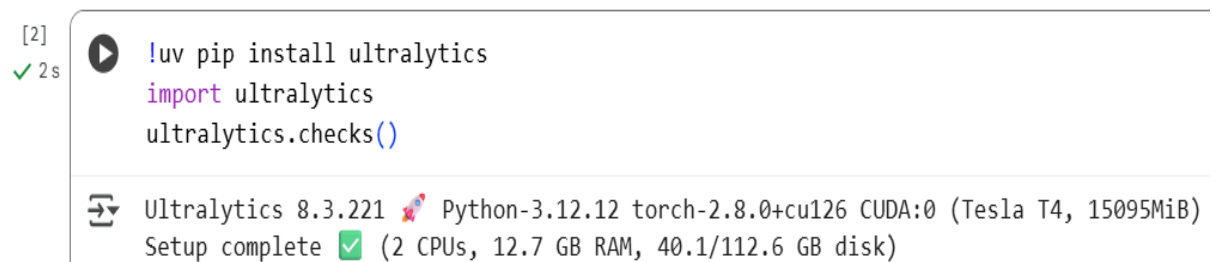
Los principales parámetros de entrenamiento se definieron en la tabla 8:

Tabla 8. Parámetros de Entrenamiento.

Parámetro	Valor
Modelo base	yolo11.pt
Tamaño de imagen	640 × 640 píxeles
Número de épocas	60
Batch size	16
Optimizador	Adam
Tasa de aprendizaje inicial	0.001
Archivo de configuración (data.yaml)	Dataset personalizado con 3 clases

- **Preparación del entorno de Google Colab**

La figura 20 muestra la preparación inicial del entorno en Google Colab para entrenar el modelo YOLOV11. En primer lugar, se instala la librería Ultralytics mediante el comando `!uv pip install ultralytics`, que descarga todas las dependencias necesarias para el uso de los modelos YOLO, incluido PyTorch. Luego, se importa el módulo `ultralytics` para poder acceder a sus funciones desde Python. Finalmente, con `ultralytics.checks()` se realiza una verificación del entorno, comprobando que la instalación se haya completado correctamente y mostrando detalles como la versión de Python, PyTorch, el modelo de GPU disponible (en este caso una Tesla T4 de 15 GB), la memoria RAM y el espacio en disco. Esta comprobación confirma que el entorno está correctamente configurado y listo para iniciar el proceso de entrenamiento del modelo.



```
[2] ✓ 2s !uv pip install ultralytics
import ultralytics
ultralytics.checks()

Ultralytics 8.3.221 Python-3.12.12 torch-2.8.0+cu126 CUDA:0 (Tesla T4, 15095MiB)
Setup complete (2 CPUs, 12.7 GB RAM, 40.1/112.6 GB disk)
```

Figura 20. Instalación de dependencias.

- **Importar el dataset desde roboflow**

Para importar automáticamente el conjunto de datos (dataset) desde la plataforma Roboflow hacia el entorno de trabajo en Google Colab, utilizamos el código de la figura 21.

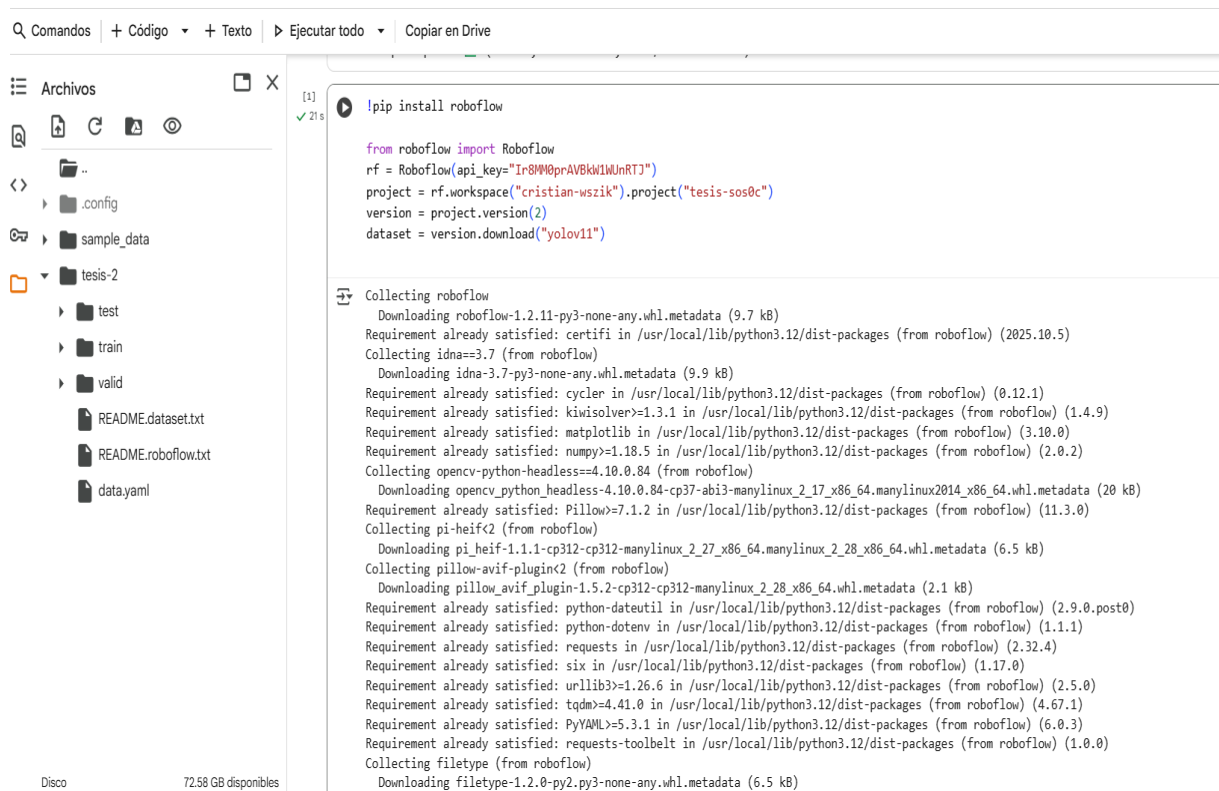
Primero, el comando `!pip install roboflow` instala la librería necesaria para conectarse a Roboflow mediante su API. Luego, con `from roboflow import Roboflow` se importa la clase principal que permite interactuar con los proyectos almacenados en la cuenta del usuario.

La línea `rf = Roboflow(api_key="Ir8MM0prAVBkW1WUnRTJ")` autentica al usuario mediante su clave API personal, lo que habilita el acceso a los datasets privados o públicos de su cuenta. Posteriormente, se especifica el proyecto (`project = rf.workspace("cristian-wszik").project("tesis-sos0c")`) y la versión del conjunto de datos (`version =`



project.version(2)), ya que Roboflow permite mantener diferentes versiones del mismo dataset conforme se realizan mejoras o ajustes.

Finalmente, `dataset = version.download("yolov11")` descarga el dataset en el formato compatible con YOLOv11, organizando automáticamente las carpetas de imágenes de entrenamiento, validación y prueba, junto con el archivo `data.yaml` necesario para el entrenamiento del modelo. Este proceso simplifica considerablemente la preparación de datos, asegurando una estructura estandarizada y lista para su uso inmediato.



The screenshot shows a Google Colab environment. On the left, the file explorer shows a directory structure with folders like `test`, `train`, and `valid`, and files like `README.dataset.txt`, `README.roboflow.txt`, and `data.yaml`. The main code editor shows the following Python code:

```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="Ir8MM0prAVBk1WUnRTJ")
project = rf.workspace("cristian-wszik").project("tesis-sos0c")
version = project.version(2)
dataset = version.download("yolov11")
```

The output of the code execution shows the following progress:

```
Collecting roboflow
  Downloading roboflow-1.2.11-py3-none-any.whl.metadata (9.7 kB)
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from roboflow) (2025.10.5)
Collecting idna==3.7 (from roboflow)
  Downloading idna-3.7-py3-none-any.whl.metadata (9.9 kB)
Requirement already satisfied: cyclor in /usr/local/lib/python3.12/dist-packages (from roboflow) (0.12.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from roboflow) (1.4.9)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (from roboflow) (3.10.0)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.12/dist-packages (from roboflow) (2.0.2)
Collecting opencv-python-headless==4.10.0.84 (from roboflow)
  Downloading opencv_python_headless-4.10.0.84-cp37-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (20 kB)
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.12/dist-packages (from roboflow) (11.3.0)
Collecting pi-heif<2 (from roboflow)
  Downloading pi_heif-1.1.1-cp312-cp312-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl.metadata (6.5 kB)
Collecting pillow-avif-plugin<2 (from roboflow)
  Downloading pillow_avif_plugin-1.5.2-cp312-cp312-manylinux_2_28_x86_64.whl.metadata (2.1 kB)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.12/dist-packages (from roboflow) (2.9.0.post0)
Requirement already satisfied: python-dotenv in /usr/local/lib/python3.12/dist-packages (from roboflow) (1.1.1)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from roboflow) (2.32.4)
Requirement already satisfied: six in /usr/local/lib/python3.12/dist-packages (from roboflow) (1.17.0)
Requirement already satisfied: urllib3>=1.26.6 in /usr/local/lib/python3.12/dist-packages (from roboflow) (2.5.0)
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.12/dist-packages (from roboflow) (4.67.1)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.12/dist-packages (from roboflow) (6.0.3)
Requirement already satisfied: requests-toolbelt in /usr/local/lib/python3.12/dist-packages (from roboflow) (1.0.0)
Collecting filetype (from roboflow)
  Downloading filetype-1.2.0-py2.py3-none-any.whl.metadata (6.5 kB)
```

Figura 21. importación del dataset a Google Colab.

