

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE TIANGUISTENCO

DIVISIÓN DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES

**Sistema de conteo y madurez del agave con inteligencia
artificial**

TESIS

Para obtener el título de
INGENIERO EN SISTEMAS COMPUTACIONALES

Presenta:

Luis Manuel Morales López

Directora de Teis:

Mtra. María Guadalupe Pineda Arizmendi

Co-directores:

Ing. Nayeli Zapata Álvarez

Ing. Jorge Díaz Landa

Santiago Tianguitenco, Estado de
México, julio 2026

HOJA DE PERMISO DE DATOS

Yo, **Luis Manuel Morales López**, titular de los derechos de autor del conjunto de datos y código asociado al proyecto **SISTEMA DE CONTEO Y MADUREZ DEL AGAVE CON INTELIGENCIA ARTIFICIAL**, autorizo expresamente su uso para fines académicos y de investigación. La única condición establecida es la obligatoria conservación y atribución de mi autoría original en cualquier reproducción o derivación. El material se proporciona "tal cual", eximiéndome de toda responsabilidad por su uso.

<https://github.com/LuisManuel222/Sistema-de-monitoreo-de-Agave-con-YOLOv5.git>

RESUMEN

El siguiente proyecto propone el desarrollo de un sistema de conteo y clasificación de madurez de agave a través de video en tiempo real, principalmente para municipios productores dentro del Estado de México. El problema actualmente en la supervisión de cultivos radica principalmente en los métodos manuales de conteo y evaluación de madurez, ya que estos resultan lentos y con margen de error alto; esta situación limita y retrasa la toma de decisiones de los procesos de producción y manejos del cultivo.

Para cubrir esta necesidad, el sistema sugiere poner en práctica un modelo de visión por computadora que facilite la identificación de cada planta para agilizar el conteo y clasificación de madurez de las plantas de agave en tiempo real a través de video. La clasificación de madurez se basa en atributos visuales como la forma, el color y el tamaño de la roseta. Asimismo, se incorpora un módulo de seguimiento de objetos que reconoce cada agave por separado y evita los conteos dobles; de esta forma se asegura que los resultados son más fiables en terrenos más extensos.

La categorización en las etapas de desarrollo, que incluyen los estados pequeño, joven y maduro, facilita la creación de información valiosa para optimizar los recursos en el manejo del cultivo y planificar la cosecha. El proyecto es una solución realista bajo un entorno de bajos recursos, ya que se utilizan tecnologías comunes, como vehículos aéreos no tripulados (drones), y procesamiento local en computadoras. De la misma forma, su uso permite disminuir costos operativos, eliminar errores humanos, cubrir grandes extensiones de tierra y obtener datos confiables en tiempo real. Con ello, se contribuye al fortalecimiento de la agricultura de precisión en el sector agavero.

ABSTRACT

This project proposes the development of a real-time video-based system for counting and classifying agave maturity, aimed primarily at production municipalities in the State of Mexico. Current crop monitoring relies on manual counting and maturity assessment, which is slow and prone to high error rates, limiting timely decision-making in production and crop management.

To address this need, the system incorporates a computer vision model capable of identifying each plant to accelerate the counting process and classify agave maturity in real time through video. The classification is based on visual attributes such as shape, color, and rosette size. Additionally, an object-tracking module is included to distinguish each agave individually and prevent double counting, ensuring more reliable results across large fields.

The categorization of development stages—young, intermediate, and mature—supports the generation of valuable information to optimize crop management resources and improve harvest planning. The project represents a realistic and low-cost solution by utilizing common technologies such as drones and local computer processing. Its implementation reduces operational costs, eliminates human errors, covers large land areas, and provides real-time, reliable data. Overall, this work contributes to strengthening precision agriculture within the agave sector.

AGRADECIMIENTOS

Principalmente, quiero agradecer a Dios por darme la fuerza, la claridad y la vida para poder culminar esta etapa de mi vida. Por guiarme en cada paso y brindarme la sabiduría necesaria para seguir adelante incluso en los momentos difíciles.

De igual manera, quiero agradecer a mi mamá, Rosalva López León, por su amor incondicional, por el apoyo en todo momento, por ser mi refugio, por creer en mí y en mis decisiones, y por enseñarme el valor de la dedicación. Gracias por todos los consejos y por ser mi mayor motivación para superar cada etapa de mi vida.

A mi papá, Manuel Morales Molina, por su dedicación y esfuerzo de cada día, para darme lo mejor, por creer en mí y en mi potencial y apoyarme sin esperar nada a cambio. Gracias por tus palabras de aliento y por ayudarme a levantarme cuando más lo necesite, eres mi mayor ejemplo de superación.

Este logro es un fruto de su sacrificio y de todo lo que han hecho para ayudarme a cumplir mis metas.

A mis hermanas, Yareli Morales López y Mitzi Guadalupe Morales López, quienes con su compañía, cariño y palabras de aliento me recordaron que nunca estoy solo. Su apoyo y compañía han sido fundamentales para poder mantenerme firme hasta el final de este camino.

A mis amigos, que estuvieron conmigo en los momentos de estrés, cansancio y celebración. Gracias por su compañía, sus palabras de ánimo, por las risas que aliviaron el camino y por estar presente cuando lo necesité, su amistad hizo este proceso más llevadero y especial.

A la maestra María Guadalupe Pineda Arizmendi, por su guía, paciencia y compromiso durante el desarrollo de este proyecto. Su apoyo y compañía han sido clave para poder realizar mi trabajo; agradezco profundamente cada una de sus enseñanzas, sugerencias y el tiempo dedicado a mi formación.

A mi abuelita, que, aunque ya no está físicamente, su amor y sus enseñanzas me acompañan en cada paso que doy. Esta tesis es para ti, por ser una inspiración y mi estrella más brillante.

Por último, quiero agradecer a todas las personas que de una u otra manera contribuyeron y apoyaron a la culminación de esta etapa; mi más sincero agradecimiento. Cada apoyo, por pequeño que parezca, fue importante para llegar hasta aquí.

DEDICATORIA

Quiero dedicar principalmente mi trabajo a Dios, por ser mi guía constante, por otorgarme la fuerza y la sabiduría necesaria para poder superar cada desafío que encontré en este camino. Gracias por iluminar mis pasos, por llenarme de esperanza en los momentos más difíciles y por permitirme llegar hasta este punto de mi vida con salud, claridad y determinación.

Dedico esta tesis a mis padres quienes fueron mi mayor inspiración y mi apoyo más firme y sincero. Gracias por su amor incondicional, por cada sacrificio silencioso y por enseñarme, con su ejemplo, que los sueños se alcanzan con esfuerzo, dedicación y perseverancia. Este logro representa no solo mi trabajo, sino también todo lo que ustedes han hecho para que yo pueda crecer, aprender y avanzar. Con profundo cariño les dedico este triunfo.

Dedico este trabajo a mis hermanas, por su compañía incondicional, sus palabras de ánimo y su apoyo constante. Gracias por recordarme que la familia es un refugio donde siempre puedo encontrar fuerza y motivación para seguir adelante.

Dedico este trabajo a mi abuelita, quien con su carácter fuerte y su ejemplo de trabajo incansable me enseñó que los sueños se construyen con dedicación. Hoy, al culminar esta etapa, sé que estás orgullosa desde el cielo.

Finalmente, dedico este esfuerzo a todas las personas que han formado parte de mi camino académico y personal. Gracias por creer en mí, por acompañarme y por contribuir de alguna manera a que hoy pueda cumplir esta meta.

ÍNDICE GENERAL

APROBACIÓN DE IMPRESIÓN DE TESIS	II
HOJA DE ORIGINALIDAD	III
HOJA DE SIMILITUD DE IA	IV
HOJA DE PERMISO DE DATOS	VI
RESUMEN	VII
ABSTRACT	VIII
DEDICATORIA	XI
ÍNDICE GENERAL	XII
ÍNDICE DE TABLAS	XV
ÍNDICE DE FIGURAS	XVI
1. INTRODUCCIÓN	18
2. ANTECEDENTES	20
3. MARCO TEÓRICO	23
3.1 Características del agave y clasificación	23
3.2 Agricultura de precisión y monitoreo de agave	24
3.3 Tecnologías de adquisición de datos	24
3.4 Visión por Computadora y Procesamiento de Imágenes	26
3.5 Aprendizaje automático y profundo	27
3.6 Desarrollo y entrenamiento de modelos	28
3.7 Seguimiento y conteo en tiempo real	30
3.8 Evaluación del modelo	31
3.9 Despliegue e Implementación del Sistema	33
3.10 Referentes Técnicos y Estudios Relacionados	34
3.11 Metodología CRISP-DM	35
4. PLANTEAMIENTO DEL PROBLEMA	38
5. JUSTIFICACIÓN	40

6. OBJETIVOS	42
<i>6.1 Objetivo general</i>	42
<i>6.2 Objetivos específicos</i>	42
7. HIPÓTESIS	43
8. METODOLOGÍA	44
<i>8.1 Comprensión del negocio</i>	45
<i>8.1.1 Objetivos del proyecto</i>	45
<i>8.1.2 Limitaciones del entorno</i>	46
<i>8.2 Comprensión de los datos</i>	47
<i>8.3 Preparación de los datos</i>	48
<i>8.4 Modelado</i>	49
<i>8.5 Evaluación</i>	50
<i>8.6 Implementación</i>	50
<i>8.7 Despliegue y mantenimiento</i>	51
<i>8.8 Resultados</i>	51
9. RESULTADOS Y DISCUSIÓN	53
<i>9.1 Análisis del sistema</i>	53
<i>9.2 Diseño del sistema</i>	55
<i>9.3 Desarrollo del sistema</i>	56
<i>9.4 Desarrollo de la aplicación web</i>	69
<i>9.5 Pruebas y validación del sistema</i>	74
<i>9.6 Resultados obtenidos</i>	76
<i>9.7 Elementos creados durante el desarrollo</i>	77
<i>9.8 Resumen</i>	77
10. CONCLUSIONES	78
11. RECOMENDACIONES	79
12. FUENTES BIBLIOGRÁFICAS	80
13. ANEXOS	86
ANEXO 1 A MANUAL DE USUARIO	86
ANEXO 1 B MANUAL TÉCNICO	100

ANEXO 2 A	107
ANEXO 2 B	108
ANEXO 2 C	109

ÍNDICE DE TABLAS

Tabla 1 Objetivos del proyecto. (Elaboración propia).....	46
Tabla 2 Limitaciones del entorno. (Elaboración propia).	47
Tabla 3 Técnicas de preparación de los datos. (Elaboración propia).....	49
Tabla 4 Evaluación del sistema con métricas. (Elaboración propia).....	50
Tabla 5 Requerimientos funcionales del sistema. (Elaboración propia).	54
Tabla 6 Requerimientos técnicos del sistema. (Elaboración propia).....	55
Tabla 7 Clases del etiquetado. (Elaboración propia).....	61
Tabla 8 Pruebas y validación del sistema del primer modelo. (Elaboración propia).	75
Tabla 9 Pruebas y validación del sistema del segundo modelo. (Elaboración propia).	75

ÍNDICE DE FIGURAS

Figura 1 Agave del Estado de México. (Estado de México, 2018).	23
Figura 2 Agave Azul. (Rancho Tissue, 2025).....	23
Figura 3 Dron DJI mini 3. (Sevengib, 2024).	25
Figura 4 Metodología CRISP-DM.	36
Figura 5 Metodología CRISP-DM aplicada a mi proyecto. (Elaboración propia).....	44
Figura 6 Agricultor monitoreando un cultivo de agave. (EFE, 2025).	45
Figura 7 Tipo de iluminación. (Elaboración propia).	47
Figura 8 Tipo de distancias. (Elaboración propia).	47
Figura 9 Tipo de rotación. (Elaboración propia).	47
Figura 10 Preparación de los datos (Zoom y traslación). (Elaboración propia).	48
Figura 11 Preparación de los datos (Ajuste de brillo). (Elaboración propia).	48
Figura 12 Preparación de los datos (Rotación). (Elaboración propia).	49
Figura 13 Flujo de interacción de la interfaz. (Elaboración propia).	51
Figura 14 Pagina web con resultados (Elaboración propia).	51
Figura 15 Resultados de la detección en video. (Elaboración propia).	52
Figura 16 Agricultor monitoreando un cultivo de agave. (EFE, 2025).	53
Figura 17 Monitoreo y riego de agave con Dron DJI (Xinhua, 2024).	54
Figura 18 Imagen aérea recolectada para el conjunto de datos. (Elaboración propia).....	57
Figura 19 Imagen aérea recolectada para el conjunto de datos. (Elaboración propia).....	57
Figura 20 Imagen aérea recolectada para el conjunto de datos. (Elaboración propia).....	57
Figura 21 Selección y limpieza de imágenes (Primer conjunto de datos). (Elaboración propia).	58
Figura 22 Selección y limpieza de imágenes (Segundo conjunto de datos). (Elaboración propia).	59
Figura 23 Proceso de etiquetado en Roboflow, (Primer conjunto de datos). (Elaboración propia).....	59
Figura 24 Proceso de etiquetado en Roboflow, (Segundo conjunto de datos). (Elaboración propia).....	60
Figura 25 Etiquetado de las 4 clases del primer conjunto de datos. (Elaboración propia).	61
Figura 26 Etiquetado de las 3 clases del segundo conjunto de datos. (Elaboración propia).	62
Figura 27 Aumento de datos. (Elaboración propia).	63
Figura 28 División del conjunto de datos. (Elaboración propia).....	64
Figura 29 Preprocesamiento de imágenes. (Elaboración propia).....	65
Figura 30 Preprocesamiento de imágenes. (Elaboración propia).....	65
Figura 31 Entorno de Google Colab. (Elaboración propia).	66
Figura 32 Configuración del entorno de entrenamiento. (Elaboración propia).....	66

Figura 33 Clonación del repositorio de YOLOv5. (Elaboración propia).....	67
Figura 34 Definición del archivo de configuración. (Elaboración propia).	67
Figura 35 Ejecución del entrenamiento (Elaboración propia).	68
Figura 36 Ajuste de parámetros. (Elaboración propia).	68
Figura 37 Validación del modelo entrenado del primer conjunto de datos. (Elaboración propia).	68
Figura 38 Validación del modelo entrenado del segundo conjunto de datos. (Elaboración propia).....	69
Figura 39 Interfaz gráfica de la página web. (Elaboración propia).	69
Figura 40 Módulos de la interfaz. (Elaboración propia).	70
Figura 41 Video resultado con bounding boxes. (Elaboración propia).	70
Figura 42 Generación de tablas y reportes. (Elaboración propia).	71
Figura 43 Generación de graficas. (Elaboración propia).	71
Figura 44 Dashboard histórico. (Elaboración propia).	72
Figura 45 Dashboard histórico. (Elaboración propia).	72
Figura 46 Análisis automático. (Elaboración propia).	73
Figura 47 Tracking IoU. (Elaboración propia).....	74
Figura 48 Tracking IoU. (Elaboración propia).....	¡Error! Marcador no definido.
Figura 49 Resultados de video procesada con el modelo uno. (Elaboración propia).	76
Figura 50 Porcentaje de confianza mínima y máxima con el modelo uno. (Elaboración propia).....	76
Figura 51 Resultados de video procesada con el modelo dos. (Elaboración propia).	76
Figura 52 Porcentaje de confianza mínima y máxima con el modelo dos. (Elaboración propia).	77

1. INTRODUCCIÓN

El cultivo del agave en el Estado de México es la base principal de la industria mezcalera; refiere a un sector de gran importancia tanto económica como cultural y social en municipios como Malinalco, Ocuilan, Tenancingo, Zacualpan y Zumpahuacán, entre otros, donde se cultivan entre 970-984 hectáreas aproximadamente (Gobierno del Estado de México, 2021). Estas extensiones cultivadas buscan la conservación y poder ampliarse mediante políticas de fomento y producción de plantas en viveros estatales; esto refleja la relevancia estratégica del cultivo para mantener la sostenibilidad del sector mezcalero regional (Secretaría del campo, 2023).

A pesar de la importancia que tienen, los métodos tradicionales de monitoreo basados en conteo manual de plantas y clasificación de madurez presentan diferentes limitaciones, ya que estos requieren tiempo, personal especializado y generan márgenes de error del 15-25% que afectan la planificación productiva y la toma de decisiones (Calvario et al., 2020). Se ha demostrado que estos métodos dificultan la toma de decisiones oportunas sobre la fertilización, riego, control de plagas y programación de cosecha, que terminan impactando en la productividad y rentabilidad de las plantaciones agaveras (Tapia Rodríguez et al., 2023).

Diversas investigaciones que se han realizado han demostrado la viabilidad en la detección automática de plantas de agave y otros cultivos, y a pesar de esto aún no existe una brecha en el desarrollo de sistemas integrales que combinen el conteo preciso, la clasificación de madurez en tiempo real y una plataforma que sea accesible y que permita el uso directo por productores y técnicos agrícolas sin requerir conocimientos especializados.

El problema de investigación radica principalmente en la falta de herramientas tecnológicas accesibles y eficientes que permitan realmente hacer el conteo y la clasificación automática del estado de madurez del agave, de manera precisa en tiempo real y en cualquier momento, para mejorar la toma de decisiones que sirvan para la planificación productiva y el manejo sostenible de las plantaciones agaveras.

Para entender la problemática, el presente proyecto busca desarrollar un sistema de conteo y clasificación de madurez de agave basado en visión por computadora que procese video en tiempo real a través de drones DJI para identificar, contar y clasificar plantas de agave de manera precisa y eficiente.

El sistema utiliza redes neuronales convolucionales, en este caso YOLOv5, para detectar cada planta de manera individual mediante rectángulos delimitadores, demostrando una precisión superior al 95% en estudios similares de detección de agaves con drones (Hernández Calvario et al., 2022). Posteriormente, se clasifica el estado de madurez, analizando características físicas como tamaño, forma y coloración, basándose en criterios establecidos en investigaciones taxonómicas. Por último, se incorporó un módulo de seguimiento de objetos que evita los conteos dobles durante los vuelos dinámicos de dron, garantizando los resultados confiables en parcelas extensas.

El sistema se implementó mediante una aplicación web interactiva desarrollada en el framework de Streamlit que permite cargar imágenes, videos o transmisiones en vivo, mostrando resultados visuales claros con detecciones marcadas y conteos por categoría de madurez. La interfaz no requiere de conocimientos técnicos avanzados, facilitando su uso directo por productores y técnicos agrícolas.

El desarrollo de este tipo de tecnologías representa un avance tecnológico muy significativo para la agricultura de precisión en el sector del agave mexicano, reduciendo significativamente el tiempo y costos de monitoreo, además de mejorar la precisión de los datos críticos para la planificación productiva (Hernández Calvario et al., 2022).

2. ANTECEDENTES

La agricultura de precisión es utilizada para el eficiente manejo de cultivos; este tipo de agricultura busca integrar nuevas tecnologías como sensores remotos, vehículos aéreos no tripulados y algoritmos de inteligencia artificial. Con el apoyo de estas tecnologías se busca optimizar la producción agrícola y reducir significativamente costos operativos que están asociados a las prácticas tradicionales (Mora Castro et al., 2025).

El cultivo del agave, utilizado principalmente en la producción de bebidas fermentadas y otros derivados industriales, enfrenta un enorme reto, que va relacionado con la evaluación del estado del cultivo y la clasificación de la madurez de la planta (Mora Castro et al., 2025). Los métodos tradicionales utilizados para el conteo y el monitoreo manual resultan lentos, costosos por los salarios y la movilidad y son propensos a errores humanos (Hernández Calvario et al., 2022).

Este tipo de métodos utilizados limita la planificación de la cosecha y la optimización de los recursos, ya que el conteo manual tradicional toma más tiempo que los sistemas automatizados propuestos (Calvario et al., 2020). Derivado de estas limitaciones, se ha impulsado el desarrollo para sistemas basados en visión por computadora y aprendizaje automático para la detección y clasificación del agave, de esta manera evolucionando desde los métodos manuales deficientes hasta soluciones tecnológicas estables (Mora Castro et al., 2025; Flores et al., 2021).

Ante la problemática presentada, diversos estudios recientes han abordado el tema mediante el uso de imágenes aéreas de alta resolución y técnicas avanzadas de visión por computadora y procesamiento digital. Calvario y colaboradores, propusieron una metodología que fue pionera para el conteo de agave; se basa en una morfología matemática aplicada a imágenes RGB capturada por drones DJI. Lo realizaron mediante operaciones secuenciales de erosión, dilatación morfológica y reconstrucción topográfica; de esta forma lograron segmentar e identificar cada planta de agave de manera individual con una precisión que varía de 0.83 en parcelas heterogéneas, hasta 0.98 en plantaciones regulares con condiciones óptimas de

iluminación solar directa. Supero los métodos manuales tradicionales tanto en velocidad como en exactitud.