- **Inicialización del entrenamiento**

```

Archivo  Editar  Ver  Insertar  Entorno de ejecución  Herramientas  Ayuda
Q Comandos  + Código  + Texto  ▶ Ejecutar todo  Copiar en Drive

Archivos
- config
- runs
- tesis-2
  - test
  - train
  - valid
  - README.dataset.txt
  - README.robotflow.txt
  - data.yaml
  - yolo11.pt
  - yolo11n.pt

[yolo train model=yolo11.pt data="/content/tesis-2/data.yaml" epochs=60 imgsz=640]
Downloading https://github.com/ultralytics/assets/releases/download/v8.3.0/yolo11.pt to 'yolo11.pt': 100% 49.0MB 40.1MB/s 1.2s
Ultralytics 8.3.221 Python-3.12.12 torch-2.8.0-cu126 CUDA:0 (Tesla T4, 15095MiB)
engine/trainer: agnostic_rms=False, amp=True, augment=False, auto_augment=andaugment, batch=16, bgr=0.0, box=7.5, cache=False, cfg=None, classes=0
Downloading https://github.com/ultralytics/assets/releases/download/v8.3.0/yolo11n.pt to '/root/.config/Ultralytics/yolo11n.pt': 100% 735.1KB 25.1MB/s 0.0s
Overriding model.yaml nc=80 with nc=3

   from  n  params module                                arguments
   --    --  --    --
0       1    1856 ultralytics.nn.modules.conv.Conv      [3, 64, 3, 2]
1       1    73984 ultralytics.nn.modules.conv.Conv      [64, 128, 3, 2]
2       1    173824 ultralytics.nn.modules.block.C3k2      [128, 256, 2, True, 0.25]
3       1    590336 ultralytics.nn.modules.conv.Conv      [256, 256, 3, 2]
4       1    693712 ultralytics.nn.modules.block.C3k2      [256, 512, 2, True, 0.25]
5       1    2368320 ultralytics.nn.modules.conv.Conv      [512, 512, 3, 2]
6       1    2234368 ultralytics.nn.modules.block.C3k2      [512, 512, 2, True]
7       1    2368320 ultralytics.nn.modules.conv.Conv      [512, 512, 3, 2]
8       1    2234368 ultralytics.nn.modules.block.C3k2      [512, 512, 2, True]
9       1    656896 ultralytics.nn.modules.block.SPPF      [512, 512, 5]
10      1    1455616 ultralytics.nn.modules.block.C2PSA      [512, 512, 2]
11      1    0 torch.nn.modules.upsampling.Upsample      [None, 2, 'nearest']
12     [-1, 6] 1    0 ultralytics.nn.modules.conv.Concat      [1]
13      1    2 2496512 ultralytics.nn.modules.block.C3k2      [1024, 512, 2, True]
14      1    0 torch.nn.modules.upsampling.Upsample      [None, 2, 'nearest']
15     [-1, 4] 1    0 ultralytics.nn.modules.conv.Concat      [1]
16      1    2 756736 ultralytics.nn.modules.block.C3k2      [1024, 256, 2, True]
17      1    590336 ultralytics.nn.modules.conv.Conv      [256, 256, 3, 2]
18     [-1, 13] 1    0 ultralytics.nn.modules.conv.Concat      [1]
19      1    2 2365440 ultralytics.nn.modules.block.C3k2      [768, 512, 2, True]
20      1    2368320 ultralytics.nn.modules.conv.Conv      [512, 512, 3, 2]
21     [-1, 10] 1    0 ultralytics.nn.modules.conv.Concat      [1]
22      1    2 2496512 ultralytics.nn.modules.block.C3k2      [1024, 512, 2, True]
23     [16, 19, 22] 1 1413337 ultralytics.nn.modules.head.Detect [3, [256, 512, 512]]

YOLO11n summary: 357 layers, 25,312,793 parameters, 25,312,777 gradients, 87.3 GFLOPs

Transferred 1009/1015 items from pretrained weights
Freezing layer 'model.23.dfl.conv.weight'
AMP: running Automatic Mixed Precision (AMP) checks...
Downloading https://github.com/ultralytics/assets/releases/download/v8.3.0/yolo11n.pt to 'yolo11n.pt': 100% 5.4MB 102.0MB/s 0.1s
AMP: checks passed
train: Fast image access (ping: 0.040.0 ms, read: 2080.9±640.6 MB/s, size: 71.5 KB)
train: Scanning /content/tesis-2/train/labels... 3144 images, 0 backgrounds, 0 corrupt: 100% 3144/3144 2.6kIt/s 1.2s
train: New cache created: /content/tesis-2/train/labels.cache
augmentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01, method='weighted_average', num_output_channels=1)
val: Fast image access (ping: 0.040.0 ms, read: 699.7±393.8 MB/s, size: 38.6 KB)
val: Scanning /content/tesis-2/valid/labels... 299 images, 0 backgrounds, 0 corrupt: 100% 299/299 2.1kIt/s 0.1s
val: New cache created: /content/tesis-2/valid/labels.cache
Plotting labels to /content/runs/detect/train/labels.jpg...
optimizer: 'optimizer-auto' found, ignoring 'lr=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr0' and 'momentum' automatically...
optimizer: Adam(lr=0.001429, momentum=0.9) with parameter groups 167 weight(decay=0.0), 174 weight(decay=0.0005), 173 bias(decay=0.0)
Image sizes 640 train, 640 val
Using 2 dataloader workers
Logging results to /content/runs/detect/train
Starting training for 60 epochs...
  
```

Figura 22. Proceso de entrenamiento.

En la figura 22 se observa el proceso de inicio del entrenamiento del modelo YOLOv11 en el entorno de Google Colab, utilizando la librería Ultralytics. En esta fase, el sistema carga el modelo base preentrenado (yolo11n.pt) y verifica la configuración del entorno, mostrando información sobre la versión de Python, CUDA (aceleración por GPU) y la arquitectura del modelo. También se observa cómo se descargan los archivos necesarios desde los repositorios oficiales de Ultralytics y cómo se estructura la red neuronal, detallando las capas convolucionales y parámetros utilizados.

Posteriormente, el modelo comienza a preparar el conjunto de datos definido en el archivo data.yaml, escaneando las carpetas train, val y test para identificar y organizar las imágenes y etiquetas que se usarán en el entrenamiento. Finalmente, se indica que el proceso se realizará durante 60 épocas, utilizando el optimizador Adam y aprovechando la GPU Tesla T4.



- **Descarga del modelo Final**

Para localizar y acceder al modelo final (best.pt), diríjase al explorador de archivos en el panel izquierdo de Google Colab. Siga la estructura de directorios generada durante el entrenamiento:

1. Ingrese a la carpeta runs.
2. Ingrese a la carpeta detect
3. Acceda a la subcarpeta train.
4. Abra la carpeta weights.

Aquí se encuentra el archivo de pesos entrenados, nombrado habitualmente como best.pt, además de otros archivos relevantes para la validación del modelo, como matriz de confusión, etc. Véase figura 23.

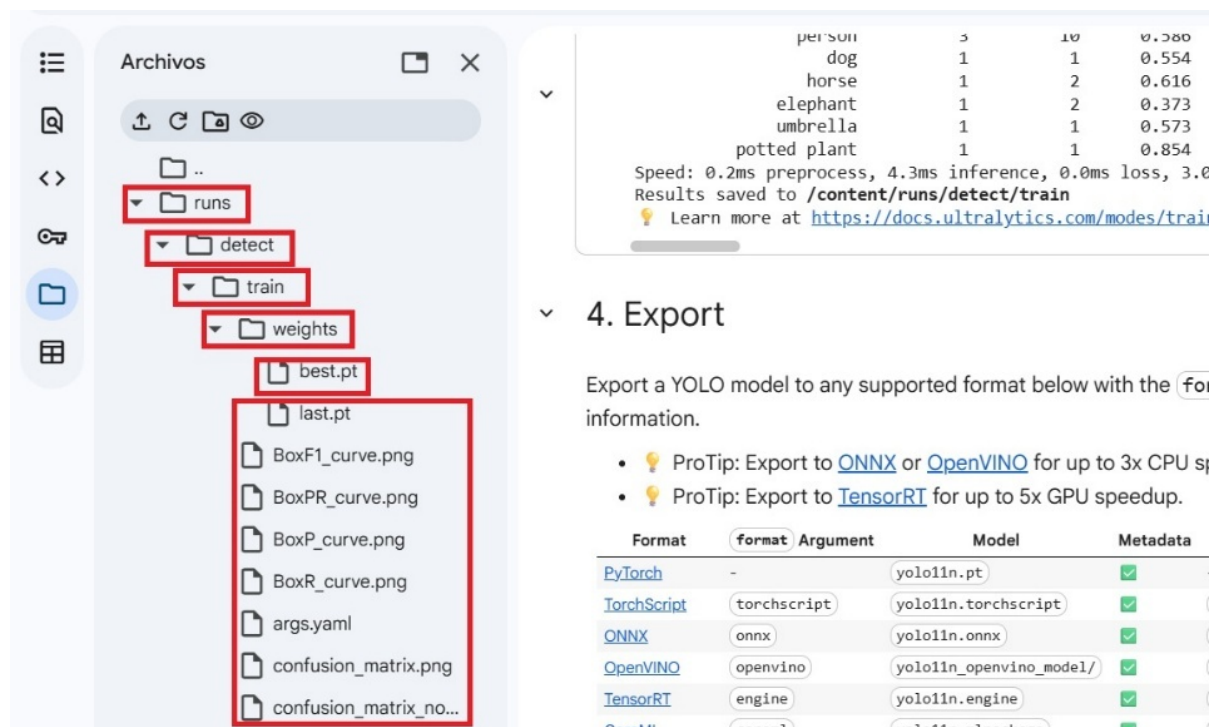
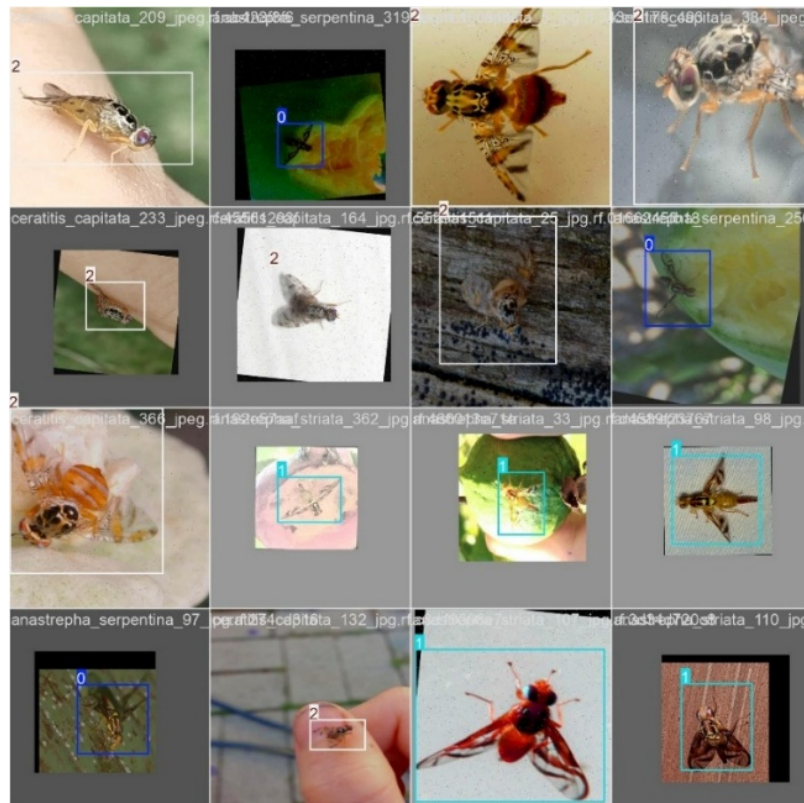