A su vez, Escobar Flores y colaboradores realizaron una investigación donde compararon de una manera exhaustiva 6 algoritmos de clasificación supervisada, todos aplicados a imágenes UAV para la detección autónoma de agaves. Los resultados fueron los siguientes: análisis orientado a objetos (0.914), máquinas de soporte vectorial (0.928), redes neuronales artificiales (0.914), distancia de Mahalanobis (0.752), distancia mínima (0.421) y K-means clustering. Con este tipo de evaluación sistemática seleccionaron los modelos más eficientes y precisos para aplicaciones de monitoreo a gran escala; de esta manera se confirmó la superioridad de técnicas orientadas a objetos sobre métodos espectrales tradicionales.

Por su parte, Hernández y otros autores desarrollaron un algoritmo híbrido que está basado en CNN y YOLOv5; fue entrenado con 5000 imágenes aéreas capturadas desde drones multirrotores equipados con cámaras RGB y cámaras térmicas. Implementaron aumento de datos, tales como rotaciones, flips horizontales-verticales y ajustes de brillo-contrastes, para tener una robustez ante variabilidad lumínica. El modelo alcanzó resultados de mAP@0.5 de 0.89 y una velocidad de procesamiento de 28 FPD; de igual manera, demostraron una viabilidad real para el monitoreo en tiempo real directamente desde drones y superaron significativamente la eficiencia de métodos manuales tradicionales.

Asimismo, Villatoro y otros autores, aplicaron nuevas técnicas innovadoras de segmentación para el Agave Weber utilizando drones multispectrales MicaSense RedEdge-MX y el índice ARVI (Atmospherically Resistant Vegetation Index) adaptado en espacio HSV. Esta estrategia les permitió discriminar con una gran exactitud la cobertura vegetal del agave frente al suelo desnudo y la maleza en condiciones reales de sombra parcial; obtuvieron métricas de rendimiento altas, precisión 0.94, Recall 0.91 y F1-score 0.925, mejorando de manera notable la toma de decisiones en la gestión del cultivo.

En la investigación de Espinoza y otros autores, emplearon redes neuronales convolucionales profundas sobre imágenes RGB que adquirieron por sistemas aéreos remotos pilotados (RPAS) para la estimación automática de densidad poblacional de agaves. La metodología que utilizaron incorporó ortocorrecciones geométricas, extracción de descriptores texturales

mediante matrices GLCM (Gray Level Co-occurrence Matrix) y clasificación supervisada; lograron una correlación $R^2=0.87$ entre conteos automáticos y validaciones manuales terrestres, lograron la automatización completa de la evaluación de grandes extensiones territoriales que de una forma tradicional requerían días.

De las ultimas investigaciones, López Alfaro y otros autores presentaron una metodología diferente que integraba vehículos aéreos no tripulados, así como procesamiento digital avanzado de imágenes aéreas y técnicas de Machine Learning para el reconocimiento preciso y la localización georreferenciada de plantas de agave, todo en condiciones de campo reales. Su pipeline validó la integración efectiva de captura UAV-procesamiento IA-localización GIS, así proporcionando una herramienta más escalable para inventarios agrícolas más extensivos.

Por último, estudios más recientes como Bit-STED de Villatoro Geronimo y otros autores, representan vanguardia tecnológica, demostrando el potencial de los transformadores ligeros optimizados para el conteo preciso de agave en imágenes UAV. Este tipo de modelo supera arquitecturas CNN tradicionales (YOLOv8 mAP@0.5: 0.884, Faster R-CNN: 0.901), logrando alcanzar mAP@0.5 de 0.937 con 62% menos parámetros computacionales, llegando a capturar dependencias espaciales de largo alcance mediante mecanismos de atención multi-cabeza y de esta manera demostrar la viabilidad práctica para la implementación en edge computing directamente en plataformas UAV.

Todos los antecedentes revisados evidencian un avance tecnológico exponencial en la aplicación de una sistemática de tecnologías de percepción remota y aprendizaje profundo a la agricultura de precisión en el sector agavero. La problemática tradicional del conteo manual y clasificación visual de madurez del agave, que se identificó como lenta, costosa y propensa a errores humanos significativos, ha sido superada mediante la integración de vehículos aéreos no tripulados, algoritmos avanzados de inteligencia artificial y análisis multiespectral automatizado, así habilitando sistemas de monitoreo continuo en tiempo real que optimizan drásticamente la planificación productiva, la gestión eficiente de recursos y la toma de decisiones estratégicas en la industria agavera.

3. MARCO TEÓRICO

3.1 Características del agave y clasificación

3.1.1 Agave

El agave es un género de planta suculenta perenne¹ nativa de regiones áridas y semiáridas de México, Estados Unidos y América Central; este tipo de planta es perteneciente a la familia de Asparagácea². Se caracteriza por formar rosetas basales de hijas carnosas, gruesas y fibrosas dispuestas en espiral alrededor de su tallo corto (Mendoza, 2007).



Figura 1 Agave del Estado de México. (Estado de México, 2018).

3.1.2 Agave azul

Es una planta suculenta CAM que acumula grandes cantidades de carbohidratos de reserva, que son principalmente fructanos, durante su desarrollo vegetativo. Estos fructanos son el principal sustrato fermentable en la producción de tequila (Guardado Fierros et al., 2024).



Figura 2 Agave Azul. (Rancho Tissue, 2025).

3.1.3 Jima (Cosecha del agave)

¹ Perennes - Algo continuo o que dura indefinidamente

² Asparagácea - Es una familia de plantas monocotiledóneas perteneciente al orden Asparagales

La jima se realiza cuando la planta ha alcanzado un estado de reserva de carbohidratos favorable para la producción fermentable. Esto suele coincidir con plantas en etapas avanzadas de desarrollo vegetativo (6–7 años), antes de la transición reproductiva (Hernández Soriano et al., 2023).

3.1.4 Fenología del agave

La fenología del agave sigue un patrón monocárpico; esto quiere decir que la planta crece vegetativamente durante varios años para poder acumular reservas de carbohidratos y así preparar la transición a la reproducción sexual solamente una vez. El inicio de la inflorescencia está asociado con cambios en vías genéticas con la edad y el metabolismo de azúcares, marcando el final de la etapa productiva (Hernández Soriano et al., 2023).

3.2 Agricultura de precisión y monitoreo de agave

3.2.1 Agricultura de precisión

Es una estrategia que ayuda a recopilar, procesar y analizar datos temporales y espaciales que optimiza el uso de insumos en la producción agrícola. Esta aproximación gestiona la variabilidad de cultivos a través de tecnologías site-specific que sirven para mejorar la rentabilidad y sostenibilidad (Mizik, 2022). De igual manera, se describe como sistemas que brindan apoyo a decisiones que maximizan el rendimiento y minimizan los impactos ambientales en unidades de tierra específicas (Pérez Ortega et al., 2021).

3.2.2 Internet de las cosas agrícola

Integra sensores interconectados que sirven para monitorear en tiempo real diferentes variables como el suelo y el clima en entornos rurales. Se entiende como una red que habilita agricultura de precisión inteligente mediante recopilación masiva de datos para decisiones más eficientes (Ahmed et al., 2018).

3.3 Tecnologías de adquisición de datos

3.3.1 Vehículos aéreos no tripulados

Se definen como plataformas aéreas autónomas que toman datos geoespaciales de alta resolución para un monitoreo agrícola más preciso. Estos permiten la captura sistemática de imágenes aéreas sobre grandes áreas para un análisis de variabilidad de cultivos (Tsouros et al., 2019). Los drones facilitan el mapeo no invasivo de estrés vegetal y requerimientos de insumos en tiempo real (Radoglou Grammatikis et al., 2020).



Figura 3 Dron DJI mini 3. (Sevengib, 2024).

3.3.1.1 Drones con cámara RGB

Los drones con cámara RGB capturan imágenes en bandas roja, verde y azul para generar mapas visuales de alta resolución (2-5 cm/píxel) en agricultura. Este tipo de cámaras convencionales detecta anomalías visibles como deficiencias nutricionales y daños mecánicos mediante fotogrametría básica. Estos proveen ortomosaicos RGB accesibles para análisis cualitativo inicial de salud del cultivo (Deng et al., 2018).

3.3.1.2 Drones con cámaras multispectrales

Registra reflectancia en múltiples bandas espectrales que son el verde, rojo, NIR, rojo borde; esto sirve para calcular índices de vegetación cuantitativos. Detectan estrés temprano invisible al ojo humano mediante diferencias espectrales, con aplicaciones en dosificaciones variables de agroquímicos. Sirven como base cartográfica para superposición de índices vegetativos y planos de aplicación (Liu et al., 2023).

3.3.2 Ortomosaicos

Los ortomosaicos son imágenes geoespacialmente corregidas generadas por fotogrametría aérea que combinan cientos de imágenes aéreas en mosaicos continuos sin distorsionarse, estos integran datos GPS/IMU de drones de precisión milimétrica, así permitiendo el análisis espacial de variabilidad agronómica (Nex & Remondino, 2014).

3.3.3 Percepción remota

La percepción remota en la agricultura captura datos sin contacto físico utilizando sensores electromagnéticos desde vehículos aéreos no tripulados o desde satélites. Mide la reflectancia foliar para evaluar vigor, agua y nutrientes a escala de campo mediante interacción luz-material vegetal (Thenkabail et al., 2019).

3.4 Visión por Computadora y Procesamiento de Imágenes

Rama de la inteligencia artificial que desarrolla algoritmos que computadoras pueden interpretar y que comprenden el contenido visual de imágenes y videos digitales, extrayendo así información semántica útil como objetos, escenas, patrones y relaciones espaciales de manera similar a la visión humana. Combina técnicas clásicas de procesamiento de señales con aprendizaje profundo moderno para automatizar tareas de percepción visual en tiempo real (Szeliski, 2021; Pajares & Cruz, 2006).

3.4.1 Inteligencia artificial

Es un conjunto de algoritmos y sistemas diseñados para imitar las capacidades cognitivas de un ser humano. En el ámbito científico, optimiza la precisión y la velocidad de los procesos mediante herramientas avanzadas para el análisis masivo de datos. Todas estas aplicaciones contribuyen a mejorar la calidad y la eficiencia de la ciencia (Galdames, 2023).

3.4.2 Procesamiento digital de imágenes

Es un conjunto sistemático de técnicas matemáticas que transforman imágenes digitales crudas o sin procesar en representaciones optimizadas para el análisis automático, aplicando operaciones como el filtrado espacial, corrección geométrica y segmentación por umbral. De esta manera, constituye el preprocesamiento esencial para flujos de trabajo de visión artificial (Gonzalez & Woods, 2018).

3.4.3 Detección de objetos

Tarea fundamental dentro de la visión artificial, ya que localiza y clasifica simultáneamente múltiples instancias de objetos en imágenes o videos, genera rectángulos delimitantes

precisos (bounding boxes) con coordenadas y etiquetas que categorizan, están acompañadas de probabilidades de confianza (Ren et al., 2015; Redmon et al., 2016).

3.4.4 Clasificación de imágenes

Es un proceso de aprendizaje supervisado que le asigna una categoría única a las imágenes, después de extraer características visuales jerárquicas mediante redes neuronales convolucionales profundas; estas pueden ser desde patrones de bajo nivel como bordes o texturas hasta conceptos semánticos complejos como objetos o escenas. La clasificación de imágenes revolucionó por competencias ImageNet que establecieron transfer learning como estándar, utilizando pesos pre-entrenados en miles de imágenes para acelerar la convergencia en datasets especializados (Russakovsky et al., 2015).

3.4.5 Segmentación

Divide imágenes en regiones semánticamente homogéneas que asignan etiquetas específicas a cada píxel mediante redes convolucionales que preservan información espacial completa, así eliminando la pérdida dimensional de capas fully-connected tradicionales (Long et al., 2014).

3.5 Aprendizaje automático y profundo

Es un subcampo de la IA que desarrolla algunos algoritmos capaces de aprenderse patrones complejos directamente de datos sin necesidad de una programación explícita, utilizando redes neuronales profundas con múltiples capas para extraer representaciones jerárquicas automáticas. El aprendizaje profundo supera métodos tradicionales mediante feature learning no supervisado (Goodfellow et al., 2016).

3.5.1 Aprendizaje profundo

Es una clase de aprendizaje automático que emplea redes neuronales artificiales con múltiples capas ocultas (deep neural networks) para modelar relaciones complejas en datos con alta dimensión, así como imágenes, texto y audio (LeCun et al., 2015).

3.5.2 Redes neuronales convolucionales

Son arquitecturas especializadas en procesamiento de imágenes que aplican filtros convolucionales locales (kernels 3x3, 5x5) compartidos espacialmente, seguidos de submuestreo (pooling max/average) para extraer características no variables a traslación (Krizhevsky et al., 2015; He et al., 2015).

3.5.3 Modelos de detección

Son algoritmos que combinan la clasificación de objetos y regresión de coordenadas espaciales para así predecir simultáneamente categorías y bounding boxes desde feature maps (Lin et al., 2014).

3.5.3.1 YOLO

YOLO divide imágenes en grids prediciendo directamente los bounding boxes, clases y scores desde feature maps en una sola pasada. Integra segmentación, clasificación y rastreos simultáneos (Redmon et al., 2016).

3.5.4 Data augmentation

Genera variantes sintéticas del dataset mediante transformaciones geométricas como rotación o crop, fotométricas como brillo, contraste o saturación y elásticas, preservando etiquetas. CutMix/MixUp blending superpone imágenes proporcionalmente a labels. AutoAugment busca políticas óptimas automáticamente. Expande datasets pequeños 10-50x, mejorando la generalización (Shorten & Khoshgoftaar, 2019).

3.6 Desarrollo y entrenamiento de modelos

Es un proceso completo de ingeniería que transforma datos crudos en sistemas de inteligencia artificial operativos a través de ciclos iterativos de experimentación, de optimización y validación. Estos incluyen un diseño arquitectural, gestión de datos, ajustes de hiperparámetros, evaluación cruzada y despliegue MLOps (Goodfellow et al., 2016).

3.6.1 Conjunto de datos etiquetados

Son colecciones de fotos donde manualmente se han marcado cada objeto, persona o elemento importante. Algunos ejemplos son: ImageNet (14 millones de imágenes clasificadas), COCO (objetos cotidianos con cajas y descripciones). Se dividen en

entrenamiento (70%), validación (15%) y prueba (15%) para evaluar objetivamente (Russakovsky et al., 2015).

3.6.2 Preprocesamiento de imágenes

Pasos previos que limpian y preparan imágenes para el entrenamiento, por ejemplo, cambiar de tamaño, corregir brillo/contraste, rotar imágenes, voltearlas horizontalmente, etc. De igual forma, elimina ruido, mejora la calidad y crea más ejemplos variados del mismo objeto desde diferentes ángulos (Shorten & Khoshgoftaar, 2019).

3.6.3 Entrenamiento, validación y prueba

Se dividen los datos en tres grupos separados: el entrenamiento enseña al modelo, la validación ajusta configuraciones (velocidad de aprendizaje) y la prueba mide el rendimiento real final. Igual evita que el modelo memorice datos específicos y funcione bien con fotos nuevas o diferentes a las utilizadas en el entrenamiento (Loshchilov & Hutter, 2017).

3.6.4 Plataformas de desarrollo

3.6.4.1 Google Colab

Plataforma gratuita en internet que proporciona computadoras potentes con GPU para poder entrenar modelos sin instalar nada. Se puede acceder desde cualquier navegador web, guarda archivos en Google Drive y permite compartir los proyectos fácilmente (Godin, 2020).

3.6.4.2 PyTorch / TensorFlow

Bibliotecas de programación altamente usadas para crear redes neuronales. La biblioteca de PyTorch es ideal para investigadores (más flexible), y la biblioteca de TensorFlow es más para aplicaciones comerciales (más estable). Pero ambas permiten entrenar modelos complejos en computadoras normales o servidores potentes (Paszke et al., 2019).

3.6.4.3 OpenCV

Librería gratuita que cuenta con miles de funciones listas para procesar imágenes y videos en tiempo real. Lee cámaras web, puede aplicar filtros, detecta caras/objetos y dibuja resultados. Es multiplataformas, ya que sirve en computadoras, celulares y cámaras inteligentes (Bradski, 2005).

3.7 Seguimiento y conteo en tiempo real

Procesos que permiten identificar y seguir objetos en movimiento a través de video en vivo, contando cuántos mismos objetos hay sin repetirse. Es esencial para vigilancia, control de tráfico y monitoreo agrícola automatizado (Bewley et al., 2016; Zhang et al., 2021).

3.7.1 Seguimiento de objetos

Tipo de algoritmo que mantiene la identidad de objetos específicos entre frames consecutivos de video, de esta forma prediciendo su nueva posición en cada imagen. Combina detección con la asociación temporal, manejando oclusiones y cambios de velocidad (Bewley et al., 2016).

3.7.2 Algoritmos de Tracking

3.7.2.1 ByTrack

Algoritmo simple, pero que es muy efectivo, asocia todas las detecciones (incluso las de baja confianza) usando movimiento y apariencia. Se baja el umbral de confianza temporalmente para recuperar objetos perdidos por oclusión (Zhang et al., 2021).

3.7.2.2 SORT

SORT (Simple Online and Realtime) combina el detector YOLO + Kalman filter para predicción de movimiento lineal + Hungarian algorithm para asociación óptima entre frames. SIMPLEST algoritmo efectivo, 30+ FPS, pero es sensible a oclusiones largas por falta de reidentificación (Bewley et al., 2016).

3.7.2.3 DeepSORT

Extiende a SORT agregando reidentificación profunda (CNN extractora de features + cosine distance) para distinguir objetos que son similares cuando reaparecen. Igual maneja mejores cambios de cámara y oclusiones prolongadas (Wojke et al., 2017).

3.7.3 Conteo sin duplicados

Se utilizan identificadores únicos (tracking IDs) para contar objetos una sola vez, aun cuando aparezcan en múltiples fotogramas. Entre las técnicas de conteo más comunes se encuentran el cruce de líneas virtuales (cross-line counting), el conteo por zonas definidas (zone counting) y los métodos de entrada-salida. Asimismo, el uso de filtros temporales permite

evitar conteos prematuros o duplicados durante el seguimiento de los objetos (Zhang et al., 2021).

3.7.4 Transmisión en tiempo real (RTSP, MQTT)

RTSP (Real Time Streaming Protocol) se utiliza para transmitir video continuo desde cámaras IP (usualmente codificado en H.264 o H.265) con baja latencia. MQTT es un protocolo ligero de mensajería que permite enviar alertas o metadatos con sobrecarga mínima. Un pipeline típico para transmisión y análisis en tiempo real en visión por computadora puede incluir: cámara IP → captura de video por RTSP → procesamiento con OpenCV → detección y seguimiento con redes como YOLO y FairMOT → publicación de conteos y eventos mediante MQTT hacia un dashboard web (Rejeb et al., 2022; Zhang et al., 2021).

3.8 Evaluación del modelo

Es un proceso sistemático que sirve para medir qué tan bien funciona un modelo de inteligencia artificial con datos nuevos que nunca ha visto. Compara predicciones del modelo con la realidad para poder identificar fortalezas, debilidades y áreas de mejora antes de usarlo en producción (Goodfellow et al., 2016).

3.8.1 Métricas de evaluación

3.8.1.1 Precisión

De todas las predicciones positivas emitidas por el modelo, qué proporción corresponde efectivamente a la clase de interés. Se calcula como:

$$Precisión = \frac{(VP)}{(VP) + (FP)}$$

donde VP (verdaderos positivos) son los aciertos y FP (falsos positivos) son los casos que el modelo clasificó erróneamente como positivos. Una precisión elevada implica baja tasa de falsas alarmas (Powers, 2011).

3.8.1.2 Recall

Se mide que tanta capacidad tiene el modelo para poder identificar todos los objetos reales de una clase o más. Se calcula de la siguiente manera: cociente entre los verdaderos positivos (VP) y la suma de verdaderos positivos más falsos negativos (FN). Refleja, de todas las instancias positivas reales, qué proporción fue correctamente detectada. Es crítica en aplicaciones donde omitir un objeto tiene consecuencias graves (Davis & Goadrich, 2006).

$$\text{Recall} = \frac{(\text{VP})}{(\text{VP}) + (\text{FN})}$$

3.8.1.3 F1-Score

Es el promedio equilibrado entre precisión y Recall (Van Rijsbergen, 2000).

$$\text{F1} = 2 \times \frac{\text{Precisión} \times \text{Recall}}{\text{Precisión} + \text{Recall}}$$

3.8.1.4 IoU

IoU (Intersection over Union) compara que tan bien coinciden las cajas que dibuja el modelo con las cajas que sí son reales. De igual manera, mide el área donde se superponen ambas cajas. Los valores altos (cerca de 1.0) indican excelente localización espacial; el $\text{IoU} > 0.5$ es considerado detección correcta (Everingham et al., 2009).

$$\text{IoU} = \frac{\text{Área de Intersección}}{\text{Área de Unión}} = \frac{A_p \cap A_g}{A_p \cup A_g}$$

3.8.1.5 mAP

Métrica global que resume la capacidad de detección del modelo para todas las clases y en diferentes niveles de exigencia de localización. Se calcula como el promedio de la precisión media (Average Precision) de cada clase (Lin et al., 2014).

$$\text{mAP} = \frac{1}{n} \sum_{i=1}^n \text{AP}_i$$

3.8.2 Matriz de confusión

La matriz de confusión es una tabla que cuenta con cuatro situaciones posibles:

- Los aciertos reales (verdaderos positivos): En este, el modelo detectó correctamente un objeto que si existe.
- Las falsas alarmas (falsos positivos): En este, el modelo marcó algo que en realidad no era un objeto.
- Los objetivos perdidos (falsos negativos): En este existe un objeto real, pero el modelo no lo detectó.
- Correctamente ignorados (verdaderos negativos): En este, el modelo ignoró correctamente áreas sin objeto. Esta es la base para calcular todas las demás métricas de rendimiento (Fawcett, 2006).