Figura 23. Como descargar el modelo final desde Google Colab.

- **Muestras del Conjunto de Datos de Entrenamiento y Validación para la Clasificación de Moscas de la Fruta**



*Figura 24. Conjunto de Entrenamiento.*



*Figura 25. Conjunto de Validación.*

En la figura 24 y 25 se muestran ejemplos utilizados en el proceso de entrenamiento y validación del modelo de detección, cada recuadro indica el contenedor delimitador (bounding box) de los individuos presentes y su respectiva clase. Las especies representadas corresponden a *Anastrepha serpentina*, *Anastrepha striata*, y *Ceratitis capitata*. Estas muestras fueron fundamentales para el aprendizaje supervisado del modelo, permitiendo la identificación de variaciones en iluminación, fondo, posición y orientación de los insectos.

- **Pantalla inicial**

La pantalla inicial del dispositivo se visualiza en la Figura 26, donde el usuario puede ingresar los datos básicos del técnico y del lugar de trabajo, como el nombre, la ubicación y la provincia. También permite seleccionar el modelo que usará el sistema para la detección. Una vez completados estos campos, se habilita el botón “Continuar” para avanzar.



*Figura 26. Pantalla inicial*

- **Interfaz de captura de datos y generación de reportes**

Se presenta la vista previa de la cámara activa junto con el rendimiento en tiempo real (FPS y tiempo de inferencia), véase figura 27. En esta interfaz el usuario puede capturar imágenes, subir archivos, cambiar la cámara, ajustar el umbral de detección y observar el conteo de especies detectadas en una tabla lateral. Además, dispone de opciones para cambiar el modelo, reiniciar el conteo y generar reportes en formato PDF o CSV, lo que facilita el manejo y registro de los resultados obtenidos.

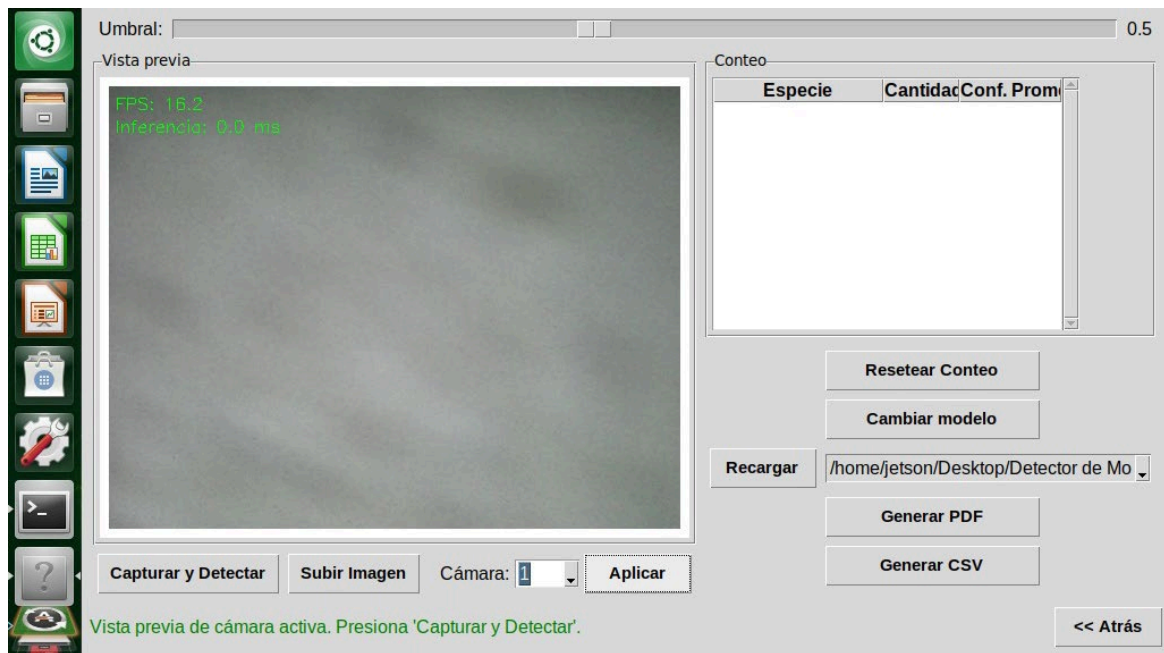


Figura 27. Interfaz de captura de datos y generación de reportes.

- **Interfaz de visualización de resultados y re etiquetado de imágenes**

La pantalla de gestión de etiquetas y visualización de resultados, se observa en la figura 28, donde el sistema muestra las detecciones realizadas sobre la imagen, identificando ejemplares de moscas de la fruta mediante cuadros delimitadores y sus respectivas clases. En esta interfaz el usuario puede seleccionar la clase actual, añadir nuevas clases o eliminarlas, así como visualizar y gestionar la lista de etiquetas generadas, las cuales incluyen la especie detectada y las coordenadas del recuadro correspondiente. Además, cuenta con opciones para eliminar etiquetas individuales o todas a la vez y un botón de confirmación para guardar los cambios realizados en la clasificación.

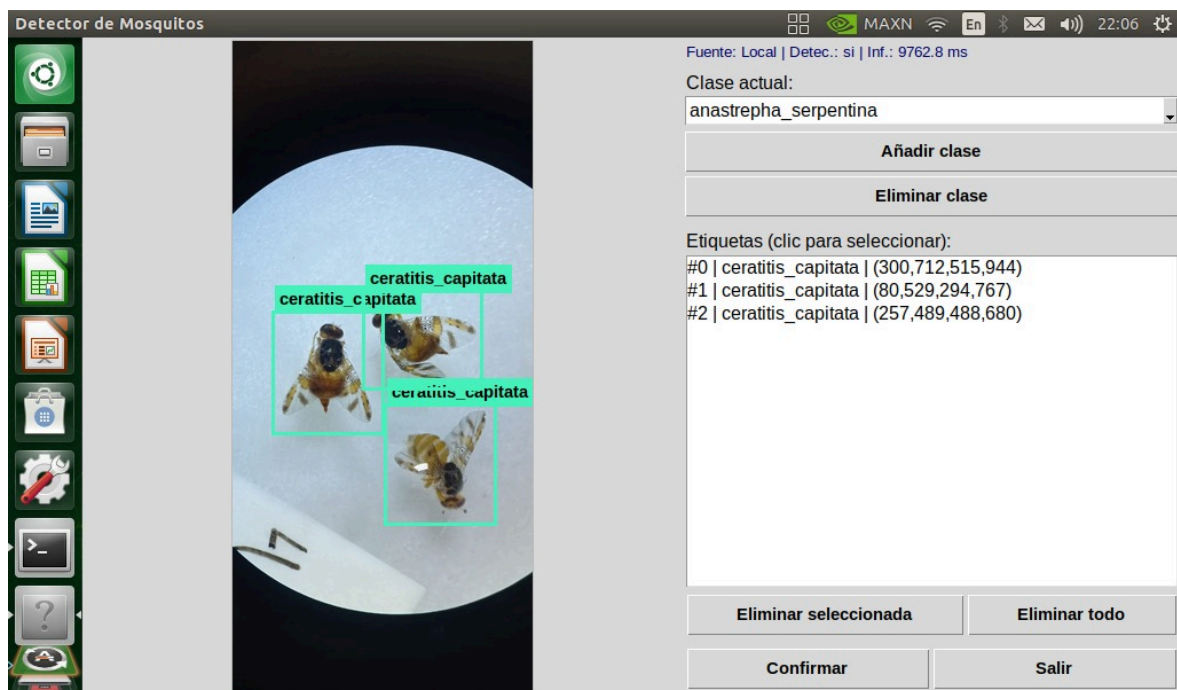


Figura 28. Interfaz de visualización de resultados y re etiquetado de imágenes



#### Fase Cuatro

Esta fase se enfoca en comprobar el desempeño del sistema propuesto, contrastando sus resultados con métodos convencionales de identificación entomológica y verificando su fiabilidad en entornos reales.

- **Pruebas experimentales**

El prototipo de detección automática de moscas de la fruta se somete a una serie de pruebas experimentales con el fin de evaluar su rendimiento. Dichas pruebas se realizan en el Laboratorio de Entomología de AGROCALIDAD, ubicado en Tumbaco, y tienen como objetivo principal comparar el tiempo de respuesta y la capacidad de identificación de las tres especies de interés (*Ceratitis capitata*, *Anastrepha striata* y *Anastrepha serpentina*) del sistema propuesto frente al desempeño humano, bajo condiciones controladas.

- **Diseño experimental**

Las pruebas se estructuran en cuatro rondas independientes, como se observa en la Figura 29. Para cada ronda, se selecciona un conjunto aleatorio de moscas, cuya especie es previamente verificada por expertos con el fin de garantizar la validez de la prueba.



*Figura 29. Pruebas.*

- **Procedimiento de evaluación**

A cada participante se le solicita identificar la especie de los ejemplares presentes en cada una de las cuatro pruebas, basándose únicamente en la observación directa. Durante este proceso se registran de manera individual dos métricas clave, como se observa en la Figura 30:

1. Tiempo total de identificación por prueba.
2. Exactitud en la clasificación de las especies.



*Figura 30. Prueba Manual.*

En paralelo, el prototipo realiza la identificación de los mismos conjuntos de moscas, como se muestra en la Figura 30, manteniendo las mismas condiciones de iluminación y distancia. Los resultados del dispositivo se presentan de forma automática.



*Figura 31. Prueba Automática.*

- **Variables de Comparación**

Para establecer una comparación directa entre el desempeño humano y el del prototipo, se analizan las siguientes variables:

1. Tiempo de Identificación (segundos): Comparativa entre el tiempo requerido por el participante y el tiempo de procesamiento del prototipo.
2. Eficiencia (Valida/ No Valida): Precisión en la identificación correcta de cada especie.
3. Nivel de Conocimiento: Análisis de la diferencia de desempeño entre los participantes de nivel alto, medio y bajo.

- **Consideraciones de validez**

Todo el proceso experimental es supervisado por un especialista, quien verifica rigurosamente:

1. La aleatorización de los ejemplares de moscas en cada prueba.
2. Las condiciones homogéneas de presentación para ambos métodos (manual y automático).
3. La integridad y trazabilidad de los datos y tiempos registrados.

Estas medidas aseguran que los resultados obtenidos reflejan objetivamente la capacidad comparativa entre la metodología manual tradicional y la solución automatizada propuesta.

- **Comparación con técnicas tradicionales**

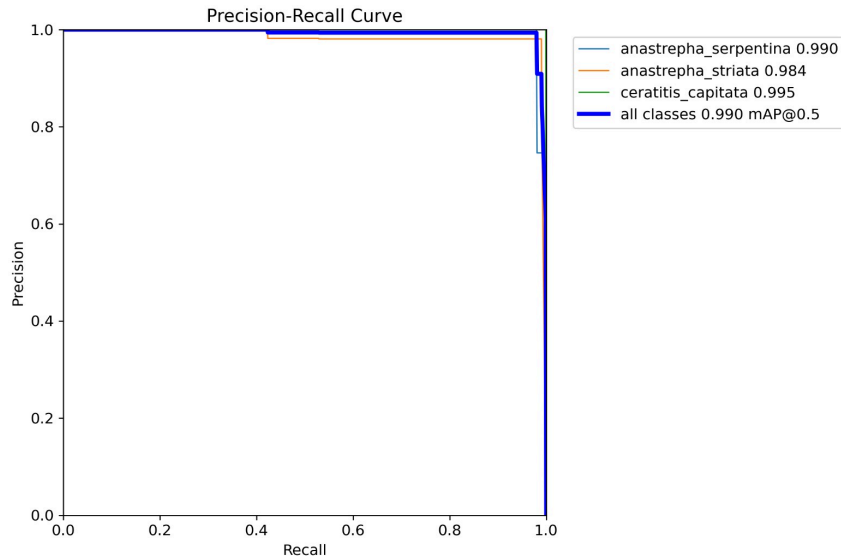
La Tabla 9 presenta una comparación entre el sistema desarrollado y los métodos entomológicos tradicionales utilizados en Agrocalidad para la identificación de moscas de la fruta.

*Tabla 9 Comparación entre el método tradicional y el sistema basado en visión artificial YOLOv11.*

<b>Criterio</b>	<b>Método tradicional (identificación visual)</b>	<b>Sistema propuesto (visión artificial YOLOv11)</b>
<b>Tiempo de análisis</b>	53.8438 segundos	7.3450 segundos
<b>Requiere personal especializado</b>	Sí	No (detección automatizada)
<b>Exactitud promedio</b>	80–85% (dependiente del experto)	95–99% (constante y replicable)
<b>Costo operativo</b>	Medio–Alto (labor y laboratorio)	Bajo (solo consumo energético)
<b>Portabilidad</b>	Limitada (equipos de laboratorio)	Alta (Jetson Nano compacta)
<b>Escalabilidad</b>	Reducida (procesamiento manual)	Alta (procesamiento masivo de imágenes)

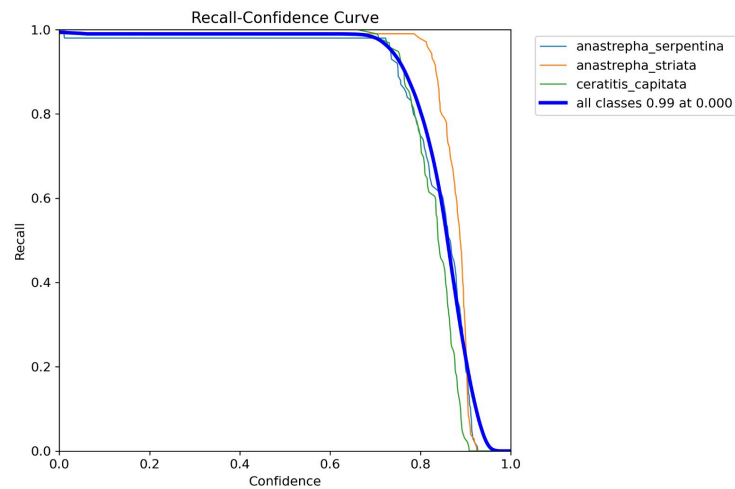
- **Evaluación del rendimiento del modelo yolov11 entrenado**

En esta sección se analizan las métricas cuantitativas y los resultados visuales obtenidos tras el entrenamiento del modelo.



*Figura 32. Curva Precisión–Recall.*

La figura 32 ilustra la relación entre el porcentaje de detecciones precisas (precisión) y el porcentaje de casos reales detectados (recall) para cada especie y su promedio. Las curvas que se acercan al vértice superior derecho muestran un alto acierto con escasas omisiones, lo cual se resume en un  $mAP@0.5 \approx 0.99$ , demostrando así una detección confiable.



*Figura 33. Recall vs. umbral de confianza.*

El Recall se mantiene alto (cerca de 0.98-1.00) véase figura 33, hasta alrededor de 0.80-0.85, y después se reduce rápidamente al requerir un mayor nivel de certeza para aceptar una detección (umbral más riguroso), lo cual hace que aumenten los falsos negativos o las omisiones; esta relación orienta la elección del punto operativo: los umbrales moderados (aproximadamente entre 0.60 y 0.70) dan prioridad a la cobertura, mientras que los umbrales altos disminuyen las advertencias inciertas a costa de perder casos reales.



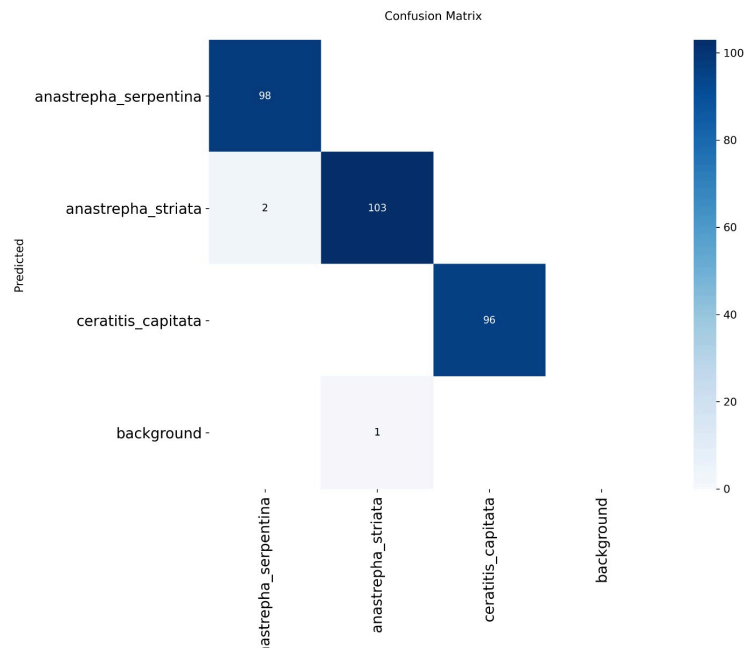


Figura 34. Matriz de confusión.

La matriz de confusión, que se muestra en la figura 34, contrasta las clases reales (filas) y las predichas (columnas); los aciertos se agrupan en la diagonal, mientras que los errores (falsos negativos para la clase real y falsos positivos para la predicha) están fuera de ella. El hecho de que haya un predominio de la diagonal pone de manifiesto una clasificación sólida con confusiones residuales sobre todo entre A. serpentina y A. striata, por su similitud.