3.8.3 Sobreajuste y subajuste

El sobreajuste es el modelo que se aprende de memoria las fotos de entrenamiento, pero falla con las fotos nuevas.

El subajuste es un modelo muy básico que no aprende bien ni siquiera con los datos de entrenamiento (Goodfellow et al., 2016).

3.9 Despliegue e Implementación del Sistema

3.9.1 Edge Computing

El Edge Computing procesa todas las imágenes directamente en el dispositivo que se usa para la captura, ya sean con cámaras inteligentes, drones o computadoras embebidas, así en lugar de transmitir todo a servidores centralizados. Esto reduce significativamente el tiempo de respuesta a milisegundos, minimiza el consumo de ancho de banda y permite el funcionamiento autónomo en áreas sin conectividad estable. En resumen, solo transmite resultados resumidos, así manteniendo la privacidad de los datos originales (Shi et al., 2016).

3.9.2 Streamlit

Es framework de código abierto, basado en Python; permite desarrollar interfaces web que sean interactivas para aplicaciones de análisis de datos y aprendizaje automático de manera fácil y rápida (Martínez Rodríguez, 2024).

3.9.3 Reportes Automatizados

Estos reportes permiten generar documentos estructurados cada cierto tiempo que resumen métricas, resultados y estadísticas del sistema en función. Son esenciales para poder evaluar desempeños, trazabilidad de datos y las tomas de decisiones basadas en evidencias o hechos (Chanchí et al., 2021).

3.9.4 Sistemas de Alertas Tempranas

Los sistemas de alertas tempranas están diseñados para poder detectar condiciones anormales o eventos críticos y así notificar a los usuarios de manera pronta o inmediata. Este tipo de alertas en sistemas computacionales se basan en reglas, umbrales o modelos predictivos y son fundamentales para poder reducir riesgos, prevenir fallos y mejorar todo tipo de respuesta a eventos no esperados.

3.9.5 API REST

Arquitectura de servicios web que permite una comunicación entre sistemas mediante protocolos de HTTP. Facilita la interoperabilidad, el acceso remoto de datos y la integración entre diferentes plataformas, ampliamente utilizados en sistemas distribuidos y aplicaciones IoT (Pathmudi et al., 2023).

3.10 Referentes Técnicos y Estudios Relacionados

3.10.1 Conteo de Agave con Morfología Matemática

La morfología matemática es un conjunto de operaciones no lineales aplicadas al procesamiento de imágenes para analizar formas y estructuras. Estas técnicas han sido utilizadas en agricultura para segmentar plantas, eliminar ruido y realizar conteos automáticos basados en propiedades geométricas de los objetos detectados (Pathmudi et al., 2023).

3.10.2 Detección Automática con CNN y UAVs

Las redes convolucionales (CNN) combinadas con imágenes capturadas por drones se han consolidado como una técnica robusta para tareas de clasificación, segmentación y detección en agricultura de precisión. En revisiones especializadas, se reporta que estos métodos han permitido procesar grandes volúmenes de datos aéreos y extraer características relevantes

para poder identificar atributos como fenotípicos y condiciones de cultivo con alta precisión (Shahi T. et al., 2023).

3.10.3 Clasificación de Madurez con Imágenes Satelitales

Se logra mediante la extracción de índices espectrales y con la aplicación de modelos que correlacionan estos índices con características fisiológicas de las plantas; así permite determinar la etapa de desarrollo de manera remota y escalable (Sharma & Shivandu, 2024).

3.10.4 Modelos Ligeros para Drones (Bit-STED, EfficientDet)

El EfficientDet, ha sido diseñado principalmente para ofrecer un equilibrio entre la precisión y la eficiencia computacional, lo que los hace adecuados para el uso en plataformas móviles y de recursos limitados como drones y dispositivos Edge (Shahi et al., 2022).

3.10.5 Integración IoT en Monitoreo Agrícola

La integración de sensores IoT, redes de comunicación y análisis de datos forma la base de los sistemas de monitoreo agrícola. Ya que estos sistemas recopilan información continuamente del campo, permitiendo automatizar decisiones, y aparte son capaces de mejorar la eficiencia en el uso de recursos, reducir pérdidas y brindar soporte para acciones predictivas en agricultura inteligente (Pathmudi et al., 2023)

3.11 Metodología CRISP-DM

La metodología CRISP-DM organiza los proyectos de ciencia de datos en 6 fases claras que se repiten hasta lograr un sistema que funcione bien en el campo real. Es el método más usado en el mundo para no saltarse pasos importantes y entregar resultados que realmente sirvan al negocio (Chapman et al., 2000), como se muestra en la Figura 4, Metodología CRISP-DM.

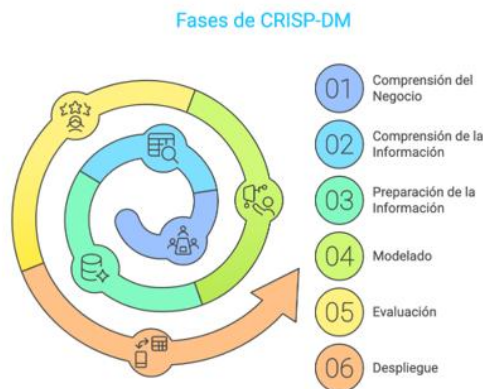


Figura 4 Metodología CRISP-DM.

3.11.1 Comprensión del negocio

La comprensión del negocio es la etapa inicial, la cual se centra en identificar y clarificar los objetivos del proyecto. Se establece qué se desea hacer y lograr, cuáles son las restricciones que tiene el problema, cómo se puede medir el éxito y qué conocimiento se espera obtener de los datos. Proporciona el marco para todas las decisiones que se tomarán a lo largo del proyecto y ayuda a orientar la selección de técnicas y métricas apropiadas para el análisis (Chapman et al., 2000).

3.11.2 Comprensión de los datos

Se aborda la recolección inicial de datos y se busca un análisis de estos para entender su estructura, contenido y calidad. Se examinan variables, se identifican los valores faltantes o inconsistencias, se detectan patrones iniciales y se evalúa si los datos disponibles son adecuados para responder las preguntas previamente planteadas. Esta etapa es fundamental para evitar problemas posteriores por falta de calidad o representatividad de los datos (Chapman et al., 2000).

3.11.3 Preparación de los datos

Se elabora la preparación de datos; esto incluye actividades de limpieza, transformación, integración y selección de características para poder generar conjuntos de datos adecuados para el modelo. Puede involucrar normalización de valores, codificación de variables

categorías, eliminación de duplicados, manejo de valores ausentes y reducción de dimensiones (Chapman et al., 2000).

3.11.4 Modelado

Se seleccionan y aplica los algoritmos de aprendizaje automático o métodos estadísticos más apropiados para resolver el problema. Implica entrenar el modelo con los datos preparados, ajustar hiperparámetros, evaluar diferentes técnicas y comparar los resultados. La elección de la técnica de modelado se basa principalmente en los objetivos del proyecto y las características que presentan los datos (Chapman et al., 2000).

3.11.5 Evaluación

Se verifica si el modelo o los modelos entrenados cumplen con los objetivos planteados y son generalizables a datos nuevos. Se utilizan algunas métricas cuantitativas y métodos de validación para calificar el desempeño del modelo. De igual manera, se revisa si los resultados tienen sentido desde la perspectiva del negocio o del dominio de la aplicación, asegurando que las conclusiones puedan ser interpretadas y utilizadas por el usuario final (Chapman et al., 2000).

3.11.6 Despliegue

Implica integrar el modelo o la solución desarrollada en un entorno operativo real, donde pueda ser utilizado por los sistemas y usuarios finales. Esto implica la creación de servicios API, la generación de reportes automatizados, la integración con dashboards o aplicaciones web (Chapman et al., 2000).

4. PLANTEAMIENTO DEL PROBLEMA

En el Estado de México, el cultivo del agave mezcalero se extiende aproximadamente 970-984 hectáreas en 15 municipios del estado, así constituyendo un sector estratégico que sustenta la industria regional de destilados tradicionales, siendo el gobierno estatal quien impulsa viveros para conservar esta superficie cultivable ante el riesgo de extinción local de variedades nativas (Gobierno del Estado de México, 2021; Secretaria del campo, 2023). Esta importancia productiva hace crítico contar con herramientas de monitoreo precisas que optimicen la gestión de dichas hectáreas.

El cultivo de agave y el monitoreo eficiente representa un desafío importante para los productores, ya que de esto depende decisiones sumamente importantes como la programación de la jima, la estimación de los rendimientos y la planeación de los procesos industriales. Algunos métodos tradicionales que se basan en recorridos de campo manualmente implican un valor económico elevado, ya que la mano de obra puede representar hasta el 75% de los costos totales del cultivo (Gutiérrez Vaca et al., 2013). Además, demandan una cantidad considerable de tiempo humano y son propios de un esfuerzo desgastante, especialmente en parcelas que abarcan decenas o centenares de hectáreas. Esta situación, que a menudo presenta márgenes de error, reduce la efectividad del monitoreo y contribuye a una planificación ineficiente de la cosecha, lo que puede derivar en pérdidas económicas para los productores (Ponce Corona, 2020).

En diversas investigaciones realizadas por diferentes autores han mostrado el potencial que puede llegar a tener el uso de la inteligencia artificial y drones para un eficiente conteo y clasificación del agave y otros cultivos. Pese a ello, la mayoría de las investigaciones muestran soluciones que se basan en el análisis de imágenes estáticas aéreas, lo que ha limitado su aplicación en escenarios donde se requiere resultados rápidos e inmediatos.

Es así, que el problema de investigación persiste en la ausencia de sistemas tecnológicos que sean accesibles y funcionales en condiciones reales que permitan la detección, el conteo y la clasificación de madurez del agave en tiempo real a través de video emitido desde drones

DJI, con esta ausencia se limita la disponibilidad de la información confiable y actualizada para la gestión eficiente del cultivo.

Sin la presencia de este tipo de sistemas se cree que ha impactado de manera negativa en el ámbito productivo, ya que limita la optimización de los recursos y la planeación de la cosecha, de la misma manera en el ámbito científico, al evidenciar la necesidad urgente de modelos computacionales eficientes que operen bajo condiciones de campo reales.

Tras analizar limitaciones encontradas se identificó una posible solución y oportunidad en el desarrollo de un sistema basado en visión por computadora, poda procesar video en tiempo real a través de drones que puedan detectar, contar y clasificar el estado de madurez del agave.

5. JUSTIFICACIÓN

Para el cultivo del agave, el monitoreo preciso y eficiente es un elemento clave e importante para mejorar la productividad y la sostenibilidad del sector agrícola. En el Estado de México y en la mayoría del país se utilizan métodos manuales para el conteo y clasificación de madurez, métodos que muchas veces limitan la precisión de los resultados; como consecuencia, afecta la planeación de la cosecha y la estimación de rendimientos, consecuencias que muchas veces generan pérdidas de tiempo y de recursos (Pérez et al., 2017).

Ante la necesidad de responder a la necesidad que se menciona, se planteó desarrollar un sistema de monitoreo mediante drones DJI y visión por computadora; se pretende automatizar la detección de plantas de agave y clasificar el estado de madurez de esta, buscando mejorar la eficiencia en el manejo del cultivo. Asimismo, con el uso del procesamiento en tiempo real, poder posibilitar la obtención de información actualizada que impacte positivamente en la planeación de la jima, la estimación de rendimientos y la programación de los procesos industriales.

Los principales beneficiarios serán los productores, los técnicos agrícolas y las empresas dedicadas al procesamiento del agave. Ya que se podrán emplear los resultados obtenidos del sistema para poder planificar actividades y tiempos de manera más precisa, siendo más rentable en comparación con los métodos manuales tradicionales.

Con el desarrollo de la investigación y la creación del sistema se busca beneficiar instituciones académicas y de investigación, ya que se tendrá a disposición una herramienta útil para el análisis y el desarrollo de nuevas aplicaciones dentro de la agricultura de precisión.

En una perspectiva económica y social, la aplicación de este tipo de sistema puede contribuir a la reducción de costos operativos, al poder optimizar el uso de recursos humanos y materiales, así como fortalecer la competitividad del sector agrícola, el cual es una fuente importante de empleo e identidad cultural en el Estado de México y otras regiones.

Asimismo, el uso de tecnologías de monitoreo automatizado favorece prácticas agrícolas más sostenibles, al permitir una mejor gestión del cultivo, reducir recorridos innecesarios en campo y optimizar la aplicación de insumos, contribuyendo indirectamente a la protección del medio ambiente.

El proyecto busca contribuir a un avance tecnológico en el sector agroindustrial, al estar empleando sistemas de visión por computadora, inteligencia artificial (IA) y vehículos aéreos no tripulados en un contexto real. La implementación de estos busca promover la adopción de prácticas innovadoras que fortalezcan la competitividad en el sector agavero.

La investigación es factible debido a la disponibilidad de herramientas tecnológicas accesibles, como drones comerciales, modelos de aprendizaje profundo preentrenados y plataformas de desarrollo web, así como al acceso a datos visuales del cultivo. Estas condiciones permiten la implementación del sistema dentro de los recursos y el tiempo establecidos para el desarrollo del proyecto.

La investigación propone una solución práctica y replicable ante un problema real que afecta a todo un sector de importancia económica y cultural en México, ya que es la base de bebidas fermentadas y otros derivados. Se prevé que los resultados que sean obtenidos impulsen la modernización del manejo agrícola del agave y por consecuencia, sienten las bases para futuras aplicaciones de monitoreo automatizado en diferentes tipos de cultivos.

Se busca fortalecer la relación entre la tecnología y el campo, mostrando que la inteligencia artificial puede aplicarse de formas prácticas y accesibles para resolver problemas agrícolas reales. Contribuyendo a la innovación en el sector primario, buscando una agricultura más precisa, eficiente y sustentable, capaz de responder a las demandas actuales y futuras del mercado.

6. OBJETIVOS

6.1 Objetivo general

Desarrollar un sistema basado en redes neuronales convolucionales y vehículos aéreos no tripulados para la detección, conteo y clasificación de la planta de agave, con el fin de poder optimizar el monitoreo agrícola, mejorar la planificación productiva y así contribuir a la conservación sostenible del cultivo en zonas rurales.

6.2 Objetivos específicos

- Tomar imágenes aéreas de cultivos de agave a través de un dron para conseguir información visual representativa del cultivo.
- Etiquetar y organizar cada imagen recolectada, para que se pueda determinar qué planta es y categorizarla de acuerdo con su estado de madurez.
- Desarrollar y ajustar un modelo para detectar objetos, con el propósito de identificar y contabilizar las plantas de agave automáticamente.
- Poner en funcionamiento un módulo de clasificación que establezca el grado de madurez de cada planta identificada.
- Verificar el rendimiento del sistema a través de indicadores de eficiencia de conteo, a través de las métricas Precisión y Recall, Accuracy, f1-score.
- Realizar una comparación de los resultados alcanzados con los métodos manuales tradicionales para medir su eficacia.
- Elaborar informes y representaciones gráficas que hagan más sencilla la interpretación de los resultados para los productores.

7. HIPÓTESIS

Como hipótesis de investigación nos planteamos la siguiente pregunta:

¿Es viable y eficiente un sistema de monitoreo en tiempo real del cultivo de agave mediante drones y visión por computadora, capaz de detectar plantas y clasificar su madurez con precisión suficiente para apoyar la toma de decisiones agrícolas?

8. METODOLOGÍA

La metodología empleada en el presente proyecto fue CRISP-DM; se utilizó debido a que esta metodología es un estándar ampliamente utilizado en proyectos de ciencia de datos, visión por computadora e inteligencia artificial; ejemplo de ello son los siguientes proyectos: Modelo predictivo de diabetes con machine learning, detección de granos de maíz anómalos (visión + deep learning). Principalmente usada porque proporciona una estructura clara, ordenada y cíclica para el desarrollo de modelos que están basados en datos.

De igual manera, permite alinear los objetivos con las necesidades reales del negocio; de esta manera asegura que el modelo a desarrollar no solo funcione desde el punto de vista computacional, sino que también sea útil para la planeación y la toma de decisiones. Además, CRISP-DM facilita la documentación del proyecto, el análisis de resultados y la mejora continua del sistema.

CRISP-DM funciona mediante una serie de fases interconectadas, donde cada etapa influye en la siguiente, permitiendo regresar a fases anteriores para realizar ajustes cuando sea necesario. En este proyecto, la metodología fue adaptada para cubrir desde la definición de los objetivos hasta la implementación del sistema y su mantenimiento, asegurando un desarrollo completo y confiable, como se describe en la Figura 5.



Figura 5 Metodología CRISP-DM aplicada a mi proyecto. (Elaboración propia).

A continuación, se describen las fases de la metodología CRISP-DM aplicadas al proyecto:

8.1 Comprensión del negocio

En la primera fase, se definió con claridad el problema principal que se iba a resolver y su impacto que tendría en el sector agrícola, específicamente en la optimización de los procesos que están relacionados con el monitoreo y la planeación de la cosecha del cultivo del agave.

Se definieron las métricas de confiabilidad con las cuales se iba a evaluar el desempeño para poder establecer criterios medibles para el éxito del sistema durante las fases del proyecto.



Figura 6 Agricultor monitoreando un cultivo de agave. (EFE, 2025).

8.1.1 Objetivos del proyecto

Los objetivos que se plantearon responden a una necesidad real del sector agrícola, donde los métodos tradicionales de conteo y evaluación de madurez suelen ser manuales, costosos y poco estables. El uso de visión por computadora y redes neuronales convolucionales permite procesar grandes volúmenes de información visual de forma automática y consistente.

Objetivo	Descripción	Métrica de evaluación	Valor mínimo
Detección de agave	Identificar automáticamente plantas de agave en imagen y video	Precisión	$\geq 70\%$

Conteo de plantas	Calcular el número total de plantas por área	Error de conteo	$\leq 30\%$
Clasificación de madurez	Clasificar plantas según su etapa de madurez	Exactitud	$\geq 60\%$
Generación de reportes	Mostrar resultados visuales y estadísticos	Disponibilidad del reporte	Correcta

Tabla 1 Objetivos del proyecto. (Elaboración propia).

El establecimiento de un umbral mínimo del 70% de precisión garantiza que el sistema sea funcional y útil en un entorno real, sirviendo como base para futuras mejoras y ajustes del modelo.

8.1.2 Limitaciones del entorno

La identificación temprana de las limitaciones que tiene el sistema permitió diseñar estrategias técnicas para mitigarlas durante la preparación y modelado de los datos. Factores como la iluminación variable y el solapamiento de plantas son muy comunes en escenarios reales; por ende, el sistema fue concebido para operar bajo condiciones medianamente controladas.

Limitación	Descripción	Impacto en el sistema	Estrategia aplicada
Iluminación variable	Cambio de luz natural en el campo	Detección imprecisa del sistema	Aumento de datos, para variar el tipo de iluminación de la imagen
Plantas solapadas	Superposición parcial de agaves	Conteo incorrecto, ya sea mayor o menor conteo	Crear un modelo más robusto, con tracking IoU

Baja resolución	Imágenes borrosas	Perdida de detalles de la planta	Normalización de las imágenes
-----------------	-------------------	----------------------------------	-------------------------------

Tabla 2 Limitaciones del entorno. (Elaboración propia).

8.2 Comprensión de los datos

El análisis inicial de los datos fue fundamental para conocer la diversidad visual del agave y su comportamiento en diferentes contextos. La variabilidad en ángulos, distancias e iluminación permitió entrenar un modelo más generalizable.



Figura 7 Tipo de iluminación. (Elaboración propia).

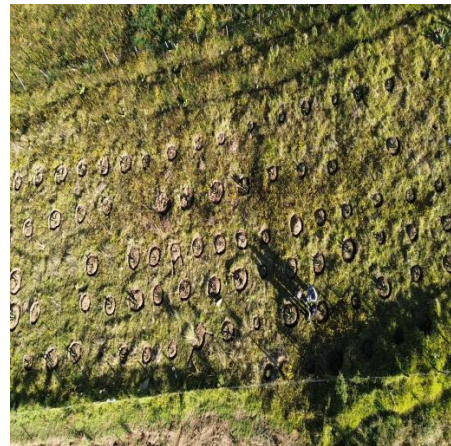


Figura 8 Tipo de distancias. (Elaboración propia).