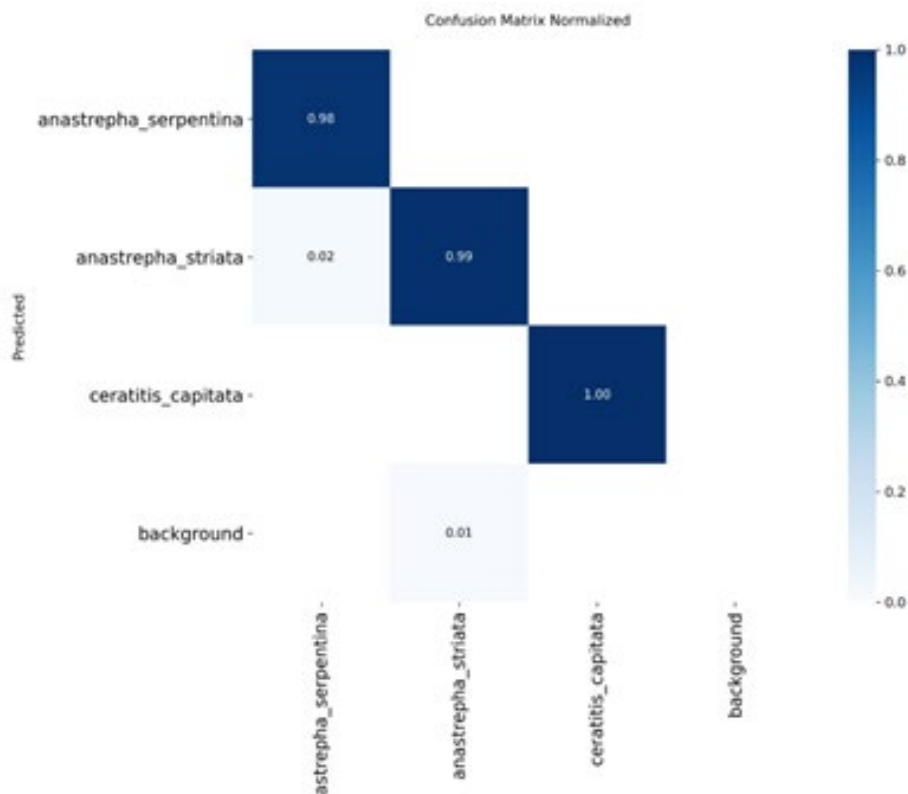
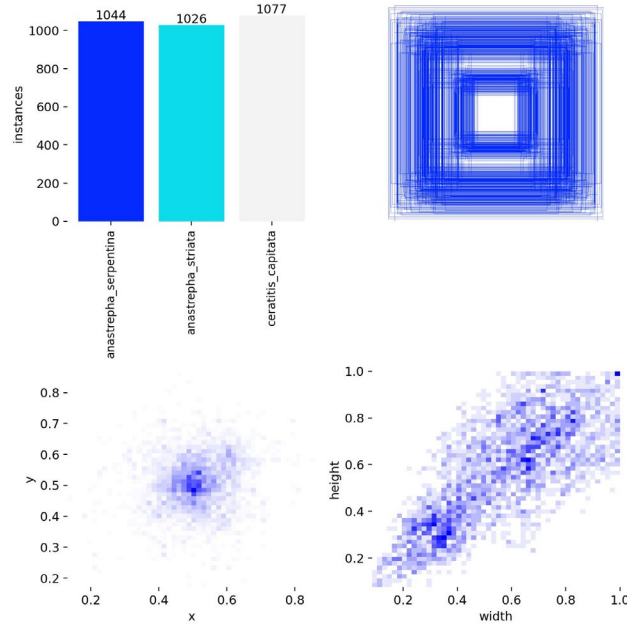


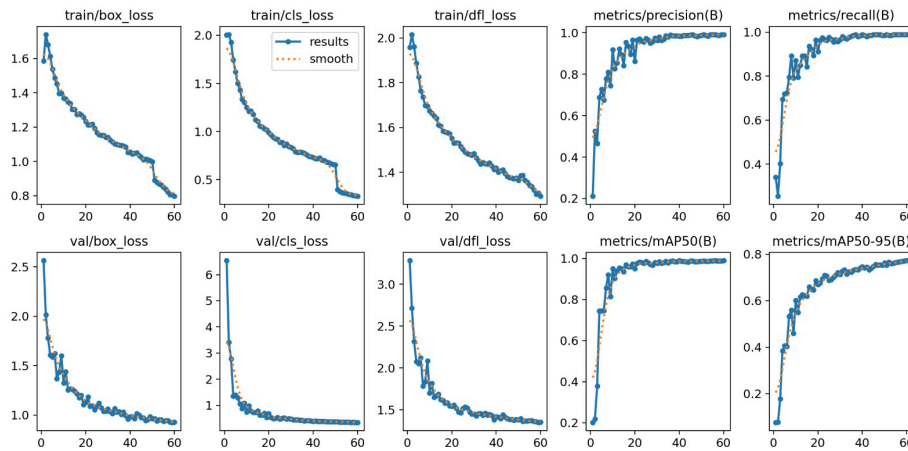
Figura 35. Matriz de confusión normalizada.

La matriz de confusión normalizada se observa en la figura 35. Es parecida a la de la figura 34, pero con filas al 100 %, lo que posibilita la comparación del rendimiento por especie sin tener en cuenta el tamaño de muestra. Las diagonales cercanas a 1.0 corroboran la consistencia de la clasificación por especie y los valores alejados de la diagonal que están próximos a 0.0 señalan errores poco comunes, proporcionando una visión proporcional que hace más sencillo el rastreo de degradaciones particulares durante el despliegue.



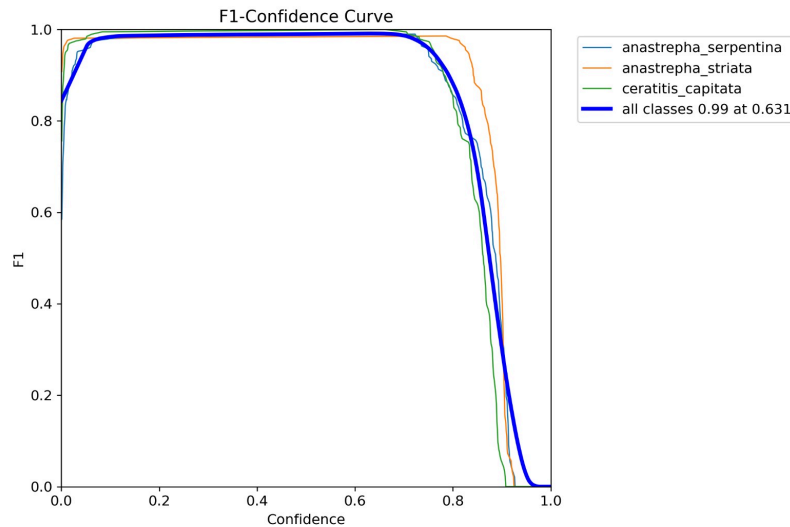
*Figura 36. Conjunto de datos y distribución de cajas.*

Se muestra un balance de instancias por clase, lo que evita sesgos, una amplia cobertura espacial de cajas delimitadoras y variaciones de tamaño, lo que evidencia diversidad en las posiciones y escalas, véase figura 36. Esta variabilidad favorece la generalización del modelo bajo circunstancias reales y apoya su habilidad para diferenciar especies morfológicamente similares con un error mínimo.



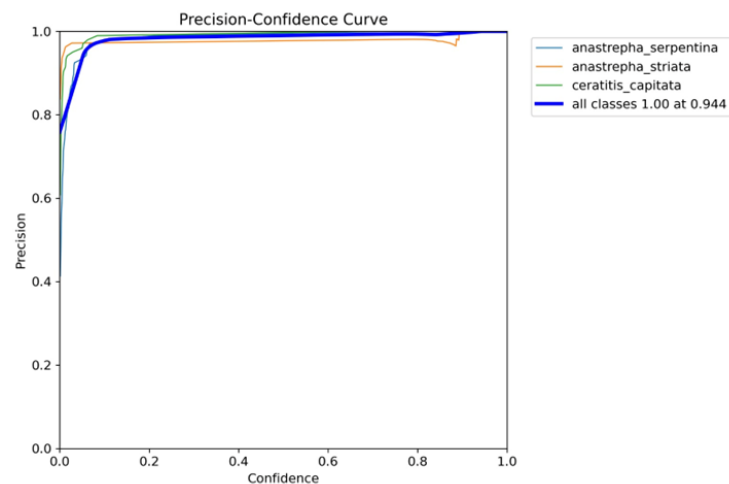
*Figura 37. Curvas de entrenamiento y validación.*

Las pérdidas de localización (box), clasificación (cls) y dfl como se observan en la figura 37 disminuyen de manera constante y que las métricas convergen a  $mAP@0.5 \approx 0.99$  y  $mAP@0.5-0.95 \approx 0.77-0.78$ , esta última más rigurosa al calcular el promedio de IoU entre 0,50 y 0,95, sin diferencias significativas entre entrenamiento y validación; este patrón indica un aprendizaje estable, la falta de sobreajuste importante y el modelo listo para ser puesto en funcionamiento.



*Figura 38. F1 vs. umbral de confianza.*

En la figura 38 el F1 alcanza su máximo umbral de confianza  $\approx 0.63$ , punto operativo recomendado cuando se busca equilibrio entre falsos positivos y falsos negativos; en presencia de especies parecidas (A. striata/A. serpentina), este ajuste ofrece un balance adecuado entre no omitir casos y evitar confusiones.



*Figura 39. Precisión vs. umbral de confianza.*

Frente a una falsa alarma, este sistema es útil; si se adopta, la reducción de cobertura puede ser compensada con una frecuencia de muestreo más alta o con reglas de reverificación. En la figura 39, la precisión se aproxima a 1.0 hacia confianza  $\approx 0.94$ , casi sin falsos positivos, aunque eso implique un menor Recall (más omisiones).

## CAPITULO IV. RESULTADOS Y DISCUSIÓN

### 4.1 RESULTADOS

En este capítulo se exponen y analizan los resultados generados por el equipo de monitoreo durante la fase de pruebas, centrándose inicialmente en la descripción de los datos de tiempo y en la comparación de las mediciones obtenidas mediante el método manual frente al automático, para luego presentar y analizar la información estadística obtenida del estudio, enfocándose en la variable Tiempo en función de las variables de agrupación Método y Nivel de Conocimiento.

#### 4.1.1 Prueba de normalidad

Para proceder al análisis de resultados, se lleva a cabo una prueba de normalidad para determinar si los datos de la variable Tiempo en cada categoría del Método se ajustan a una distribución normal.

- **Hipótesis Nula ( $H_0$ ):** Los datos medidos de Tiempo tienen una distribución normal.
- **Hipótesis Alternativa ( $H_1$ ):** Los datos medidos de Tiempo no tienen una distribución normal.