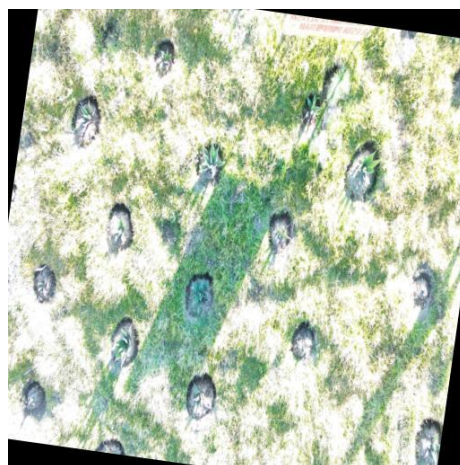


Figura 9 Tipo de rotación. (Elaboración propia).

Las estadísticas descriptivas facilitaron la detección de posibles sesgos en el conjunto de datos, así como la identificación de imágenes que requerían limpieza o corrección antes del entrenamiento.

8.3 Preparación de los datos

La fase de preparación fue clave para mejorar el desempeño del modelo. La normalización de imágenes permitió un procesamiento homogéneo, mientras que el aumento de datos ayudó a reducir el sobreajuste y mejorar la capacidad de generalización del modelo ante escenarios no vistos. La correcta división de los conjuntos de datos aseguró una evaluación objetiva y confiable del desempeño del sistema.



Figura 10 Preparación de los datos (Zoom y traslación). (Elaboración propia).



Figura 11 Preparación de los datos (Ajuste de brillo). (Elaboración propia).



Figura 12 Preparación de los datos (Rotación). (Elaboración propia).

Técnica	Descripción	Propósito
Normalización	Ajuste de tamaño y escala	Homogenizar los datos
Rotación	Giro de la imagen	Simular ángulos
Ajuste de brillo	Variación de la iluminación	Robustez
Zoom y traslación	Cambio de escala y posición	Generalización

Tabla 3 Técnicas de preparación de los datos. (Elaboración propia).

8.4 Modelado

El diseño del modelo se realizó considerando la complejidad del entorno real, ya que existen variaciones significativas en tamaño, forma y disposición. La selección que se hizo en la red neuronal convolucional (CNN), la cual fue YOLOv5, permitió una detección eficiente en tiempo real, cumpliendo así con los requerimientos del proyecto.

Así mismo, el uso de métricas como precisión y recall permitió evaluar el desempeño y el equilibrio entre detecciones correctas y errores, asegurando así un modelo más confiable tanto para el conteo como para la clasificación de madurez.

8.5 Evaluación

En la evaluación se validó el desempeño que tuvo el modelo en imágenes utilizadas durante el entrenamiento, se documentaron los casos de error, su análisis y las posibles mejoras que podía tener, cumpliendo así los objetivos que se definieron en la primera fase, que fue la comprensión del negocio. La evaluación confirmó que el sistema cumple con los objetivos definidos en la fase de comprensión del negocio.

Se evaluaron dos modelos diferentes y estos fueron los resultados:

Métrica	Descripción	Resultado prueba 1	Resultado prueba 2
Precisión	Detecciones correctas	0.522	0.949
Recall	Objetos detectados	0.71	0.97
mAP50	Precisión media con cajas que se superponen hasta 50%	0.599	0.985
mAP50-95	Precisión media con cajas muy precisas (10-95% superposición)	0.366	0.697

Tabla 4 Evaluación del sistema con métricas. (Elaboración propia).

8.6 Implementación

La implementación mediante Streamlit permitió desarrollar una interfaz intuitiva y accesible, facilitando la interacción del usuario con el sistema sin necesidad de conocimientos técnicos avanzados. La integración de procesamiento en tiempo real amplía las posibilidades de uso del sistema en campo. El registro automático de resultados y métricas contribuye a la trazabilidad del sistema y a la generación de evidencia para análisis posteriores.

Flujo de interacción de la interfaz:

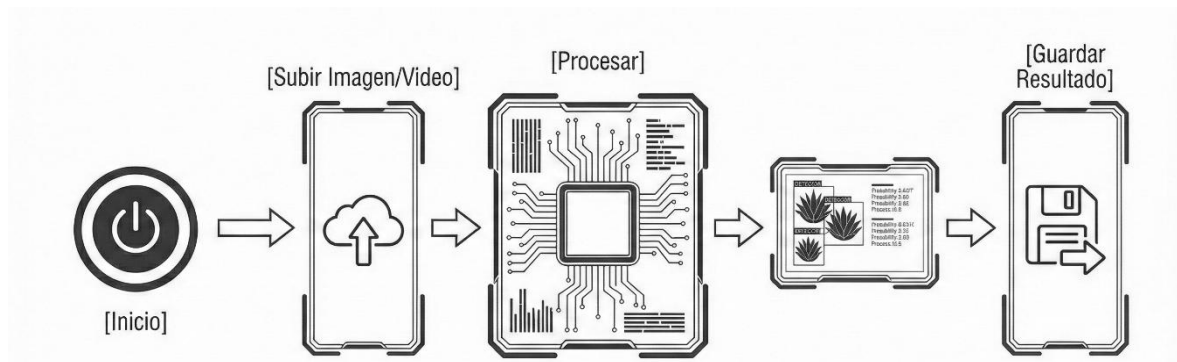


Figura 13 Flujo de interacción de la interfaz. (Elaboración propia).

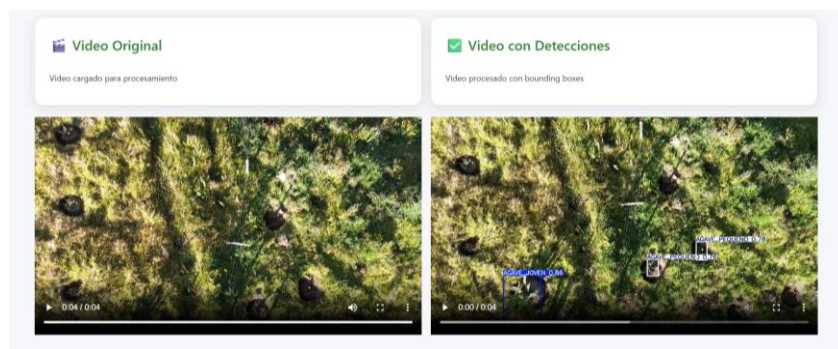


Figura 14 Pagina web con resultados (Elaboración propia).

8.7 Despliegue y mantenimiento

El despliegue en un entorno de prueba permitió validar el funcionamiento del sistema en condiciones similares a las reales. La retroalimentación obtenida a partir de los resultados reales es fundamental para la mejora continua del modelo.

El mantenimiento considera la actualización periódica del modelo y el ajuste de hiperparámetros para conservar un desempeño óptimo conforme se incorporen nuevos datos.

8.8 Resultados

La adaptación de la metodología CRISP-DM permitió estructurar el proyecto de manera ordenada y sistemática, asegurando la trazabilidad entre los objetivos, los datos y los resultados obtenidos. Esta metodología facilitó el desarrollo de un sistema eficiente, escalable y documentado.

Los resultados obtenidos demuestran que el enfoque propuesto es viable para aplicaciones agrícolas inteligentes, sentando las bases para futuras mejoras y ampliaciones del sistema.



Figura 15 Resultados de la detección en video. (Elaboración propia).

9. RESULTADOS Y DISCUSIÓN

9.1 Análisis del sistema

9.1.1 Identificación del problema

Como primer paso, se realizó un análisis del contexto del cultivo del agave, donde se descubrió que el conteo manual de plantas y la evaluación visual de madurez requerían demasiado tiempo y personas. Estos procesos requerían recorridos extensos dentro de los cultivos, lo que limitaba la frecuencia de monitoreo y generaba inconsistencias por factores humanos. Se identificó que la evaluación que se realiza a la madurez del agave no sigue criterios completamente estandarizados, lo que realmente genera inconsistencias en los registros obtenidos. Esta situación limita la frecuencia con la que se puede realizar el monitoreo del cultivo y así dificulta la planeación eficiente de la cosecha, el control del crecimiento de la planta y la toma de decisiones estratégicas.



Figura 16 Agricultor monitoreando un cultivo de agave. (EFE, 2025).

Ante la problemática, se identificó la necesidad de desarrollar un sistema automatizado capaz de detectar, contar y clasificar plantas de agave de manera confiable utilizando técnicas de visión por computadora e inteligencia artificial, apoyada por imágenes y videos aéreos capturados a través de un dron.



Figura 17 Monitoreo y riego de agave con Dron DJI (Xinhua, 2024).

9.1.2 Análisis de requerimientos del sistema

Una vez que se identificó el problema, se definieron los requerimientos funcionales y técnicos que el sistema debía llevar. Los requerimientos principales incluyeron el procesamiento de imágenes aéreas estáticas, videos pregrabados y transmisiones en tiempo real, considerando diferentes resoluciones, ángulos de captura, altitudes del dron y tamaños de parcelas. Así el sistema debía mostrar resultados visuales claros, donde cada planta de agave se debía ver delimitada por un rectángulo y se clasificara de acuerdo con su estado de madurez. Además, se definió como prioridad la necesidad de generar conteos automáticos por categoría (germinación, juvenil y maduro), así como un conteo total de plantas por área analizada.

ID	Requerimiento	Descripción
RF-01	Procesar imágenes	Imágenes aéreas de cultivos de agave
RF-02	Procesar videos	Videos grabados desde el dron
RF-03	Video en tiempo real	Transmisión de video directa del dron
RF-04	Clasificación	Por madurez (pequeño, joven, maduro)
RF-05	Conteo	Conteo total de plantas de agave y por clase

Tabla 5 Requerimientos funcionales del sistema. (Elaboración propia).

En cuanto a los requerimientos técnicos, se estableció la integración de un modelo de red neuronal convolucional especializada en detección de objetos, específicamente un modelo basado en la arquitectura de YOLOv5. De igual manera, se definió la medición de desempeños del modelo mediante métricas estándar como precisión, recall y mAP, con el fin de evaluar de forma objetiva la calidad de las detecciones.

ID	Requerimientos	Descripción
RT-01	Modelo IA	Red neuronal Convolucional (CNN) YOLOv5
RT-02	Plataforma	Web
RT-03	Métricas	Precisión, Recall, mAP
RT-04	Visualización	Bounding boxes

Tabla 6 Requerimientos técnicos del sistema. (Elaboración propia).

Se eligió YOLOv5 por su alta precisión y velocidad en detección de objetos, lo que lo hace ideal para entrenar modelos en Google Colab. Esta arquitectura ha demostrado ser efectiva en proyectos reales, como la detección de enfermedades en hojas de arroz y detección de drones, garantizando resultados confiables y eficientes.

Se utilizaron precisión, recall y mAP porque son métricas estándar en detección de objetos y permiten evaluar tanto la exactitud de las predicciones como la capacidad del modelo para detectar todas las instancias relevantes. Estas métricas proporcionan una visión completa del desempeño, a diferencia de otras métricas generales como accuracy, que no reflejan adecuadamente errores por falsos positivos o negativos en tareas de detección.

9.2 Diseño del sistema

9.2.1 Diseño de la arquitectura del sistema

Se diseñó un tipo de arquitectura modular con el objetivo de facilitar su desarrollo, mantenimiento y escalabilidad. Esta arquitectura se compuso de cuatro módulos principales:

- Adquisición de datos visuales, encargada de la captura de imágenes y videos mediante dron.

- Procesamiento y análisis con inteligencia artificial, donde se integra el modelo de red neuronal entrenado.
- Visualización de resultados, responsable de mostrar las detecciones, clasificaciones y conteos.
- Interfaz de usuario, desarrollada para permitir la interacción con el sistema de forma intuitiva.

De esta manera se permitió desarrollar y probar cada componente de manera individual antes de integrarlo. Cada componente cumplió funciones específicas dentro del flujo completo del sistema.

9.2.2 Diseño del flujo de procesamiento

Se establecieron cinco etapas secuenciales para el procesamiento:

- Carga de datos visuales (imagen, video o transmisión de datos)
- Preprocesamiento de imágenes
- Análisis mediante el modelo CNN (YOLOv5)
- Organización de resultados
- Generación de visualizaciones.

Se definieron parámetros técnicos como umbrales de confianza mínimos para las detecciones, el tamaño de imagen de entrada y criterios de clasificación de madurez.

9.3 Desarrollo del sistema

9.3.1 Recolección de imágenes mediante dron

Se realiza la captura de imágenes y videos aéreos en cultivos de agave utilizando el dron. Las tomas se efectuaron desde diferentes altitudes y ángulos de visión, con el propósito de obtener variabilidad en el tamaño aparente de las plantas y en su orientación dentro de las imágenes. La recolección se realizó en diferentes cultivos en condiciones diurnas, cubriendo así áreas con gran densidad de la planta. Este material compone el conjunto de datos para el entrenamiento del modelo de inteligencia artificial.



Figura 18 Imagen aérea recolectada para el conjunto de datos. (Elaboración propia).



Figura 19 Imagen aérea recolectada para el conjunto de datos. (Elaboración propia).



Figura 20 Imagen aérea recolectada para el conjunto de datos. (Elaboración propia).

9.3.2 Preparación y etiquetado del conjunto de datos

Una vez que se concluyó la etapa de recolección de imágenes aéreas, seguimos con la preparación y etiquetado del conjunto de datos, el cual constituye uno de los elementos más importantes para el correcto funcionamiento del modelo de inteligencia artificial. La calidad que contiene el etiquetado influye directamente en la capacidad del modelo para aprender patrones visuales representativos del cultivo de agave y diferenciar correctamente los distintos estados de madurez.

Se etiquetaron dos conjuntos de datos, ya que se utilizaron clases diferentes en cada uno, esto con el fin de observar cuál tenía un mejor comportamiento.

El conjunto de datos consta de 1611 imágenes.

9.3.2.1 Selección y limpieza de imágenes

Después, se realizó una revisión manual de todas las imágenes recolectadas con el dron. En esta etapa se eliminaron aquellas imágenes que presentaban problemas significativos, tales como desenfoque excesivo, sobreexposición, sombras extremas, presencia de objetos no relevantes o ángulos de captura que dificultaran la identificación clara de las plantas. Este proceso permitió asegurar que únicamente imágenes con calidad visual adecuada formaran parte del conjunto de datos.

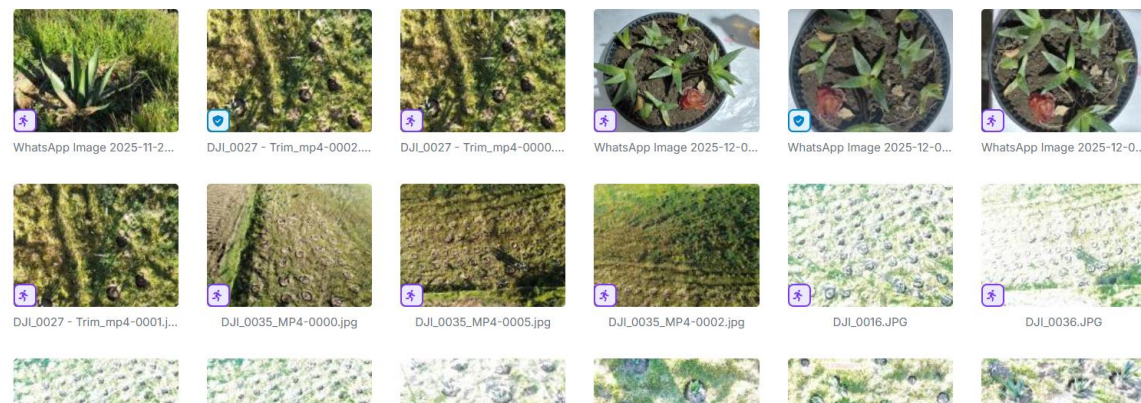


Figura 21 Selección y limpieza de imágenes (Primer conjunto de datos). (Elaboración propia).

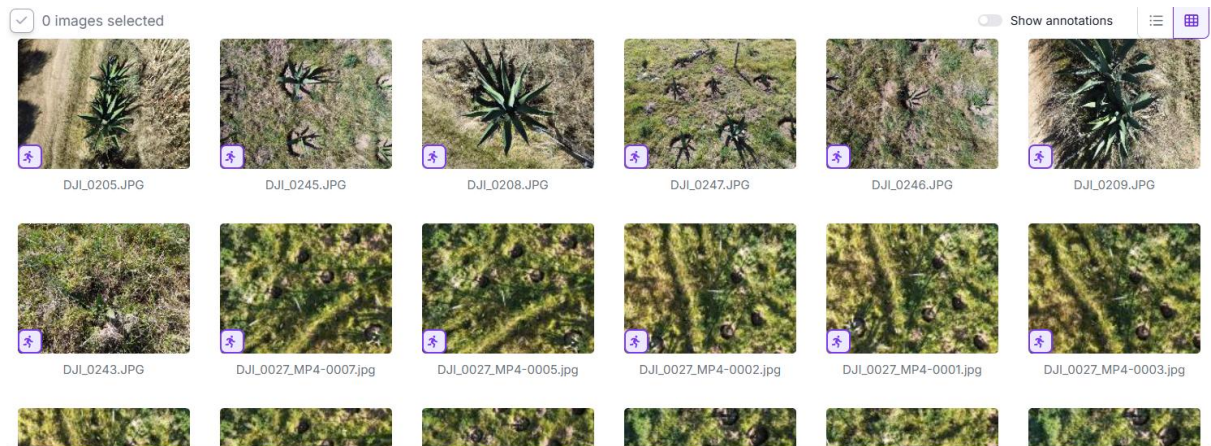


Figura 22 Selección y limpieza de imágenes (Segundo conjunto de datos). (Elaboración propia).

9.3.2.2 Proceso de etiquetado en Roboflow

El etiquetado de las imágenes se realizó utilizando la plataforma Roboflow, la cual facilita la anotación manual de objetos y la gestión de conjuntos de datos para proyectos de visión por computadora. Cada imagen fue cargada en el entorno de etiquetado y analizada individualmente.



Figura 23 Proceso de etiquetado en Roboflow, (Primer conjunto de datos). (Elaboración propia).

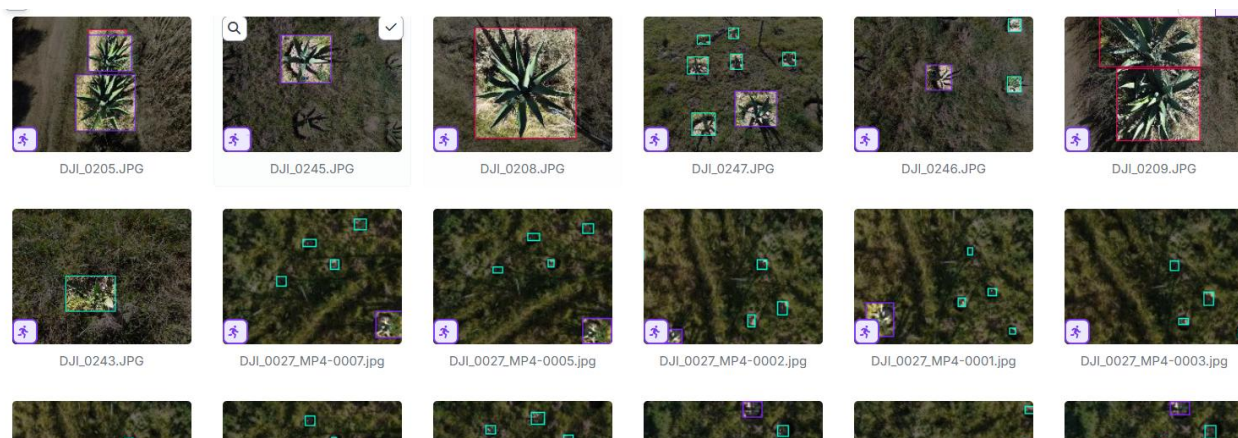


Figura 24 Proceso de etiquetado en Roboflow, (Segundo conjunto de datos). (Elaboración propia).

Para cada planta de agave visible en la imagen, se dibujó un rectángulo delimitador (bounding box), asegurando que el contorno cubriera completamente la planta sin incluir elementos externos como suelo, sombras o vegetación ajena. A cada rectángulo fue asignado a una de las siguientes categorías previamente definidas:

Clase	Descripción
Agave	Se utilizó como una categoría general de referencia para representar plantas de agave sin especificar una etapa de madurez particular. Esta clase permitió reforzar el reconocimiento global de la planta y facilitar la detección en escenarios donde la clasificación precisa de madurez no era clara. Esta clase solo se utilizó en el primer conjunto de datos.
Germinación o Pequeño	En la clase, se incluyeron plantas en etapas tempranas de crecimiento, caracterizadas por un tamaño reducido, menor número de hojas visibles y una disposición compacta cercana al suelo. Estas plantas presentan una forma menos definida y un color generalmente más claro, lo que las hace visualmente distintas del resto de las etapas, aunque más difíciles de detectar debido a su pequeño tamaño. Se utiliza en ambos conjuntos de datos.
Juvenil o joven	La clase corresponde a plantas en una etapa intermedia de desarrollo. En esta fase, las plantas muestran un incremento notable en el tamaño y

	número de hojas, con una estructura más abierta y definida. Sin embargo, visualmente pueden presentar similitudes tanto con plantas en germinación avanzada como con plantas maduras jóvenes, lo que representa un mayor reto para el modelo durante el proceso de aprendizaje. Se utiliza en ambos conjuntos de datos.
Maduro	La clase MADURO agrupa plantas completamente desarrolladas, caracterizadas por un mayor diámetro, hojas largas y bien definidas, así como una estructura radial claramente visible desde la vista aérea. Estas plantas suelen ser más fáciles de identificar debido a su tamaño y forma distintiva, lo que se refleja en un mejor desempeño del modelo para esta categoría. Se utiliza en ambos conjuntos de datos.

Tabla 7 Clases del etiquetado. (Elaboración propia).

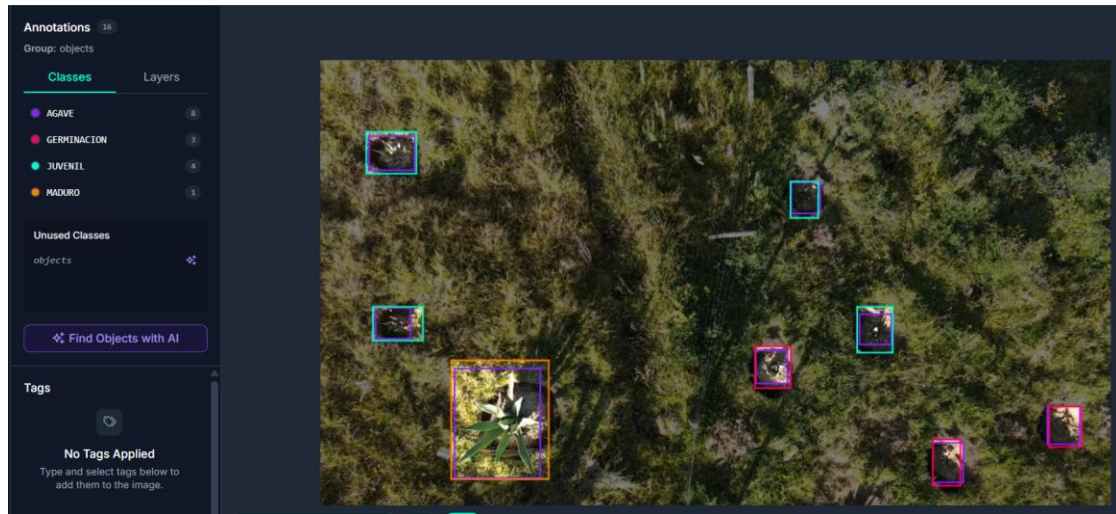


Figura 25 Etiquetado de las 4 clases del primer conjunto de datos. (Elaboración propia).

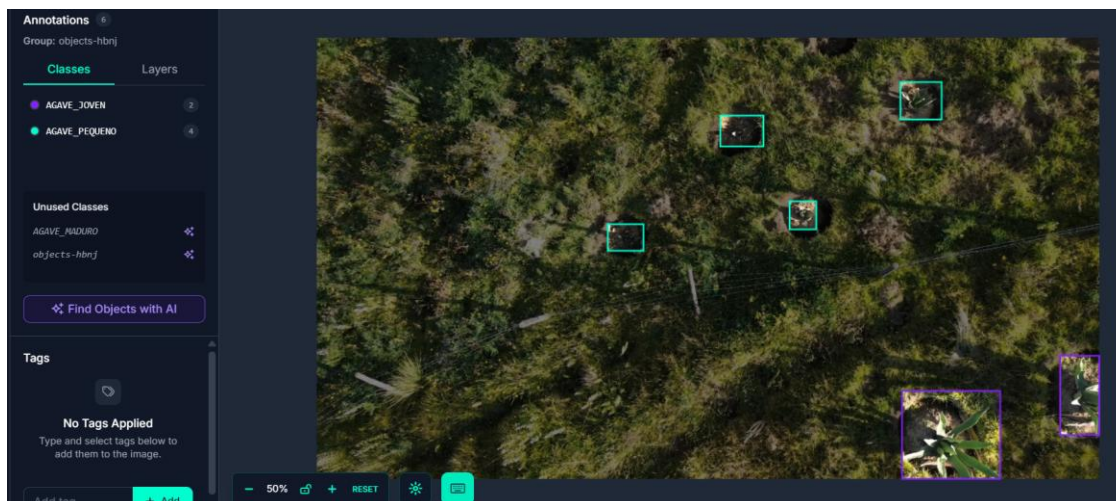


Figura 26 Etiquetado de las 3 clases del segundo conjunto de datos. (Elaboración propia).

Este proceso se repitió de manera exhaustiva para cada imagen, garantizando consistencia en los criterios de clasificación y evitando ambigüedades entre clases similares.

9.3.2.3 Aumento de datos y preprocesamiento automático

Durante la preparación del conjunto de datos, la plataforma de Roboflow permitió aplicar técnicas para el aumento de datos con el objetivo de poder mejorar la capacidad de generalización del modelo. Entre las transformaciones utilizadas se incluyeron:

- Rotaciones leves de las imágenes.
- Ajustes de brillo y contraste.
- Escalado y recorte controlado.
- Inversiones horizontales.

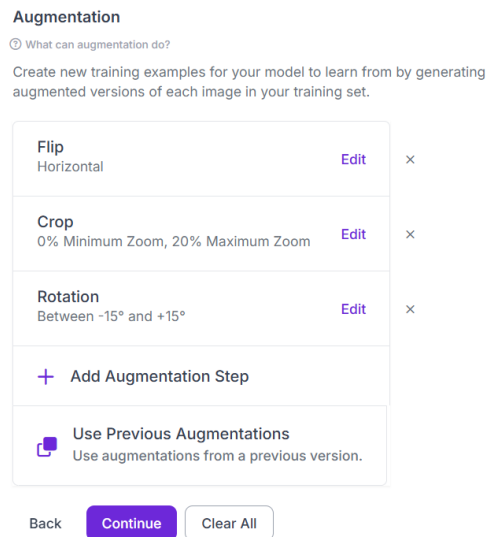


Figura 27 Aumento de datos. (Elaboración propia).

Este paso se realizó en los dos conjuntos de datos.

Estas transformaciones simulan variaciones reales en las condiciones de captura, como cambios en la iluminación, orientación del dron y perspectiva, lo cual contribuye a que el modelo aprenda características más robustas y reduzca el sobreajuste.

9.3.2.4 División del conjunto de datos

El conjunto de datos se dividió en tres subconjuntos:

- Conjunto de entrenamiento fue un 70%, se utilizó para el aprendizaje del modelo.
- Conjunto de validación fue de 15%, fue empleado para ajustar los hiperparámetros durante el entrenamiento.
- Conjunto de prueba fue de 15%, fue destinado a evaluar el desempeño final del modelo.

La división 70% entrenamiento, 15% validación y 15% prueba se utilizó porque es un estándar comprobado en proyectos de visión por computadora y ha demostrado funcionar bien en modelos similares. Esta proporción ofrece suficiente información para que el modelo aprenda, permite ajustar hiperparámetros de forma estable y garantiza una evaluación final confiable. Además, proyectos recientes —como el de detección de enfermedades en hojas

publicado en Scientific Reports— emplean exactamente esta división con excelentes resultados, lo que respalda su uso en este trabajo (Chinnadurai & Selvakumar, 2025).

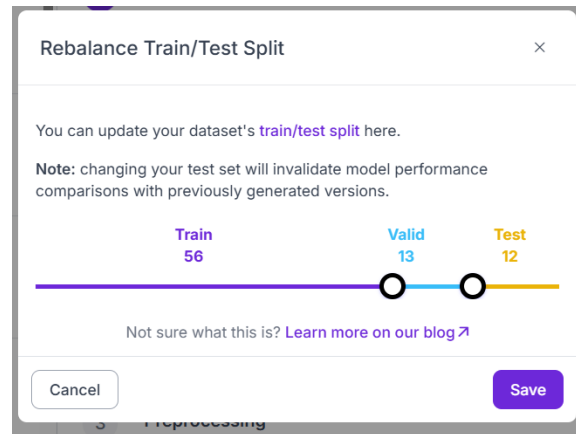


Figura 28 División del conjunto de datos. (Elaboración propia).

Este paso se realizó en los dos conjuntos de datos.

9.3.2.5 Preprocesamiento de imágenes

Antes de utilizar el conjunto de datos en el modelado, las imágenes fueron sometidas a un proceso de preprocesamiento automático durante el entrenamiento. Este preprocesamiento incluyó las siguientes etapas:

- Redimensionamiento: Todas las imágenes se ajustaron a un tamaño estándar de 640x640 píxeles, requerido por el modelo YOLOv5.
- Normalización de valores de píxeles: Los valores de los píxeles fueron escalados a un rango normalizado para facilitar el aprendizaje de la red neuronal.
- Conversión de formato: Las imágenes se procesaron en un formato compatible con el modelo, manteniendo la información de color necesaria para distinguir las plantas del entorno.
- Carga eficiente en lotes (batching): Las imágenes se cargaron en lotes de tamaño 16, optimizando el uso de recursos computacionales durante el entrenamiento.

Preprocessing

Decrease training time and increase performance by applying image transformations to all images in this dataset.

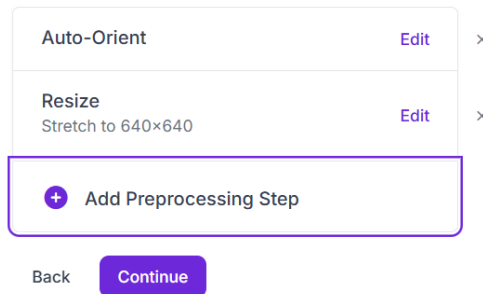


Figura 29 Preprocesamiento de imágenes. (Elaboración propia).

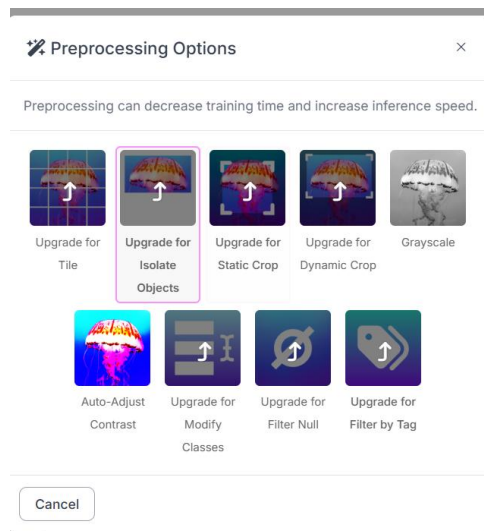


Figura 30 Preprocesamiento de imágenes. (Elaboración propia).

Este paso se realizó en los dos conjuntos de datos.

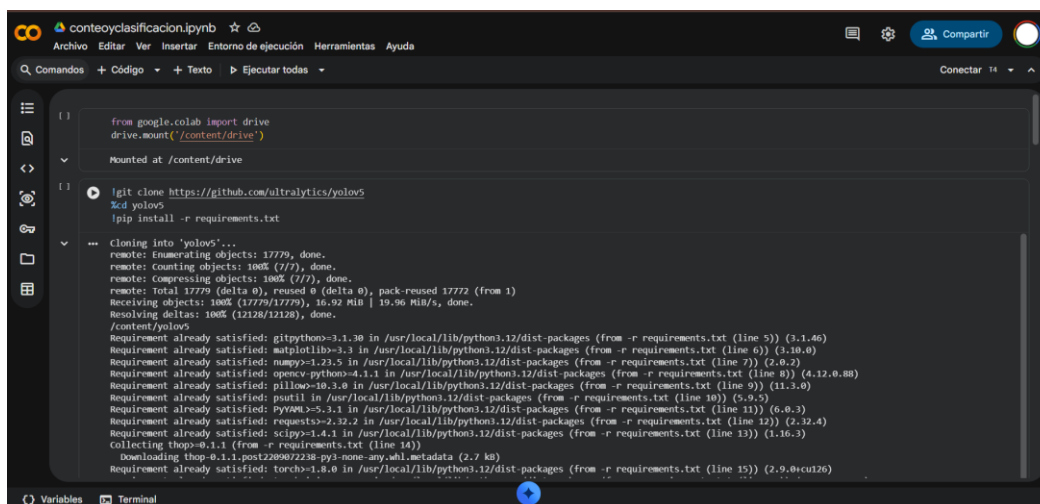
Este preprocesamiento permitió estandarizar la información de entrada y mejorar la estabilidad del proceso de entrenamiento.

9.3.3 Entrenamiento del modelo de red neuronal

El entrenamiento del modelo se realizó de manera progresiva y estructurada utilizando YOLOv5 en el entorno de Google Colab. Se eligió YOLOv5 como arquitectura para el entrenamiento del modelo porque es una de las técnicas de detección de objetos más utilizadas en visión por computadora moderna, conocida por su alta precisión, velocidad y

eficiencia en tareas de detección en tiempo real. A diferencia de otros métodos tradicionales (p. ej., Faster R-CNN o SSD), YOLOv5 combina detección rápida con una buena capacidad para localizar objetos en múltiples escalas, lo que lo hace adecuado para proyectos de clasificación y reconocimiento con datasets propios.

Se entrenaron los dos conjuntos de datos con el mismo entrenamiento.



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

Mounted at /content/drive

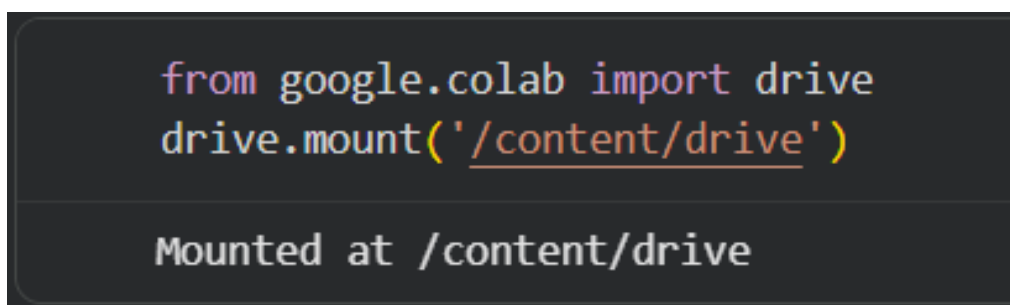
!git clone https://github.com/ultralytics/yolov5
!cd yolov5
!pip install -r requirements.txt

Cloning into 'yolov5'...
remote: Enumerating objects: 17779, done.
remote: Counting objects: 100% (1777/1777), done.
remote: Compressing objects: 100% (1777/1777), done.
remote: Total 17779 (delta 0), reused 0 (delta 0), pack-reused 17772 (from 1)
Receiving objects: 100% (17779/17779), 16.92 MiB | 19.96 MiB/s, done.
Resolving deltas: 100% (12128/12128), done.
/content/yolov5
Requirement already satisfied: gitpython>3.1.30 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 5)) (3.1.46)
Requirement already satisfied: matplotlib>3.3 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 6)) (3.10.0)
Requirement already satisfied: numpy>=1.23.5 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 7)) (2.0.2)
Requirement already satisfied: opencv-python>=4.1.1 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 8)) (4.12.0.88)
Requirement already satisfied: pillow>=10.3.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 9)) (11.3.0)
Requirement already satisfied: putils in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 10)) (5.9.5)
Requirement already satisfied: pyyaml>=5.3.1 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 11)) (6.0.3)
Requirement already satisfied: requests>=2.12.2 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 12)) (2.12.4)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 13)) (1.16.3)
Collecting thop>=0.1.1 (from -r requirements.txt (line 14))
  Downloading thop-0.1.1.post20200222.tar.gz (none-any.whl, metadata (2.7 kb))
Requirement already satisfied: torchvision in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 15)) (0.19.0+cu126)
```

Figura 31 Entorno de Google Colab. (Elaboración propia).

9.3.3.1 Configuración del entorno de entrenamiento

Como primer paso, se montó el almacenamiento de Google Drive para acceder al conjunto de datos etiquetado.



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

Mounted at /content/drive
```

Figura 32 Configuración del entorno de entrenamiento. (Elaboración propia).

Posteriormente, se clonó el repositorio oficial de YOLOv5 desde GitHub y se instalaron todas las dependencias necesarias para la ejecución del modelo.

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

Cloning into 'yolov5'...
remote: Enumerating objects: 17779, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 17779 (delta 0), reused 0 (delta 0), pack-reused 17772 (from 1)
Receiving objects: 100% (17779/17779), 16.92 MiB | 19.96 MiB/s, done.
Resolving deltas: 100% (12128/12128), done.
/content/yolov5
```

Figura 33 Clonación del repositorio de YOLOv5. (Elaboración propia).

9.3.3.2 Definición del archivo de configuración

Se creó un archivo de configuración en formato YAML donde se especificaron las rutas de los conjuntos de entrenamiento y validación, el número de clases y los nombres correspondientes a cada categoría. Este archivo permitió al modelo identificar correctamente la estructura del conjunto de datos.

```
dataset_path = '/content/drive/MyDrive/newversion2.0'

data_yaml = """
train: /content/drive/MyDrive/newversion2.0/train
val: /content/drive/MyDrive/newversion2.0/train

nc: 4
names: ['AGAVE', 'GERMINACION', 'JUVENIL', 'MADURO']
"""

with open('/content/yolo_agave3.yaml', 'w') as f:
    f.write(data_yaml)
```

Figura 34 Definición del archivo de configuración. (Elaboración propia).

9.3.3.3 Ejecución del entrenamiento

El proceso para el entrenamiento decidió llevarse a cabo durante 100 épocas, utilizando un tamaño de imagen de 640 píxeles y un tamaño de lote de 16 imágenes. Se emplearon pesos preentrenados (yolov5s.pt) con el objetivo de aprovechar el aprendizaje previo del modelo sobre características generales de imágenes, acelerando la convergencia y mejorando el desempeño final.

```

python train.py --img 640 --batch 16 --epochs 100 --data /content/yolo_agave3.yaml --weights yolov5s.pt --cache

*** Creating new Ultralytics Settings v0.0.6 file
View Ultralytics Settings with 'yolo settings' or at '/root/.config/ultralytics/settings.json'
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see https://docs.ultralytics.com/quickstart/ultralytics-settings.
wandb: WARNING ⚠ wandb is deprecated and will be removed in a future release. See supported integrations at https://github.com/ultralytics/yolov5#integrations.
2026-01-15 04:33:04.233212: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:467] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR

```

Figura 35 Ejecución del entrenamiento (Elaboración propia).

Durante el entrenamiento, el modelo ajustó automáticamente millones de parámetros internos, minimizando la función de pérdida y aprendiendo a identificar patrones visuales asociados a cada etapa de madurez del agave.

```

from n  params module arguments
0      -1 1      3520 models.common.Conv [3, 32, 6, 2, 2]
1      -1 1      18560 models.common.Conv [32, 64, 3, 2]
2      -1 1      18816 models.common.C3 [64, 64, 1]
3      -1 1      73984 models.common.Conv [64, 128, 3, 2]
4      -1 2     115712 models.common.C3 [128, 128, 2]
5      -1 1     295424 models.common.Conv [128, 256, 3, 2]
6      -1 3     625152 models.common.C3 [256, 256, 3]
7      -1 1     1180672 models.common.Conv [256, 512, 3, 2]
8      -1 1     1182720 models.common.C3 [512, 512, 1]
9      -1 1     656896 models.common.SPPF [512, 512, 5]
10     -1 1     131584 models.common.Conv [512, 256, 1, 1]
11     -1 1      0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
12     [-1, 6] 1      0 models.common.Concat [1]
13     -1 1     361984 models.common.C3 [512, 256, 1, False]
14     -1 1     33024 models.common.Conv [256, 128, 1, 1]
15     -1 1      0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
16     [-1, 4] 1      0 models.common.Concat [1]
17     -1 1     90880 models.common.C3 [256, 128, 1, False]
18     -1 1     147712 models.common.Conv [128, 128, 3, 2]
19     [-1, 14] 1      0 models.common.Concat [1]
20     -1 1     296448 models.common.C3 [256, 256, 1, False]
21     -1 1     590336 models.common.Conv [256, 256, 3, 2]
22     [-1, 10] 1      0 models.common.Concat [1]
23     -1 1     1182720 models.common.C3 [512, 512, 1, False]
24     [17, 20, 23] 1     24273 models.yolo.Detect [4, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], [128, 1
Model summary: 214 layers, 7030417 parameters, 7030417 gradients, 16.0 GFLOPs

```

Figura 36 Ajuste de parámetros. (Elaboración propia).

9.3.3.4 Validación durante el entrenamiento

Al finalizar cada época, el modelo fue evaluado con el conjunto de validación, permitiendo monitorear su desempeño y detectar posibles problemas como sobreajuste o subajuste. Este proceso permitió observar una mejora progresiva en las métricas de detección conforme avanzaba el entrenamiento.