*Tabla 10. Prueba de Normalidad Shapiro-Wilk N=36.*

Método	Estadístico	gl	(p-valor)
Manual	0,854	32	<0,001
Automático	0,880	4	0,338

Para el Método manual, el p-valor es <0,001, menor que 0,05. Esto indica que los datos para el método manual no se distribuyen normalmente.

Para el Método automático, el p-valor es mayor que 0,05. Esto indica que, para esta categoría, la hipótesis nula se acepta, y los datos sí tienen una distribución normal.

Dado que una de las categorías no sigue una distribución normal, se utiliza una prueba no paramétrica para la comparación de medianas.

#### 4.1.2 Prueba U de Mann-Whitney para muestras independientes.

Se aplica la prueba U de Mann-Whitney para muestras independientes para comparar las distribuciones de Tiempo entre las categorías del Método.

- **Hipótesis Nula ( $H_0$ ):** La distribución de Tiempo es la misma entre categorías de Método.

- **Hipótesis Alternativa ( $H_i$ ):** La distribución de Tiempo es diferente entre categorías de Método.

*Tabla 11. Prueba U de Mann-Whitney para muestras independientes  $N=36$ .*

<b>Prueba</b>	<b>Hipótesis Nula</b>	<b>(p-valor)</b>	<b>Decisión</b>
Prueba U de Mann-Whitney	La distribución de Tiempo es la misma entre categorías de Método.	<0,001	Rechaza la hipótesis nula

El p-valor obtenido es <0,001, que es menor que el nivel de significación de 0,050. Por lo tanto, se rechaza la hipótesis nula, concluyendo que sí existe una diferencia significativa en la distribución del tiempo entre el método manual y el método automático.

Los resultados descriptivos de la prueba confirman esta diferencia:

- El método manual tiene un Rango promedio de 20,50 y una Media de 53,8438.
- El método automático tiene un Rango promedio de 2,50 y una Media de 7,3450.

#### **4.1.3 Resultados y Análisis Comparativo del Tiempo según el Nivel de Conocimiento.**

Se realiza una comparación del Tiempo en función de las categorías de la variable Nivel de Conocimiento (Alto, Medio, Bajo).

- **Prueba de Kruskal-Wallis para muestras independientes.**

Se utiliza la prueba de Kruskal-Wallis para determinar si existe una diferencia significativa en la distribución de la variable Tiempo entre los tres niveles de conocimiento, ya que la prueba de normalidad para el método manual (que incluye a la mayoría de los casos) indicó una distribución no normal.

- **Hipótesis Nula ( $H_0$ ):** La distribución de Tiempo es la misma entre categorías de Nivel\_Conocimiento.
- **Hipótesis Alternativa ( $H_i$ ):** La distribución de Tiempo es diferente entre categorías de Nivel\_Conocimiento.

El p-valor obtenido es <0,001, que es menor que el nivel de significación de 0,050. Esto conduce a rechazar la hipótesis nula y concluir que sí existe una diferencia significativa en la distribución del tiempo en función del nivel de conocimiento.

Tabla 12. Prueba de Kruskal-Wallis para muestras independientes N=32.

Prueba	Hipótesis Nula	(p-valor)	Decisión
Prueba de Kruskal-Wallis	La distribución de Tiempo es la misma entre categorías de Nivel_Conocimiento.	<0,001	Rechaza la hipótesis nula

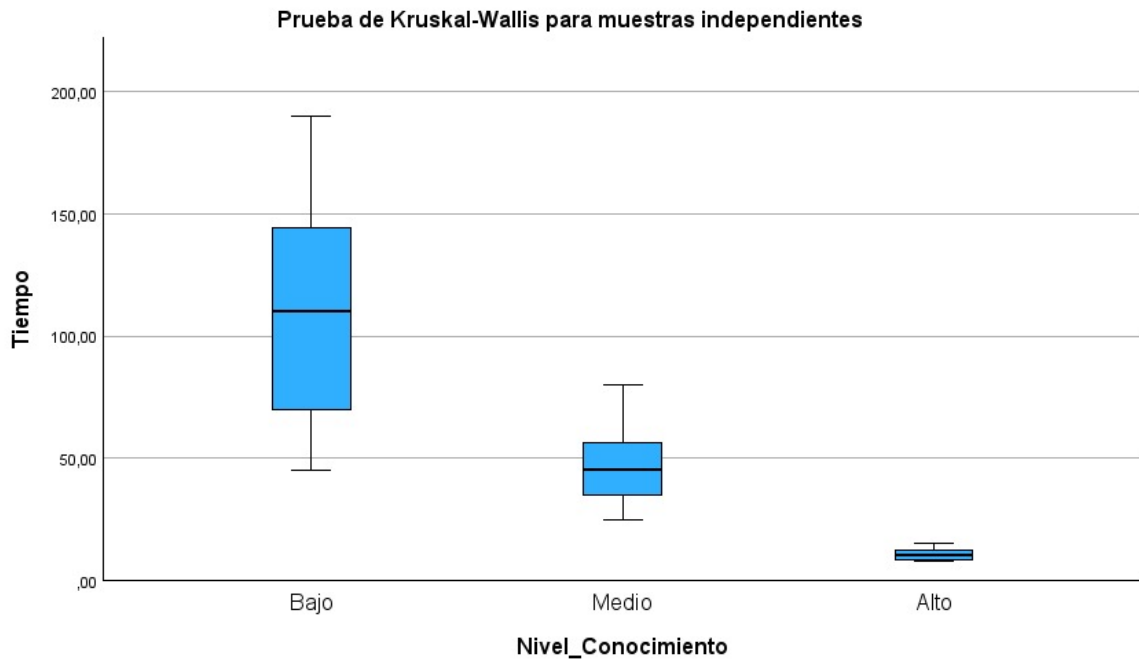


Figura 40. Diagrama de caja del Tiempo por Nivel\_Conocimiento.

En la figura 40 se observa una clara relación: a medida que aumenta el nivel de conocimiento, el Tiempo empleado para realizar la tarea disminuye, lo que confirma el impacto de esta variable en el rendimiento.

## CAPITULO V CONCLUSIONES Y RECOMENDACIONES

### 5.1 CONCLUSIONES

- El sistema de clasificación basado en Visión Artificial con el modelo YOLOv11 demuestra ser significativamente más eficiente en tiempo de respuesta que el método manual tradicional. Con la prueba U de Mann-Whitney se confirma una diferencia significativa, con el método automático teniendo un Rango promedio de 2.50 y una Media de 7.3450 segundos, en comparación con el método manual, que tuvo un Rango promedio de 20.50 y una Media de 53.8438 segundos.
- La selección de la arquitectura YOLOv11 y de la plataforma NVIDIA Jetson Nano resulta óptima para el objetivo de detección en tiempo real. El modelo entrenado alcanza una precisión (mAP@0.5) de 0,990, con un punto operativo de F1 máximo cercano a un umbral de confianza aproximado de 0,63, lo que evidencia un equilibrio sólido y robusto para la clasificación precisa de las tres especies objetivo: *Ceratitis capitata*, *Anastrepha serpentina* y *Anastrepha striata*.
- El sistema propuesto elimina la dependencia de personal altamente especializado, reduciendo los requisitos de experiencia y la subjetividad en la identificación de plagas. La comparación con el desempeño humano evidencia que el tiempo de identificación manual disminuye de manera significativa a medida que aumenta el nivel de conocimiento; sin embargo, esta variabilidad se ve mitigada mediante la automatización, la cual proporciona una exactitud constante y replicable comprendida entre el 95 % y el 99 %.

## 5.2 RECOMENDACIONES

- Se recomienda seguir recolectando y etiquetando imágenes de las moscas de la fruta en diferentes estados de conservación y condiciones de captura, incluyendo la integración con trampas electrónicas en campo. La ampliación del dataset con datos de campo ayudará a mejorar la generalización del modelo YOLOv11 ante variaciones no controladas de iluminación y posición.
- Para asegurar la máxima eficiencia del sistema Jetson Nano, se sugiere priorizar la optimización del modelo mediante TensorRT para reducir el tiempo de inferencia. Esto es crucial para mantener la velocidad de procesamiento requerida en el despliegue del prototipo de bajo consumo en entornos agrícolas.
- Se recomienda desarrollar una fase posterior del proyecto enfocada en la integración completa del dispositivo Jetson Nano con una arquitectura de Internet de las Cosas (IoT). Esto permitirá enviar los resultados de la clasificación (logs de detección y reportes en PDF/CSV) de manera autónoma y en tiempo real a una plataforma centralizada (servidor/nube), facilitando la generación de alertas automáticas para que Agrocalidad pueda aplicar acciones fitosanitarias inmediatas y fortalecer la gestión integral de plagas.