```

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs

```

Class	Images	Instances	P	R	mAP50	mAP50-95
all	152	1112	0.522	0.71	0.599	0.366
AGAVE	152	844	0.71	0.892	0.829	0.436
GERMINACION	152	120	0.483	0.905	0.673	0.375
JUVENIL	152	136	0.387	0.875	0.58	0.391
MADURO	152	12	0.508	0.167	0.313	0.263

```

Results saved to runs/train/exp

```

Figura 37 Validación del modelo entrenado del primer conjunto de datos. (Elaboración propia).

```

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs

```

Class	Images	Instances	P	R	mAP50	mAP50-95
all	1611	4617	0.949	0.97	0.985	0.697
AGAVE_JOVEN	1611	1790	0.946	0.97	0.986	0.693
AGAVE_MADURO	1611	541	0.973	0.987	0.989	0.769
AGAVE_PEQUENO	1611	2286	0.928	0.952	0.98	0.628

```

Results saved to runs/train/exp

```

Figura 38 Validación del modelo entrenado del segundo conjunto de datos. (Elaboración propia).

Como se puede observar en la Figura 37 y la Figura 38 en segundo conjunto de datos tuvo un mejor desempeño, esto se debe a una mayor cantidad de imágenes y mayores instancias.

9.4 Desarrollo de la aplicación web

9.4.1 Implementación de la interfaz gráfica

El desarrollo de la aplicación web se realizó utilizando el framework Streamlit, debido a su capacidad para crear interfaces gráficas interactivas de manera eficiente y su compatibilidad directa con modelos de visión por computadora desarrollados en Python. Esta herramienta permitió integrar de forma sencilla el modelo de detección entrenado, así como visualizar los resultados de manera clara e intuitiva para el usuario.



Figura 39 Interfaz gráfica de la página web. (Elaboración propia).

La interfaz gráfica fue diseñada bajo una estructura modular basada en pestañas, permitiendo seleccionar distintos tipos de entrada de datos: imagen estática, captura de cámara en tiempo

real, video pregrabado y dashboard. Cada módulo cuenta con controles específicos para la carga o captura de datos, botones de ejecución del análisis y secciones dedicadas a la visualización de resultados.

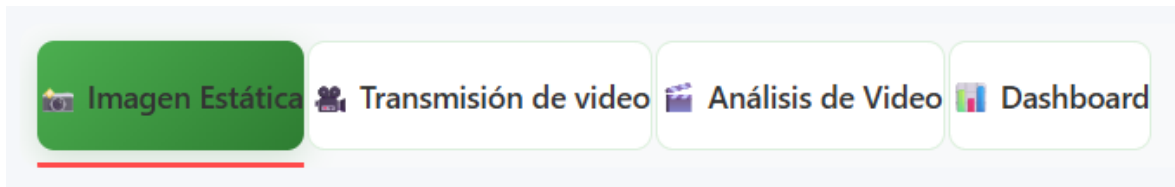


Figura 40 Módulos de la interfaz. (Elaboración propia).

El sistema muestra las imágenes y videos originales junto con los resultados procesados, en los cuales se resaltan las detecciones mediante rectángulos delimitadores (bounding boxes) y etiquetas visuales. Además, se incorporaron indicadores visuales y métricas clave como el número total de detecciones, niveles de confianza promedio y estadísticas por categoría, lo que mejora la interpretación de los resultados.



Figura 41 Video resultado con bounding boxes. (Elaboración propia).

Asimismo, la interfaz permite la descarga de resultados en formato CSV, facilitando el análisis posterior y la generación de reportes. La disposición de columnas, gráficos y tablas fue optimizada para una visualización clara y ordenada, manteniendo una experiencia de usuario accesible incluso para usuarios sin conocimientos técnicos avanzados.

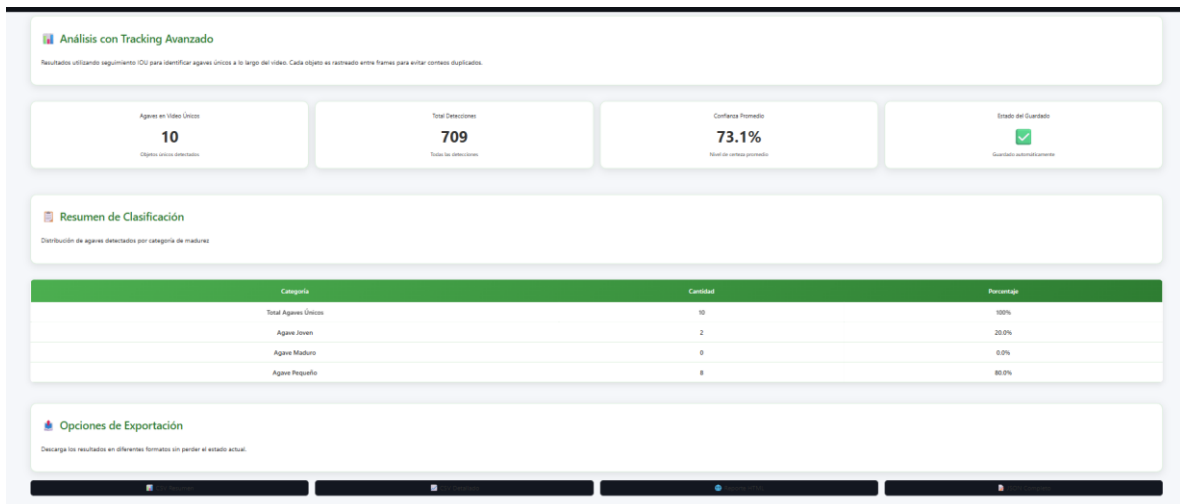


Figura 42 Generación de tablas y reportes. (Elaboración propia).



Figura 43 Generación de graficas. (Elaboración propia).

9.4.2 Dashboard

Cuenta con un apartado de Dashboard donde permite mirar las estadísticas de análisis históricos, esto quiere decir que muestra los resultados de todos los análisis que se han realizado hasta ese día.

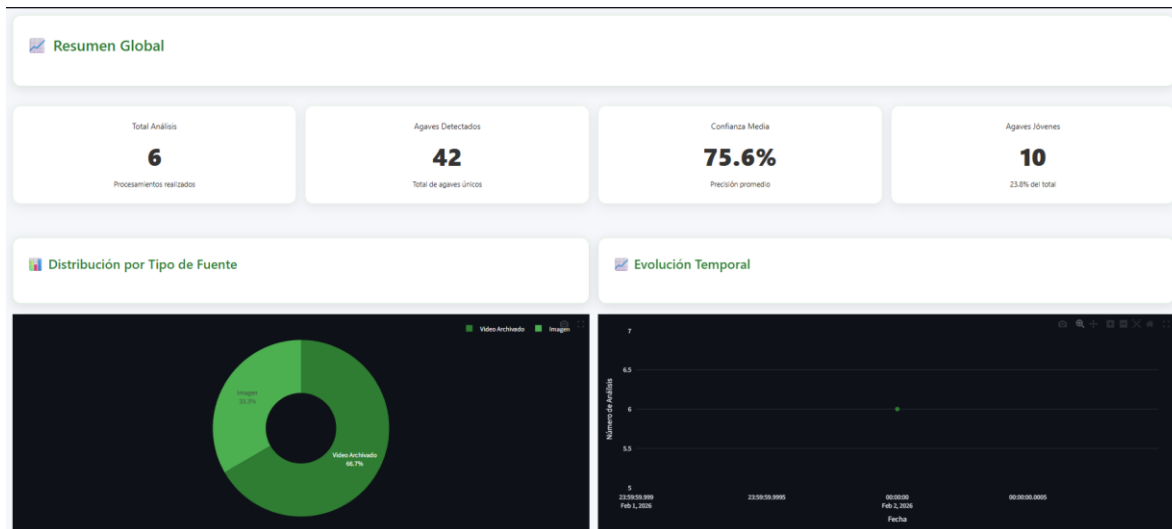


Figura 44 Dashboard histórico. (Elaboración propia).

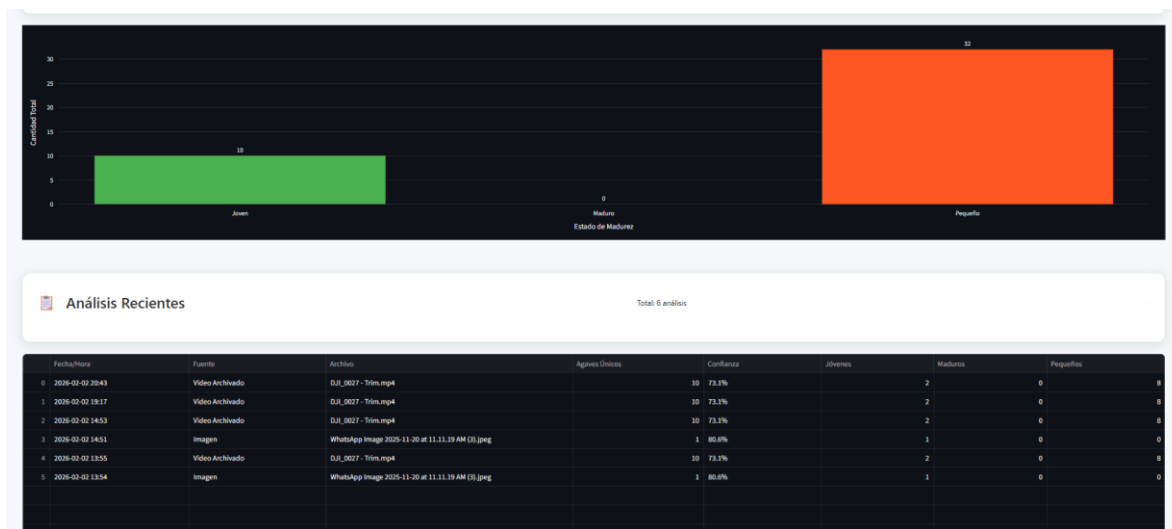


Figura 45 Dashboard histórico. (Elaboración propia).

9.4.3 Integración del análisis automático

El modelo de detección previamente entrenado fue integrado directamente en la aplicación web para permitir el análisis automático de imágenes y videos al momento de ser cargados o capturados. Esta integración se realiza mediante la ejecución del proceso de detección desde la propia aplicación, utilizando llamadas controladas al modelo y procesando sus salidas en tiempo real.

```

# --- Constantes ---
MODEL_FILE = "best.pt"
OUTPUT_DIR = pathlib.Path("runs/detect_app")
TEMP_DIR = OUTPUT_DIR / "temp_uploads"
TEMP_DIR.mkdir(parents=True, exist_ok=True)

# --- Clases de Madurez ---
CLASES = {0: 'AGAVE', 1: 'GERMINACION', 2: 'JUVENIL', 3: 'MADURO'}

# --- Título ---
st.title("🌱 Detector de Agave - Pro V2")
st.write("Sistema de conteo de alta precisión usando Superposición y Persistencia.")
st.divider()

if not os.path.exists(MODEL_FILE):
    st.error(f"❌ No se encontró el modelo '{MODEL_FILE}'.")
    st.stop()

```

Figura 46 Análisis automático. (Elaboración propia).

El sistema genera automáticamente archivos de resultados que incluyen las coordenadas de detección, la clase asignada y el nivel de confianza de cada predicción. A partir de esta información, se realiza un análisis adicional que permite contabilizar el número total de plantas detectadas y clasificarlas según su etapa de madurez: germinación, juvenil o maduro. Ver Anexo 2.

Para el análisis de video, se implementó un mecanismo de seguimiento basado en la intersección sobre unión (IoU), el cual permite identificar plantas únicas a lo largo de múltiples fotogramas. Este enfoque evita el conteo duplicado de una misma planta y mejora la precisión del sistema en escenarios dinámicos. Cada planta es asociada a un identificador único y se registran todas las clasificaciones observadas durante su permanencia en el video.

Los resultados del análisis se presentan mediante visualizaciones gráficas, incluyendo histogramas de confianza y gráficos de distribución por etapa de madurez, además de tablas resumen con métricas globales. Este proceso automático permite obtener información cuantitativa y visual confiable, contribuyendo a una mejor evaluación del estado del cultivo y apoyando la toma de decisiones en la planeación agrícola.

```

def iou_batch(bb_test, bb_gt):
    bb_gt = np.expand_dims(bb_gt, 0)
    bb_test = np.expand_dims(bb_test, 1)
    xx1 = np.maximum(bb_test[..., 0], bb_gt[..., 0])
    yy1 = np.maximum(bb_test[..., 1], bb_gt[..., 1])
    xx2 = np.minimum(bb_test[..., 2] + bb_test[..., 3], bb_gt[..., 2] + bb_gt[..., 3])
    yy2 = np.minimum(bb_test[..., 1] + bb_test[..., 3], bb_gt[..., 1] + bb_gt[..., 3])
    w = np.maximum(0., xx2 - xx1)
    h = np.maximum(0., yy2 - yy1)
    wh = w * h
    o = wh / ((bb_test[..., 2] * bb_test[..., 3]) + (bb_gt[..., 2] * bb_gt[..., 3]) - wh)
    return o

class IOUTracker:
    def __init__(self, max_lost=5, iou_threshold=0.3):
        self.next_id = 1
        self.tracks = {}
        self.max_lost = max_lost
        self.iou_threshold = iou_threshold

    def update(self, detections):
        updated_tracks = {}
        if not self.tracks:
            for det in detections:
                self.tracks[self.next_id] = list(det) + [0]
                self.next_id += 1
            return self.tracks

        track_ids = list(self.tracks.keys())
        track_boxes = np.array([self.tracks[tid][:4] for tid in track_ids])
        det_boxes = np.array(detections)