## BIBLIOGRAFIA

- [1] J. E. Vilatuña, R. David, P. Sandoval, L. Juan, and O. S. Tigrero, *MANEJO Y CONTROL DE MOSCAS DE LA FRUTA*, Primera Edición. Quito: Agencia Ecuatoriana de Aseguramiento de la Calidad del Agro Agrocalidad., 2010. Accessed: Dec. 03, 2025. [Online]. Available: [https://www.researchgate.net/publication/277060787\\_Manejo\\_y\\_control\\_de\\_moscas\\_de\\_Fruta](https://www.researchgate.net/publication/277060787_Manejo_y_control_de_moscas_de_Fruta)
- [2] M. Alejandro Bermúdez Vera, O. A. Fosado Tellez, and E. G. Cañarte Bermúdez, “MOSCAS DE LA FRUTA (DIPTERA: TEPHRITIDAE) Y SUS HOSPEDEROS EN EL ÁREA DEL CARRIZAL-CHONE, MANABÍ,” *Revista ESPAMCIENCIA para el agro*, vol. 11, no. 1, Jun. 2020, Accessed: Dec. 03, 2025. [Online]. Available: <http://repositorio.iniap.gob.ec/handle/41000/5559>
- [3] R. Hernández López, V. López Martínez, P. Juárez López, I. Alia Tejacal, D. Guillén Sánchez, and R. Hernández Pérez, “EVALUACIÓN DE ATRAYENTES ALIMENTICIOS Y TRAMPAS PARA LA CAPTURA DE MOSCAS DE LA FRUTA (DIPTERA: TEPHRITIDAE) EN NARANJA (CITRUS SINENSIS (L.) OSBECK) EN TEPALCINGO, MORELOS, MÉXICO,” *REVISTA METROPOLITANA DE CIENCIAS APLICADAS*, vol. 4, no. 3, Sep. 2021, doi: <https://doi.org/10.62452/nb58ry98>.
- [4] D. Brandt *et al.*, “Low Cost Machine Vision for Insect Classification,” Apr. 2024. doi: 10.48550/arXiv.2404.17488.
- [5] L. E. Cajahuishca Cando, “MONITOREO DE LA MOSCA DE LA FRUTA (*Anastrepha* spp. Y *Ceratitis capitata*) EN EL CULTIVO DE LA MORA (*Rubus glaucus*) EN EL CANTON SIGCHOS PROVINCIA DE COTOPAXI 2023-2024,” Tesis de Grado, Universidad Técnica de Cotopaxi, Latacunga, Ecuador, 2024. [Online]. Available: <https://repositorio.utc.edu.ec/server/api/core/bitstreams/1c5c544f-7d92-4234-8bec-50ba6bec5c73/content>
- [6] S. Tariq, A. Hakim, A. A. Siddiqi, and M. Owais, “An image dataset of fruitfly species (*Bactrocera Zonata* and *Bactrocera Dorsalis*) and automated species classification through object detection,” *Data Brief*, vol. 43, Aug. 2022, doi: 10.1016/J.DIB.2022.108366.
- [7] J. G. Ortega *et al.*, *Entomología aplicada para Agropecuarios*. Jipijapa, Manabí: Grupo COMPAS, Universidad Estatal del Sur de Manabí, 2021. [Online]. Available: [https://www.researchgate.net/profile/Julio-Gabriel/publication/349607934\\_Libro\\_entomologia\\_Febrero\\_23\\_2021/links/6037f6f24585158939cda53f/Libro-entomologia-Febrero-23-2021.pdf](https://www.researchgate.net/profile/Julio-Gabriel/publication/349607934_Libro_entomologia_Febrero_23_2021/links/6037f6f24585158939cda53f/Libro-entomologia-Febrero-23-2021.pdf)
- [8] G. A. Jorge C, *Guía armonizada de taxonomía e identificación de tefrítidos que pudieran ser considerados de importancia económica y cuarentenaria en América Latina y el Caribe*. Viena, Austria: Organismo Internacional de Energía Atómica (OIEA); Organismo Internacional Regional de Sanidad Agropecuaria (OIRSA), 219AD. [Online]. Available: <https://www.iaea.org/sites/default/files/guia210220.pdf>

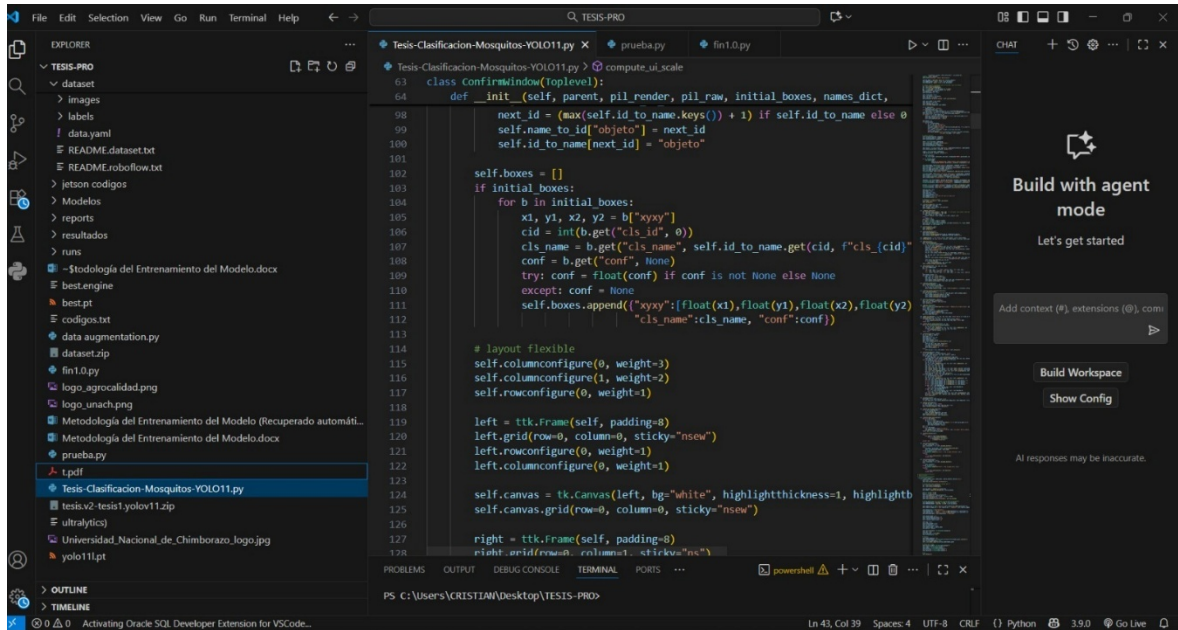
- [9] Agrocalidad, “MANUAL DE MANEJO INTEGRADO DE MOSCAS DE LA FRUTA,” Quito, Ecuador, 2025. [Online]. Available: [https://www.agrocalidad.gob.ec/wp-content/uploads/2025/08/document.send\\_.php-8.pdf](https://www.agrocalidad.gob.ec/wp-content/uploads/2025/08/document.send_.php-8.pdf)
- [10] L. M. Reimundo Topón, ““CICLO BIOLÓGICO DE LA MOSCA DE LA FRUTA DEL GÉNERO (*Anastrepha* spp.) A DOS TEMPERATURAS, SALACHE – COTOPAXI 2020,”” Tesis de Grado, Universidad Técnica de Cotopaxi, Latacunga, Ecuador, 2020. [Online]. Available: <https://repositorio.utc.edu.ec/server/api/core/bitstreams/cf19caa1-bda6-4514-be4d-ed984e6f606d/content>
- [11] V. Hernández Ortiz, J. Guillén Aguilar, and L. López, “5. taxonomía e Identificación de Moscas de la Fruta de Importancia económica en américa,” in *Moscas de la Fruta: Fundamentos y Procedimientos para su Manejo*, P. Montoya, J. Toledo, and E. Hernández, Eds., México, D.F.: S y G Editores, 2010, pp. 49–80. [Online]. Available: [https://www.researchgate.net/profile/Pablo-Montoya-2/publication/332446868\\_Montoya\\_P\\_J\\_Toledo\\_E\\_Hernandez\\_Eds\\_Mosca\\_de\\_la\\_Fruta\\_Fundamentos\\_y\\_Procedimientos\\_para\\_su\\_Manejo\\_2010/links/5cb5fc7ba6fdc1d499a1b1d/Montoya-P-J-Toledo-E-Hernandez-Eds-Mosca-de-la-Fruta-Fundamentos-y-Procedimientos-para-su-Manejo-2010.pdf](https://www.researchgate.net/profile/Pablo-Montoya-2/publication/332446868_Montoya_P_J_Toledo_E_Hernandez_Eds_Mosca_de_la_Fruta_Fundamentos_y_Procedimientos_para_su_Manejo_2010/links/5cb5fc7ba6fdc1d499a1b1d/Montoya-P-J-Toledo-E-Hernandez-Eds-Mosca-de-la-Fruta-Fundamentos-y-Procedimientos-para-su-Manejo-2010.pdf)
- [12] M. Jelali, “Deep learning networks-based tomato disease and pest detection: a first review of research studies using real field datasets,” Oct. 25, 2024, *Frontiers Media SA*. doi: 10.3389/fpls.2024.1493322.
- [13] M. El Sakka, M. Ivanovici, L. Chaari, and J. Mothe, “A Review of CNN Applications in Smart Agriculture Using Multimodal Data,” *Sensors*, vol. 25, no. 2, Jan. 2025, doi: 10.3390/s25020472.
- [14] C. Stryker and E. Kavlakoglu, “¿Qué es la inteligencia artificial o IA?,” IBM. Accessed: Dec. 03, 2025. [Online]. Available: <https://www.ibm.com/mx-es/think/topics/artificial-intelligence>
- [15] M. Larrazabal, “El impacto de la Inteligencia Artificial en la agricultura: Potencial y oportunidades (I),” Interempresas. [Online]. Available: [https://www.interempresas.net/Grandes-cultivos/600324-El-impacto-de-la-Inteligencia-Artificial-en-la-agricultura-Potencial-y-oportunidades-\(I\).html](https://www.interempresas.net/Grandes-cultivos/600324-El-impacto-de-la-Inteligencia-Artificial-en-la-agricultura-Potencial-y-oportunidades-(I).html)
- [16] H. Mehrnaz and I. Hasibul, “Aggrotech: Leveraging Deep Learning for Sustainable Tomato Disease Management,” *ArXiv*, Jan. 2025, [Online]. Available: <https://doi.org/10.48550/arXiv.2501.12052>
- [17] L. Zhang *et al.*, “Enhancing Fruit Fly Detection in Complex Backgrounds Using Transformer Architecture with Step Attention Mechanism,” *Agriculture (Switzerland)*, vol. 14, no. 3, Mar. 2024, doi: 10.3390/agriculture14030490.
- [18] S. Hou, Y. Pang, J. Wang, J. Hou, and B. Wang, “RS-YOLO: A highly accurate real-time detection model for small-target pest,” *Smart Agricultural Technology*, vol. 12, p. 101212, Dec. 2025, doi: 10.1016/J.ATECH.2025.101212.
- [19] A. Tripathi, U. Chourasia, P. Dixit, V. Chang, and R. Gandhi Proudhyogiki Vishwavidyalaya, “A Survey: Plant Disease Detection Using Deep Learning,” *IGI*

- Global*, vol. 12, no. 3, Jul. 2021, [Online]. Available: <https://www.igi-global.com/gateway/article/284431>
- [20] F. Lello *et al.*, “Fruit fly automatic detection and monitoring techniques: A review,” *Smart Agricultural Technology*, vol. 5, p. 100294, Oct. 2023, doi: 10.1016/J.ATECH.2023.100294.
- [21] A. Bhujel, N. E. Kim, E. Arulmozhi, J. K. Basak, and H. T. Kim, “A Lightweight Attention-Based Convolutional Neural Networks for Tomato Leaf Disease Classification,” *Agriculture (Switzerland)*, vol. 12, no. 2, Feb. 2022, doi: 10.3390/agriculture12020228.
- [22] S. Aboelenin, F. A. Elbasheer, M. M. Eltoukhy, W. M. El-Hady, and K. M. Hosny, “A hybrid Framework for plant leaf disease detection and classification using convolutional neural networks and vision transformer,” *Complex and Intelligent Systems*, vol. 11, no. 2, Feb. 2025, doi: 10.1007/s40747-024-01764-x.
- [23] B. Cardoso, C. Silva, J. Costa, and B. Ribeiro, “Internet of Things Meets Computer Vision to Make an Intelligent Pest Monitoring Network,” *Applied Sciences (Switzerland)*, vol. 12, no. 18, Sep. 2022, doi: 10.3390/app12189397.
- [24] V. Kumar, K. V. Sharma, N. Kedam, A. Patel, T. R. Kate, and U. Rathnayake, “A comprehensive review on smart and sustainable agriculture using IoT technologies,” *Smart Agricultural Technology*, vol. 8, Jun. 2024, doi: 10.1016/j.atech.2024.100487.
- [25] A. Franco Mesa, “Análisis de las Técnicas de Prevención y Detección de Enfermedades y Plagas en Cultivos Agrícolas, Desarrolladas Bajo Tecnologías de IoT,” Tesis de Grado, Universidad Nacional Abierta y a Distancia, 2023. [Online]. Available: <https://repository.unad.edu.co/bitstream/handle/10596/57795/afrancome.pdf?sequence=4&isAllowed=y>
- [26] Y. Uzun, M. R. Tolun, H. T. Eyyuboglu, and F. Sari, “An intelligent system for detecting Mediterranean fruit fly [Medfly; *Ceratitis capitata* (Wiedemann)],” *Journal of Agricultural Engineering*, vol. 53, no. 3, Sep. 2022, doi: 10.4081/jae.2022.1381.
- [27] A. S. Toasa Tipantuña and S. J. Gaibor Quispe, “APLICACIÓN DE LA INTELIGENCIA ARTIFICIAL EN EL SECTOR AGRÍCOLA,” *Revista Observatorio de las Ciencias Sociales en Iberoamérica*, vol. 3, no. 3, Jun. 2022, [Online]. Available: [https://www.researchgate.net/publication/361994199\\_APLICACION\\_DE\\_LA\\_INTELIGENCIA\\_ARTIFICIAL\\_EN\\_EL\\_SECTOR\\_AGRICOLA/link/62d03e24beb04023187645d4/download?\\_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19](https://www.researchgate.net/publication/361994199_APLICACION_DE_LA_INTELIGENCIA_ARTIFICIAL_EN_EL_SECTOR_AGRICOLA/link/62d03e24beb04023187645d4/download?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19)
- [28] R. Khanam and M. Hussain, “YOLOv11: An Overview of the Key Architectural Enhancements,” *ArXiv*, Oct. 2024, [Online]. Available: <http://arxiv.org/abs/2410.17725>
- [29] Ultralytics, “Ultralytics YOLO11, Documentación de Ultralytics YOLO.” Accessed: Dec. 03, 2025. [Online]. Available: <https://docs.ultralytics.com/es/models/yolo11/>
- [30] A. Vina, “YOLO11 Anchor-Free Detection: Benefits | Ultralytics.” [Online]. Available: <https://www.ultralytics.com/blog/benefits-ultralytics-yolo11-being-anchor-free-detector>

- [31] A. Khanna and S. Kaur, "Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture," *Comput Electron Agric*, vol. 157, Jan. 2019, doi: 10.1016/J.COMPAG.2018.12.039.
- [32] V. A. M. Martins, L. C. Freitas, M. S. De Aguiar, L. B. De Brisolara, and P. R. Ferreira, "Deep Learning applied to the Identification of Fruit Fly in Intelligent Traps," Brasil. [Online]. Available: [https://www.researchgate.net/publication/340227872\\_Deep\\_Learning\\_applied\\_to\\_the\\_Identification\\_of\\_Fruit\\_Fly\\_in\\_Intelligent\\_Traps](https://www.researchgate.net/publication/340227872_Deep_Learning_applied_to_the_Identification_of_Fruit_Fly_in_Intelligent_Traps)
- [33] M. Molina-Rotger, A. Morán, M. A. Miranda, and B. Alorda-Ladaria, "Remote fruit fly detection using computer vision and machine learning-based electronic trap," *Front Plant Sci*, vol. 14, Oct. 2023, doi: 10.3389/fpls.2023.1241576.
- [34] D. Čirjak, I. Miklečić, D. Lemić, T. Kos, and I. P. Živković, "Automatic Pest Monitoring Systems in Apple Production under Changing Climatic Conditions," *Horticulturae*, vol. 8, no. 6, Jun. 2022, doi: 10.3390/horticulturae8060520.
- [35] Y. Diller *et al.*, "A real-time remote surveillance system for fruit flies of economic importance: sensitivity and image analysis," *J Pest Sci (2004)*, vol. 96, no. 2, Jun. 2022, doi: 10.1007/s10340-022-01528-x.
- [36] E. A. B. Flores, H. V. Olivera, I. C. M. Valencia, and C. F. M. Cubas, "Fruit Fly Classification (Diptera: Tephritidae) in Images, Applying Transfer Learning," *ArXiv*, Feb. 2025, [Online]. Available: <http://arxiv.org/abs/2502.00939>
- [37] Á. Fragua Baeza, "EJECUCIÓN DE REDES NEURONALES EN MÓVILES ANDROID CON ACELERACIÓN HARDWARE MEDIANTE KERAS Y TENSORFLOW LITE," Trabajo de Grado, UNIVERSIDAD AUTÓNOMA DE MADRID, ESCUELA POLITÉCNICA SUPERIOR, Madrid, 2021. [Online]. Available: <http://hdl.handle.net/10486/698211>
- [38] T. Zhou, W. Zhan, and M. Xiong, "A series of methods incorporating deep learning and computer vision techniques in the study of fruit fly (Diptera: Tephritidae) regurgitation," *Front Plant Sci*, vol. 14, Jan. 2023, doi: 10.3389/fpls.2023.1337467.
- [39] E. J. Ruiz Mundaca and M. E. Ortiz Rojas, "Desarrollo de un análisis comparativo de desempeño computacional entre Nvidia Jetson Nano y el Raspberry Pi CM4, en el contexto de clasificación de frutas mediante algoritmos basados en Máquinas de Soporte Vectorial (SVM) y Redes Neuronales Convolucionales (CNN)," Tesis de Grado, Universidad Peruana de Ciencias Aplicadas, 2024. Accessed: Dec. 03, 2025. [Online]. Available: <https://repositorioacademico.upc.edu.pe/handle/10757/683333>
- [40] Coral, "System-on-Module datasheet," 2020. Accessed: Dec. 03, 2025. [Online]. Available: <https://www.coral.ai/docs/som/datasheet/>
- [41] C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens, "Rethinking the Inception Architecture for Computer Vision," *ArXiv*, Dec. 2015, Accessed: Dec. 03, 2025. [Online]. Available: <https://arxiv.org/abs/1512.00567>
- [42] M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," Jun. 2021, [Online]. Available: <http://arxiv.org/abs/2104.00298>

## ANEXOS

### Anexo A. Código Python en Visual Studio Code.



### Anexo B. Montaje del dispositivo.

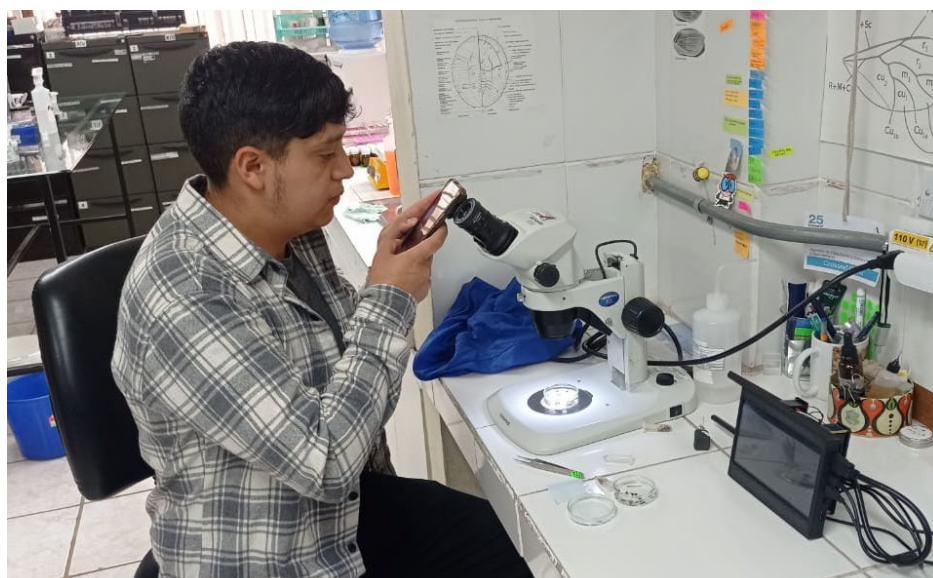




**Anexo C.** Agencia de Regulación y Control Fito y Zoosanitario del Ecuador sede Tumbaco.



**Anexo D.** Preparación de pruebas.



**Anexo F.** Adecuación del ambiente de trabajo.