        if len(det_boxes) > 0 and len(track_boxes) > 0:
            ious = iou_batch(det_boxes, track_boxes)
            if ious.size > 0:
                max_ious = ious.max(axis=1)
                max_indices = ious.argmax(axis=1)
                used_tracks = set()
                for i, (iou, track_idx) in enumerate(zip(max_ious, max_indices)):
                    if iou >= self.iou_threshold:
                        tid = track_ids[track_idx]
                        if tid not in used_tracks:
                            updated_tracks[tid] = list(det_boxes[i]) + [0]
                            used_tracks.add(tid)
                    else: pass
                for tid in self.tracks:
                    if tid not in updated_tracks:
                        box = self.tracks[tid]
                        lost_count = box[4] + 1
                        if lost_count <= self.max_lost:
                            box[4] = lost_count
                            updated_tracks[tid] = box

```

Figura 47 Tracking IoU. (Elaboración propia).

Ver ANEXO 2.

9.5 Pruebas y validación del sistema

El desempeño de los modelos entrenado fue evaluado utilizando métricas estándar ampliamente aceptadas en visión por computadora:

- Precisión (Precision): Indica qué proporción de las detecciones realizadas por el modelo son correctas.
- Recall: Mide la capacidad del modelo para detectar todas las plantas presentes en la imagen.

- F1-score: Métrica que combina precisión y recall, proporcionando una evaluación balanceada.
- mAP (mean Average Precision): Evalúa la calidad global de las detecciones considerando la ubicación y la clasificación correcta de los objetos.

Clase	PRECISIÓN	RECALL	mAP50	mAP50-95	DIAGNÓSTICO
AGAVE	71%	89.2%	82.9%	43.6%	Excelente detección
GERMINACIÓN	48.3%	90.5%	67.3%	37.5%	Muchos falsos positivos
JUVENIL	38.7%	87.5%	58%	39.1%	Baja precisión
MADURO	50.8%	16.7%	31.3%	26.3%	Muy pocas muestras

Tabla 8 Pruebas y validación del sistema del primer modelo. (Elaboración propia).

Clase	PRECISIÓN	RECALL	mAP50	mAP50-95	DIAGNÓSTICO
PEQUEÑO	94%	97%	98%	69%	Excelente detección
JOVEN	97%	98%	98%	69%	Excelente detección
MADURO	92%	95%	98%	76%	Excelente detección

Tabla 9 Pruebas y validación del sistema del segundo modelo. (Elaboración propia).

Los resultados obtenidos en el primer modelo mostraron un mejor desempeño en la clase AGAVE y GERMINACION, mientras que la clase JUVENIL presentó mayor dificultad debido a similitudes visuales con otras etapas. Aun así, los valores obtenidos demuestran que el modelo es funcional y adecuado para aplicaciones reales de monitoreo agrícola.

Los resultados obtenidos en el segundo modelo muestran un mejor desempeño en la clase JOVEN, aunque en este modelo todas las clases presentaron buenos resultados superando al modelo uno.

9.6 Resultados obtenidos

Como parte de los resultados se observó que el sistema proceso imágenes y videos exitosamente, detectado así plantas de agave con alta precisión. Comparado a métodos manuales mostro una significativa reducción de tiempo requerido para el monitoreo, manteniendo calidad aceptable en los resultados.

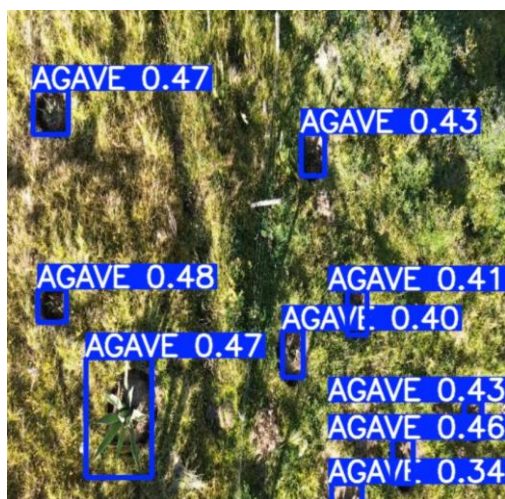


Figura 48 Resultados de video procesada con el modelo uno. (Elaboración propia).

Agaves en Foto	Confianza Promedio	Mínima Confianza
9	43.25%	33.97%

Figura 49 Porcentaje de confianza mínima y máxima con el modelo uno. (Elaboración propia).



Figura 50 Resultados de video procesada con el modelo dos. (Elaboración propia).



Figura 51 Porcentaje de confianza mínima y máxima con el modelo dos. (Elaboración propia).

9.7 Elementos creados durante el desarrollo

Durante el desarrollo del sistema se generaron los siguientes elementos:

- Un conjunto de imágenes etiquetadas
- Modelo YOLOv5 entrenado para detección y clasificación de la madurez
- Aplicación web completa desarrollada con el Framework Streamlit
- Imágenes y videos procesados con anotaciones
- Gráficas de métricas de evaluación

9.8 Resumen

Dentro del presente capítulo se mostró todas las actividades realizadas durante la residencia profesional, cubriendo las fases completas de análisis, diseño, desarrollo, pruebas e implementación del sistema. Se cumplieron los objetivos específicos del proyecto, demostrando la viabilidad técnica de la solución automatizada para monitoreo de cultivos de agave.

10. CONCLUSIONES

Como conclusión, se muestra que el objetivo general del proyecto, que consiste en desarrollar un sistema inteligente basado en redes neuronales convolucionales y tecnologías como drones para la detección, el conteo y la clasificación automática de plantas de agaves, cumplió de manera satisfactoria. En el transcurso del estudio se reafirmó que los métodos tradicionales de monitoreo en el cultivo del agave presentan limitaciones de tiempo, costo y planeación. Las limitaciones encontradas justificaron el desarrollo de una solución tecnológica diseñada para la automatización del proceso de conteo y análisis del cultivo.

El proyecto logró implementar un flujo integral que abarca la adquisición y preparación de datos, el entrenamiento de una red neuronal convolucional utilizando un conjunto de datos conformado por imágenes aéreas y terrestres de cultivos de agave tomadas por drones DJI, y la integración del modelo entrenado en una aplicación web.

Los resultados han demostrado que el uso de modelos de detección y vehículos aéreos no tripulados realmente presenta una alternativa viable para el monitoreo agrícola; su uso permitió identificar plantas de agave de manera sistemática y su clasificación. Por otra parte, se observó que la calidad del etiquetado y la variedad de imágenes realmente influyen directamente en el desempeño del modelo, el cual representa una de las principales limitaciones del estudio. No obstante, estas observaciones han abierto nuevas oportunidades de mejora para perfeccionar la precisión del sistema.

En conclusión, el proyecto aporta al campo de la agricultura al demostrar la viabilidad de integrar inteligencia artificial, visión por computadora, drones y plataformas web, abriendo nuevas brechas y sentando bases para nuevas y futuras investigaciones relacionadas con la mejora del conteo y la clasificación de madurez. De manera general, la investigación ha reforzado la importancia de implementar tecnologías emergentes en procesos agrícolas, buscando contribuir tanto al conocimiento académico como al desarrollo de herramientas prácticas para el sector agrícola.

11. RECOMENDACIONES

De acuerdo con los resultados que obtuvimos y a todo el análisis que se realizó, una de las recomendaciones principales es ampliar el conjunto de datos utilizado para el entrenamiento del modelo; se recomienda añadir imágenes captadas desde diferentes regiones, etapas fenológicas y condiciones ambientales, con el fin de poder mejorar la eficiencia y la capacidad del sistema, así como mejorar las métricas de evaluación.

Otra recomendación que ayudará al proyecto a ir a un nuevo nivel es explorar arquitecturas de redes neuronales convolucionales más avanzadas que la que se utilizó, así como ampliar las técnicas de optimización que permitan mejorar considerablemente la precisión, el recall y la velocidad de procesamiento del sistema. De igual manera, sería pertinente integrar módulos adicionales para el almacenamiento de datos históricos y la generación automática de reportes mejorados.

Por último, algunas recomendaciones que pueden ayudar a mejorar este proyecto son las pruebas más extensas en diferentes condiciones reales de campo, utilizando la transmisión de video en tiempo real desde diferentes tipos de drones, con el fin de evaluar el desempeño del sistema en escenarios dinámicos y poder valorar su funcionamiento operativo. Al final y de acuerdo con los resultados de la investigación, se recomienda analizar la posibilidad de adoptar el sistema y las tecnologías utilizadas a otros cultivos agrícolas, y así permitir evaluar su escalabilidad y ampliar su impacto en el sector productivo.

12. FUENTES BIBLIOGRÁFICAS

- Ahmed, N., De, D., & Hussain, I. (2018). Internet of Things (IoT) for Smart Precision Agriculture and Farming in Rural Areas. *IEEE Internet Of Things Journal*, 5(6), 4890-4899. <https://doi.org/10.1109/jiot.2018.2879579>
- Bewley, A., Ge, Z. O., Ramos, F., & Upcroft, B. (2016). Simple Online and Realtime Tracking. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.1602.00763>
- Bradski, G. (2005). *Dr. Dobb's Journal of Software Tools*. Obtenido de The OpenCV Library: <https://opencv.org/about/>
- Calvario, G., Alarcón, T. E., Dalmau, O., Sierra, B., & Hernández, C. (2020). An Agave Counting Methodology Based on Mathematical Morphology and Images Acquired through Unmanned Aerial Vehicles. *Sensors*, 6247. <https://doi.org/10.3390/s20216247>
- Chanchí, G. E., Ospina, M. A., & Campo, W. Y. (2021). IoT Architecture for Monitoring Variables of Interest in Indoor Plants. *Computación y Sistemas*, 25. <https://doi.org/10.13053/cys-25-4-4051>
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0: Step-by-step data mining guide*. SPSS Inc. Retrieved from <https://www.kddlab.com/files/CRISPPWP-0800.pdf>
- Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 40(4), 834-848. <https://doi.org/10.1109/tpami.2017.2699184>
- Chinnadurai, S., & Selvakumar, S. (2025). mage-based cotton leaf disease diagnosis using YOLO and faster R-CNN techniques. *Scientific Reports*, 15(1), 45785. <https://doi.org/10.1038/s41598-025-28549-7>
- Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. *In Proceedings of the 23rd international conference on Machine learning*. <https://doi.org/10.1145/1143844.1143874>
- Deng, L., Mao, Z., Li, X., Hu, Z., Duan, F., & Yan, Y. (2018). UAV-based multispectral remote sensing for precision agriculture: A comparison between different cameras. *ISPRS Journal Of Photogrammetry And Remote Sensing*, 146, 124-136. <https://doi.org/10.1016/j.isprsjprs.2018.09.008>
- Escobar-Flores, J. G., Sandoval, S., & Gámiz-Romero, E. (2022). Unmanned aerial vehicle images in machine learning for agave detection. *Environmental Science and Pollution Research*, 61662-61673. <https://doi.org/10.1007/s11356-022-18985-7>

- Espinoza Hernandez, J., Lopez Canteñs, G. d., & López Cruz, I. L. (2023). Agave plant density using convolutional neural networks on aerial imagery. *Agrociencia*, 1003–1016. <https://doi.org/10.47163/agrociencia.v57i7.2970>
- Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2009). The Pascal Visual Object Classes (VOC) challenge. *International Journal Of Computer Vision*, 88, 303–33. <https://doi.org/10.1007/s11263-009-0275-4>
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, 27(8), 861–874. <https://doi.org/10.1016/j.patrec.2005.10.010>
- Flores, D., González Hernández, I., Lozano, R., & Vazquez, N. (2021). Automated Agave Detection and Counting Using a Convolutional Neural Network and Unmanned Aerial Systems. *Drones*, 4. <https://doi.org/10.3390/drones5010004>
- Galdames, I. S. (2023). Inteligencia artificial en investigación científica. *SciComm Report*, 3, 1–3.
- Gobierno del Estado de México. (2021). *Fortalecimiento a la producción de agave mezcalero en el Estado de México*. Secretaría de Desarrollo Agropecuario (SECAmPo).
- Godin, M. (2020). Google Colab for machine learning projects. *Towards Data Science*.
- Gonzalez, R. C., & Woods, R. E. (2018). *Digital image processing (4th ed.)*. Pearson.
- Goodfellow, Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Grinberg, M. (2018). *Flask Web Development, 2nd Edition*. O'Reilly Media.
- Guardado Fierros, B. G., Tuesta Popolizio, D. A., Lorenzo Santiago, M. A., Rubio Cortés, R., Camacho Ruíz, R. M., CastañedaNava, J. J., & Contreras Ramos, S. M. (2024). PGPB Consortium Formulation to Increase Fermentable Sugar in Agave tequilana Weber var. *Blue: A Study in the Field. Plants*, 13, 1371. <https://doi.org/10.3390/plants13101371>
- Gutiérrez Vaca, C., Serwatowski Hlawinska, R. J., & Martínez Jaime, O. A. (2013). Determinación de las características de las cuchillas de corte para mecanizar la cosecha de agave. *Determinación de las características de las cuchillas de corte para mecanizar la cosecha de agave*, 4(1), 159–166.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.1512.03385>
- Hernández Calvario, O., Florian, F., Sánchez, M. G., & Avila George, H. (2022). Conteo de plantas de agave usando redes neuronales convolucionales. *Revista Iberoamericana de Sistemas de Tiempo Real*, 64–78.
- Hernández Soriano, L., Sandre, G., De Dios, E. Á., & Simpson, J. (2023). How to awaken a sleeping giant: antagonistic expression of Flowering locus T homologs and elements of the age-related pathway are associated with the flowering transition in Agave tequilana. *Plant Reproduction*, 37(2), 111–132. <https://doi.org/10.1007/s00497-023-00489-0>

- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2015). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 15, 1097–1105.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.
<https://doi.org/10.1038/nature14539>
- Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., . . . Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *In European Conference on Computer Vision*. Springer International Publishing. https://doi.org/10.1007/978-3-319-10602-1_48
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., & Berg, A. C. (2015). SSD: Single Shot MultiBox Detector. *arXiv (Cornell University)*.
<https://doi.org/10.48550/arxiv.1512.02325>
- Liu, Y., Sun, L., Liu, B., Wu, Y., Ma, J., Zhang, W., & Chen, Z. (2023). Estimation of Winter Wheat Yield Using Multiple Temporal Vegetation Indices Derived from UAV-Based Multispectral and Hyperspectral Imagery. *Remote Sensing*, 15, 4800.
<https://doi.org/10.3390/rs15194800>
- Long, J., Shelhamer, E., & Darrell, T. (2014). Fully Convolutional Networks for Semantic Segmentation. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.1411.4038>
- López Alfaro, G. A., Serrano Rubio, J. P., Rodríguez Vidal, L. M., & Herrera Guzmán, R. (2023). Methodology for agave plants recognition based on superpixels segmentation and drone images. *IEEE International Autumn Meeting on Power, Electronics and Computing*. ROPEC.
<https://doi.org/10.1109/ROPEC58757.2023.10409453>
- Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.1711.05101>
- Martínez Rodríguez, L. (2024). *Sistema de inspección automatizada de plantas*. Universidad de Alicante.
- Mendoza, A. G. (2007). *Los agaves de México*. Ciencias.
- Mizik, T. (2022). How can precision farming work on a small scale? A systematic literature review. *Precision Agriculture*, 24, 384-406. <https://doi.org/10.1007/s11119-022-09934-y>
- Mora Castro, G., Ayala Cruz, K. V., Valenzuela Hernández, J., & Romero Fitch, J. (2025). Procesamiento de imágenes con UAV en la agricultura de precisión: panorama actual, tendencias tecnológicas, una revisión sistemática. *Revista de Investigación En Tecnologías de la Información (RITI)*, 4-16.
- Nex, F., & Remondino, F. (2014). UAV for 3D mapping applications: a review. *Applied Geomatics*, 6, 1-5. <https://doi.org/10.1007/s12518-013-0120-x>
- Pajares, G., & Cruz, J. M. (2006). *Visión por computador: Imágenes y segmentación*. Alfaomega. Obtenido de https://img1.wsimg.com/blobby/go/541a1d8b-7a21-4c1f-8013-03406bd1a8ad/downloads/vision_por_computador_gonzalo_pajares.pdf

- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., & Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv (Cornell University)*, 32, 8026-8037. <https://doi.org/10.48550/arxiv.1912.01703>
- Pathmudi, V. R., Khatri, N. K., Abdul-Qawy, A. S., & Vyas, A. K. (2023). A systematic review of IoT technologies and their constituents for smart and sustainable agriculture applications. *Scientific African*, 19. <https://doi.org/10.1016/j.sciaf.2023.e01577>
- Pérez Ortega, D. J., Bolaños Alomia, F. A., & Da Silva, A. M. (2021). Variables que influyen en la aplicación de la agricultura de precisión en Colombia: revisión de estudios. *Ciencia y Tecnología Agropecuaria*, 23(1). https://doi.org/10.21930/rcta.vol23_num1_art:2298
- Pérez, L. H., Pacheco, E. V., Fletes, I. O., Campante, M. A., Hernández-Plascencia, J. A., & Rodríguez-Macías, R. (2017). Prácticas agroecológicas en Agave tequilana Weber bajo dos sistemas de cultivo en Tequila, Jalisco. *Revista Mexicana de Ciencias Agrícolas*. <https://doi.org/10.29312/remexca.v8i18.216>
- Pérez-Guerra, G. A., Sosa-Franco, I., Machado-García, N., & Ruiz-Pérez, M. E. (2023). Herramientas SIG, revisión de sus fundamentos, tipos y relación con las bases de datos espaciales. *Revista Ciencias Técnicas Agropecuarias*, 32. Retrieved from <https://cu-id.com/2177/v32n3e10>. Recuperado a partir de <https://revistas.unah.edu.cu/index.php/rcta/article/view/1747>
- Ponce Corona, J. E. (2020). Algoritmo para el conteo de agaves usando imágenes aéreas tomadas desde un vehículo aéreo no tripulado. *Instituto Tecnológico de Ciudad Guzmán / Tecnológico Nacional de México*.
- Powers, D. M. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies*.
- Radoglou Grammatikis, P., Sarigiannidis, P., Lagkas, T., & Moscholios, I. (2020). A compilation of UAV applications for precision agriculture. *Computer Networks*, 172. <https://doi.org/10.1016/j.comnet.2020.107148>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. Unified, real-time object detection. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 779-788.
- Rejeb, A., Abdollahi, A., Rejeb, K., & Treiblmaier, H. (2022). Drones in agriculture: A review and bibliometric analysis. *Computers And Electronics In Agriculture*, 198, 107017. <https://doi.org/10.1016/j.compag.2022.107017>
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv (Cornell University)*, 28, 91-99. <https://doi.org/10.48550/arxiv.1506.01497>

- Rouse, J. W., Jr, H. R., Schell, J. A., & Deering, D. W. (1974, Enero 1). *NTRS - NASA Technical Reports Server*. Retrieved from Monitoring vegetation systems in the Great Plains with ERTS: <https://ntrs.nasa.gov/citations/19740022614>
- Russakovsky, O., Deng, J., Su, H. K., Satheesh, S., Ma, S., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal Of Computer Vision*, 115, 211-252. <https://doi.org/10.1007/s11263-015-0816-y>
- Secretaria del campo. (2023). *Promueve Secretaría del Campo producción de agave mezcalero en el Estado de México*. Gobierno del Estado de México.
- Shahi, T. B., Xu, C., Neupane, A., & Guo, W. (2022). Machine learning methods for precision agriculture with UAV imagery: a review. *Electronic Research Archive*, 20(12), 4277-4317. <https://doi.org/10.3934/era.2022218>
- Shahi, T. B., Xu, C., Neupane, A., & Guo, W. (2023). Recent Advances in Crop Disease Detection Using UAV and Deep Learning Techniques. *Remote Sensing*, 15(9), 2450. <https://doi.org/10.3390/rs15092450>
- Sharma, K., & Shivandu, S. K. (2024). Integrating artificial intelligence and Internet of Things (IoT) for enhanced crop monitoring and management in precision agriculture. *Sensors International*. <https://doi.org/10.1016/j.sintl.2024.100292>
- Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. *IEEE Internet Of Things Journal*, 3(5), 637-646. <https://doi.org/10.1109/jiot.2016.2579198>
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal Of Big Data*, 6. <https://doi.org/10.1186/s40537-019-0197-0>
- Słomska Przech, K. P. (2021). Heat Maps: Perfect Maps for Quick Reading? Comparing Usability of Heat Maps with Different Levels of Generalization. *ISPRS International Journal Of Geo-Information*, 10(8), 562. <https://doi.org/10.3390/ijgi10080562>
- Szeliski, R. (2021). *Computer vision: Algorithms and applications (2nd ed.)*. Springer. Retrieved from <https://library.huree.edu.mn/data/202295/2024-06-03/Computer%20Vision%20-%20Algorithms%20and%20Applications%202nd%20Edition,%20Richard%20Szeliski.pdf>
- Tapia Rodriguez, A., Ramírez Davila, J. F., & Acosta Guadarrama, A. D. (2023). Plagas y enfermedades del agave mezcalero. *Enfoque Rural*, 1-10.
- Thenkabail, P. S., Lyon, J. G., & Huete, A. (2019). *Advanced applications in remote sensing of agricultural crops and natural vegetation*. CRC Press.
- Tovar Quiroz, A. D. (2023). Agricultura 4.0: uso de tecnológicas de precisión y aplicación para pequeños productores. *Informador Técnico*, 87. <https://doi.org/10.23850/22565035.5536>
- Tsouros, D. C., Bibi, S., & Sarigiannidis, P. G. (2019). A Review on UAV-Based Applications for Precision Agriculture. *Information*, 10, 349. <https://doi.org/10.3390/info10110349>
- Van Rijsbergen, C. J. (2000). *Information retrieval (2nd ed.)*. Butterworths.

- Velásquez Ramirez, L. (25 de Febrero de 2022). En 15 municipios del EdoMéx se cultiva agave mezcalero. *El Valle*.
- Villatoro Geronimo, D., Sanchez Ante, G., & Falcon Morales, L. E. (2025). Bit-STED: A lightweight transformer for accurate agave counting with UAV imagery. *Computers And Electronics In Agriculture*, 239. <https://doi.org/10.1016/j.compag.2025.111047>
- Villatoro, D., Sanchez-Ante, G., & Luis, F.-M. (2024). Precision Agriculture Through ARVI-Based Segmentation of Agave Weber Using UAV Multispectral Imaging. *SSRN*.
- Wohlers, T. (2002). *Applying AutoCAD 2002 Advanced, Student Edition*. McGraw-Hill Education. <https://doi.org/978-0-07-828542-4>
- Wojke, N., Bewley, A., & Paulus, D. (2017). Simple Online and Realtime Tracking with a Deep Association Metric. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.1703.07402>
- Zhang, Z., Wang, X., Wang, W., Zeng, W., Luo, Z., Zheng, L., & Yang, Y. (2021). FairMOT: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129(11), 3069–3087. <https://doi.org/10.1007/s11263-021-01436-z>

13. ANEXOS

ANEXO 1 A MANUAL DE USUARIO

Manual de usuario

Detector de Agave Pro V3

1. Introducción

El detector de agave Pro V3 es una aplicación web que fue desarrollada con el framework de Streamlit que permite la detección automática, el conteo y la clasificación por madurez de la planta de agave a partir de imágenes estáticas, video transmisión en vivo desde cámara y vehículos aéreos no tripulados (drones).

El sistema utiliza modelos de detección entrenados con YOLOv5 que es un tipo de red neuronal convolucional (CNN), combinando técnicas de tracking IoU (Intersection over Union), lo que nos ayuda a garantizar:

- Conteo preciso de agaves únicos.
- Reducción de duplicados en video.
- Clasificación por etapas de madurez.
- Métricas estadísticas confiables.

Este manual describe el paso a paso del uso completo de la aplicación, incluyendo que opciones seleccionar, que botones presionar y que tipo de imágenes o videos se debe de cargar en cada sección.

2. Requisitos del sistema

Antes de poder utilizar la aplicación, asegúrese de cumplir los siguientes requisitos:

2.1 Requisitos de software

- Sistema operativo: Windows 10/11 /Linux / macOS
- Python 3.9 o superior

- Streamlit instalado
- OpenCV
- PyTorch

2.2 Requisitos de Hardware

- Cámara web (para transmisión de video)
- GPU recomendado (opcional, para una mayor velocidad)
- Dron con cámara RGB (opcional, para análisis aéreo)

3. Acceso a la aplicación

- Abre una terminal o consola
- Ubíquese en la carpeta del proyecto
- Ejecutar el siguiente proyecto

Streamlit run app.py

- Se abrirá automáticamente el navegador con la interfaz del sistema

4. Interfaz principal

Al iniciar la aplicación se muestra la pantalla principal con la siguiente información:

- Título: Detector de Agave – Pro V3
- Descripción: Sistema de conteo de alta precisión usando superposición y persistencia.
- Cuatro pestañas principales:
 - Imagen Estática
 - Transmisión de Video
 - Análisis de Video
 - Dashboard

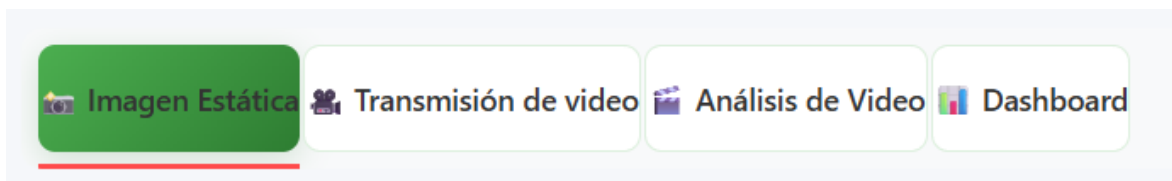


Figura 52 Módulos principales de la página web. (Elaboración propia).

5. Pestaña 1 – Imagen estática

5.1 Descripción

Esta sección permite analizar imágenes estáticas para detectar agaves y clasificarlos por madurez.

5.2 Pasos de uso

- Seleccione la pestaña Imagen Estática.
- Presione el botón “Seleccionar imagen...”.
- Cargue una imagen en formato:
 - JPG
 - PNG
 - JPEG
- La imagen original que se subió se mostrará en la pantalla.
- Presione “Detectar y Analizar Imagen”.



Figura 53 Ejemplo de imagen para analizar. (Elaboración propia).

5.3 Resultados mostrados

- Imagen con Boulding boxes de detección.
- Total de plantas de agave detectadas.
- Confianza de promedio de la detección.
- Tabla de resumen.
- Grafica de distribución de confianza.
- Grafica de pirámide de madurez.



Figura 54 Resultado del análisis. (Elaboración propia).



Figura 55 Resultados de la detección. (Elaboración propia).

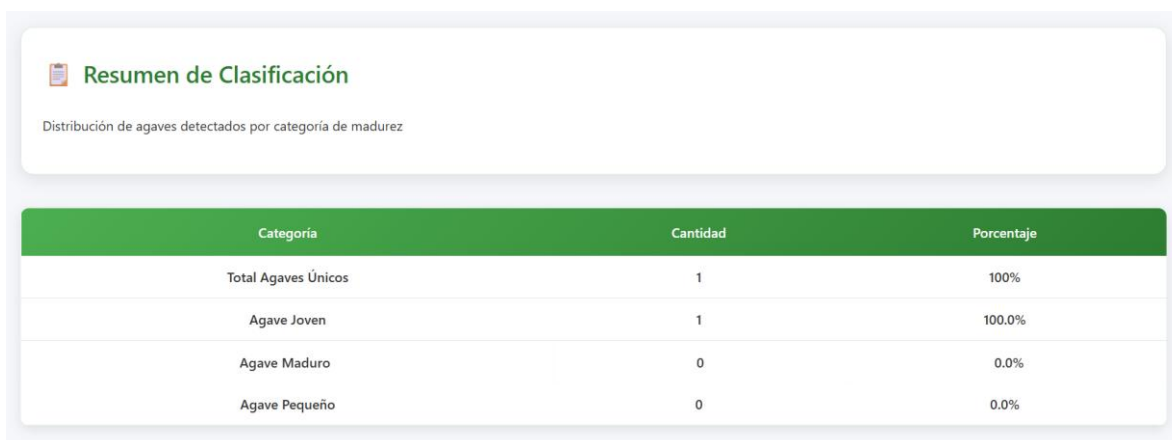


Figura 56 Resumen de clasificación. (Elaboración propia).

6. Pestaña 2 – Transmisión en vivo

6.1 Descripción

Permite grabar el video directamente desde la cámara o el dron lo que se le sea asignado y analizarlo posteriormente.

6.2 Pasos de uso transmisión de video

- Seleccionar la pestaña Transmisión en vivo.
- Presione Iniciar Grabación.
- La cámara se activará y mostrará la transmisión en tiempo real.
- Presione Finalizar cuando termine la grabación.
- Se mostrará el video grabado.
- Presione Detectar video Grabado.

6.3 Pasos de uso Dron

- Seleccione RTSP (stream en vivo).
- Ingrese la URL RTSP del dron.

Ejemplo:

rtsp://192.168.1.1:554/live

- Presione iniciar grabación.

6.4 Resultados

- Video con detecciones
- Conteo total
- Conteo de agaves únicos (tracking)
- Clasificación por madurez
- Descarga de resultados en CSV



Figura 57 Resultado de análisis de video. (Elaboración propia).

7. Pestaña 3 – Análisis de video

7.1 Descripción

Permite analizar videos previamente grabados, aplicando tracking para evitar conteos duplicados.

7.2 Pasos de uso

- Seleccione la pestaña Video.
- Presione Cargar video....
- Seleccione un archivo en formato:
 - MP4
 - AVI
 - MOV
 - MKV

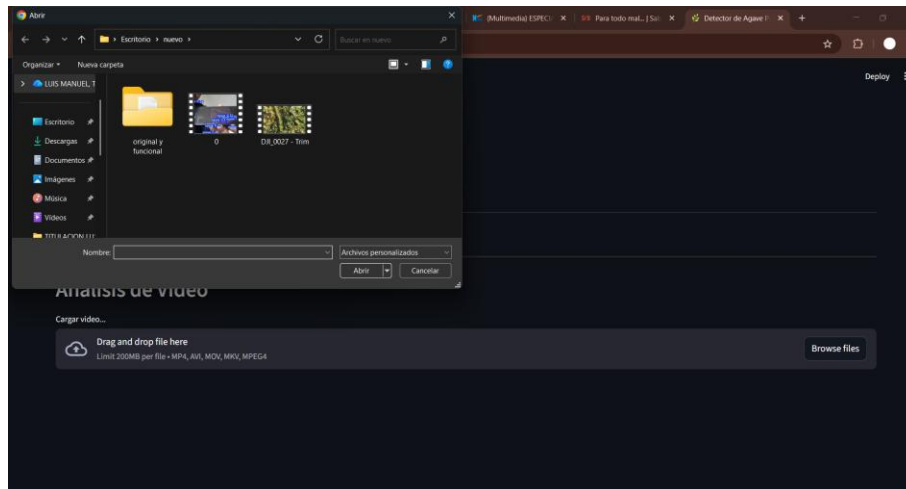


Figura 58 Proceso para subir video. (Elaboración propia).

- Visualice el video original.
- Presione Procesar Video.

7.3 Resultados

- Video con detecciones
- Conteo total de detecciones
- Conteo de agaves únicos
- Gráfica de madurez
- Métricas estadísticas



Figura 59 Resultado de análisis de video. (Elaboración propia).

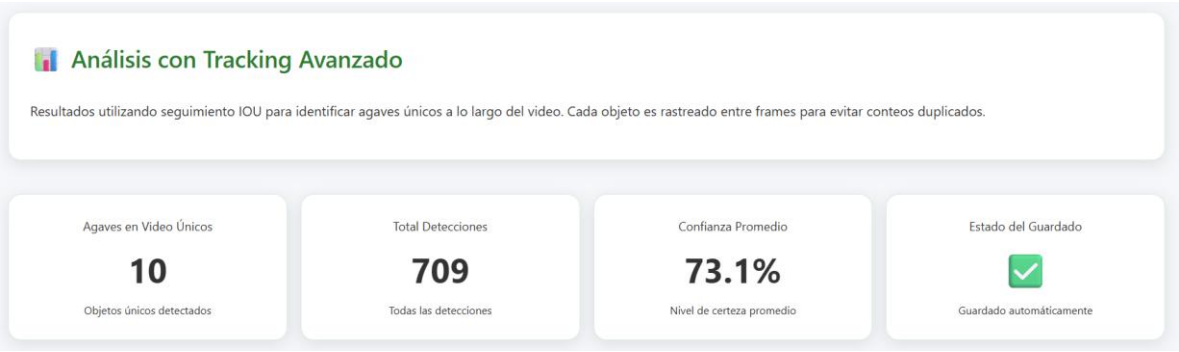


Figura 60 Análisis con Tracking avanzado. (Elaboración propia).

Categoría	Cantidad	Porcentaje
Total Agaves Únicos	10	100%
Agave Joven	2	20.0%
Agave Maduro	0	0.0%
Agave Pequeño	8	80.0%

Figura 61 Tabla de resultados. (Elaboración propia).



Figura 62 Análisis estadístico. (Elaboración propia).

8. Pestaña 4 – Dashboard

8.1 Descripción

Diseñada para mostrar datos visuales de análisis históricos

8.2 Resumen global

Muestra el resumen de todos los análisis que se han hecho hasta ese día



Figura 63 Resumen global. (Elaboración propia).

8.3 Distribución por tipo fuente y evolución temporal

Muestra una gráfica de pastel donde particiona el uso de las opciones utilizadas, la evolución temporal es el reporte en grafica de la fecha de los análisis realizados.

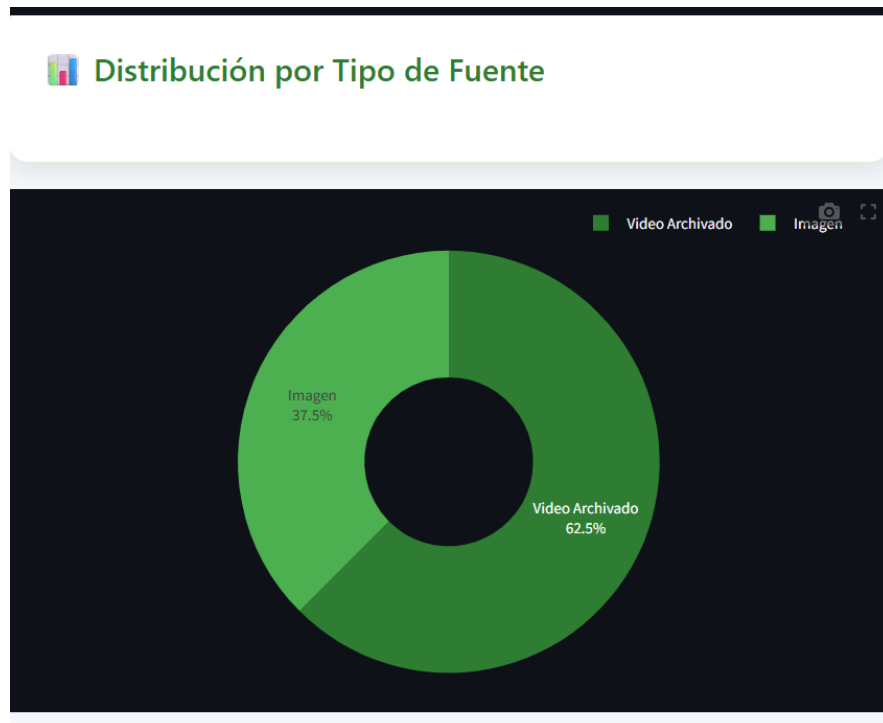


Figura 64 Distribución por tipo de fuente. (Elaboración propia).

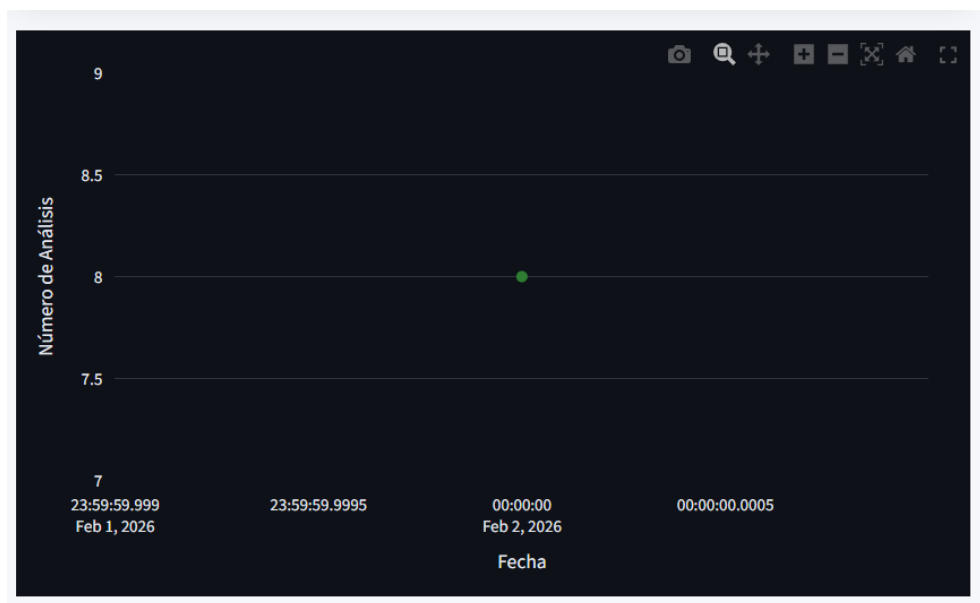


Figura 65 Evolución temporal. (Elaboración propia).

8.4 Distribución por madurez

Muestra la cantidad de agaves en total que se han encontrado hasta el momento en todos los análisis.



Figura 66 Distribución por estado de madurez. (Elaboración propia).

8.5 Análisis recientes

	Fecha/Hora	Fuente	Archivo	Agaves Unicos	Confianza	Jóvenes	Maduros	Pequeños
0	2026-02-02 22:21	Video Archivado	DJI_0027 - Trim.mp4	10	73.1%	2	0	8
1	2026-02-02 22:13	Imagen	WhatsApp Image 2025-11-20 at 11.11.19 AM (3).jpeg	1	80.6%	1	0	0
2	2026-02-02 20:43	Video Archivado	DJI_0027 - Trim.mp4	10	73.1%	2	0	8
3	2026-02-02 19:17	Video Archivado	DJI_0027 - Trim.mp4	10	73.1%	2	0	8
4	2026-02-02 14:53	Video Archivado	DJI_0027 - Trim.mp4	10	73.1%	2	0	8
5	2026-02-02 14:51	Imagen	WhatsApp Image 2025-11-20 at 11.11.19 AM (3).jpeg	1	80.6%	1	0	0
6	2026-02-02 13:55	Video Archivado	DJI_0027 - Trim.mp4	10	73.1%	2	0	8
7	2026-02-02 13:54	Imagen	WhatsApp Image 2025-11-20 at 11.11.19 AM (3).jpeg	1	80.6%	1	0	0

Figura 67 Análisis recientes. (Elaboración propia).

9. interpretación de los resultados

9.1 Clase de madurez

PEQUEÑO: Etapa temprana

JOVEN: Etapa media

MADURO: Etapa lista para cosecha

9.2 Métricas Clave

Total de detecciones

Confianza promedio

Agaves únicos (tracking)

10. Exportación de resultados

El sistema permite descargar:

- Archivos CSV
- Resúmenes estadísticos
- Reporte HTML
- JSON completo

11. Buenas practicas

- Utilizar imágenes nítidas
- Evitar sombras excesivas
- Preferir videos estables
- Usar vuelos de dron a altura constante

12. Diagramas del proyecto

12.1 Diagrama de caso de uso

Elementos:

Actor: Usuario

Casos: Analizar Imagen, Grabar Video, Analizar Video, Analizar Dron

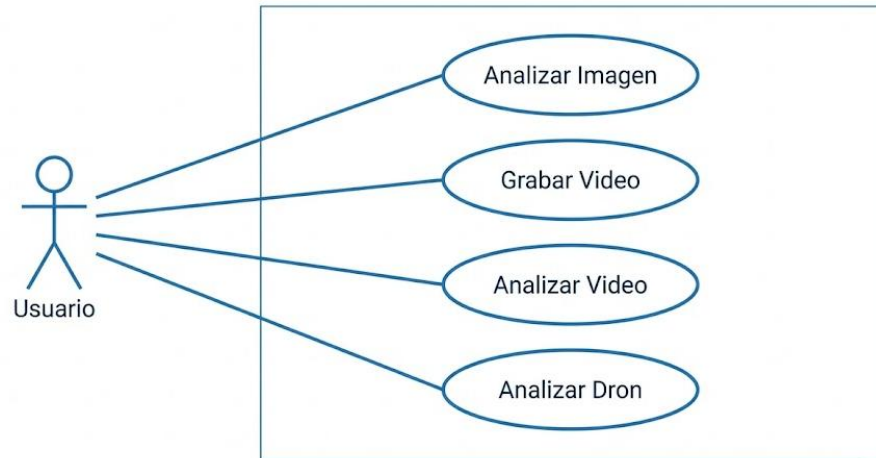


Figura 68 Diagrama de caso de uso. (Elaboración propia).

12.2 Diagrama de Flujo general del sistema

Flujo:

Entrada de datos → Modelo YOLO → Tracking IoU → Métricas → Visualización



Figura 69 Diagrama de flujo general del sistema. (Elaboración propia).

12.3 Diagrama de arquitectura

Componentes:

- Interfaz Streamlit
- Motor de Detección (YOLO)
- Módulo de Tracking
- Módulo de Métricas

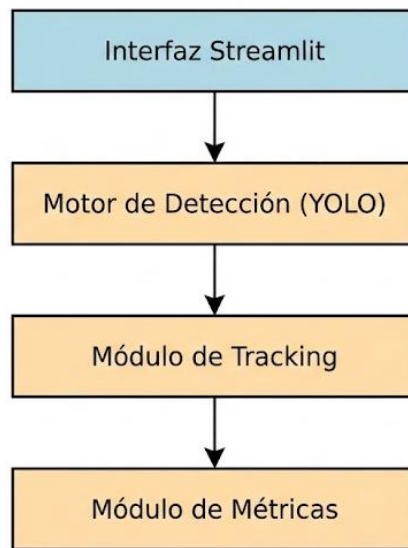


Figura 70 Diagrama de arquitectura. (Elaboración propia).

12.4 Diagrama Entrada-Proceso Salida

Entrada: Imagen / Video / RTSP

Proceso: Detección + Tracking + Clasificación

Salida: Conteo + Gráficas + CSV



Figura 71 Diagrama entrada-proceso-salida

13. Conclusiones

El Detector de Agave Pro V2 es una herramienta robusta, precisa y profesional para el análisis agrícola, permitiendo optimizar la planeación de cosechas, mejorar la toma de decisiones y reducir errores humanos mediante inteligencia artificial.

Autor: Luis Manuel Morales López

Proyecto: Sistema de Conteo y madurez de Agave con Inteligencia Artificial

Versión: Pro V3

ANEXO 1 B MANUAL TÉCNICO

MANUAL TÉCNICO

Sistema: Detector de agave

1. Introducción

El detector de Agave es un sistema computacional avanzado para la detección, conteo y clasificación de plantas de agave mediante técnicas de visión por computadora e inteligencia artificial, implementando una aplicación web interactiva desarrollada en Streamlit.

El sistema utiliza un modelo de detección basado en YOLOv5, entrenado específicamente para poder identificar plantas de agave y clasificar su estado de madurez (germinación, juvenil y maduro). Además, integra un algoritmo de tracking por IoU (Intersection over Union) para el conteo de agaves únicos en video, evitando duplicaciones entre cuadros consecutivos.

Este manual técnico describe detalladamente la arquitectura del sistema, los módulos funcionales, el flujo de procesamiento, las tecnologías empleadas y la operación interna del código, con un enfoque profesional.

2. Objetivo del manual

El presente manual tiene como objetivos:

- Documentar de manera técnica y detallada el funcionamiento del sistema.
- Describir la arquitectura lógica y modular del código fuente.
- Explicar los algoritmos de detección, clasificación y tracking utilizados.
- Facilitar el mantenimiento, escalabilidad y futura mejora del sistema.

- Servir como referencia técnica para evaluaciones académicas o implementación profesional.

3. Descripción general del sistema

El sistema permite analizar agaves a partir de cuatro tipos de entrada:

- Imagen estática
- Transmisión en vivo desde cámara local
- Video cargado por el usuario
- Video o transmisión RTSP desde dron

Para cada modalidad, el sistema realiza:

- Detección automática de agaves
- Clasificación por etapa de madurez
- Conteo total y conteo de agaves únicos (tracking)
- Generación de métricas estadísticas
- Visualización gráfica de resultados
- Exportación de reportes en formato CSV

4. Arquitectura del sistema

4.1 Arquitectura general

El sistema sigue una arquitectura modular basada en el paradigma Entrada – Proceso – Salida (E-P-S):

Entrada: Imágenes, videos, cámara en vivo o stream RTSP

Proceso: Detección YOLO + Tracking IoU + Análisis estadístico

Salida: Videos procesados, métricas, tablas y gráficas



Figura 72 Arquitectura general. (Elaboración propia).

4.2 Arquitectura de Software

- Frontend: Streamlit
- Backend: Python
- Modelo IA: YOLO (PyTorch)
- Procesamiento de imágenes: OpenCV, PIL
- Análisis de datos: Pandas, NumPy
- Visualización: Matplotlib

5. Tecnologías y librerías utilizadas

Librería	Propósito
Streamlit	Interfaz web interactiva
OpenCV	Procesamiento de imágenes y video
PyTorch / YOLO	Detección de objetos
NumPy	Operaciones matemáticas
Pandas	Manejo de datos tabulares
Matplotlib	Gráficas estadísticas
PIL	Manipulación de imágenes
Subprocess	Ejecución del modelo de YOLO

Tabla 10 Tecnologías y librerías utilizadas. (Elaboración propia).

6. Configuración inicial del sistema

6.1 Parámetros globales

- Modelo: best.pt
- Directorio de resultados: runs/detect_app

- Directorio temporal: temp_uploads
- Clases de detección:
 - PEQUEÑO
 - JOVEN
 - MADURO

7. Descripción detallada de los módulos

7.1 Interfaz grafica

La interfaz gráfica se organiza en pestañas (tabs), cada una dedicada a un tipo de entrada específico:

- Imagen estática
- Transmisión en vivo
- Video
- Dashboard

La configuración inicial define el título, icono y diseño de pantalla completa.

7.2 Módulo de detección (YOLO)

El sistema utiliza el script detect.py para ejecutar inferencias sobre los datos de entrada. Este módulo:

- Carga el modelo entrenado
- Procesa imágenes o videos
- Genera Boulding boxes
- Guarda resultados visuales
- Exporta etiquetas en formato TXT con confianza

Formato de salida YOLO:

class_id x_center y_center width height confidence

7.3 Módulo de Tracking por IoU

Para evitar el conteo múltiple de una misma planta en video, se implementa un tracker basado en IoU.

Funcionamiento:

- Cada detección se compara con tracks existentes
- Se calcula la superposición entre bounding boxes
- Si el IoU supera un umbral, se asigna el mismo ID
- Si no, se crea un nuevo ID

Este enfoque permite:

- Identificar agaves únicos
- Mantener persistencia entre frames
- Mejora la confiabilidad del conteo

7.4 Clasificación de madurez

Cada agave detectado se clasifica según su etapa de desarrollo:

- Germinación: Es etapa temprana
- Juvenil: Es una etapa intermedia
- Maduro: Es la etapa final

7.5 Procesamiento estático

Se generan métricas como las siguientes:

- Total de detecciones
- Total de agaves únicos
- Conteo por etapa de madurez
- Confianza promedio
- Distribución de confianza

Los datos se estructuran en DataFrames de Pandas para facilitar su análisis

7.6 Visualización de resultados

El sistema genera:

- Histogramas de confianza
- Graficas de barras por etapas de madurez
- Tablas de resumen descargables

Estas visualizaciones permiten evaluar la calidad del modelo y el estado del cultivo

8. Flujo de operaciones del sistema

- El usuario selecciona tipo de entrada
- Se carga o captura el medio
- Se ejecuta la detección por medio de YOLO
- Se procesan las etiquetas
- Se aplica el tracking si es video
- Se calcula las métricas
- Se muestran resultados
- Se generan reportes descargables

9. Gestión de archivos y resultados

- El sistema maneja automáticamente:
- Limpieza de resultados anteriores
- Conversión de videos anteriores
- Organización de carpetas por ejecución

Esto garantiza el orden y reproducibilidad

10. Consideración de rendimiento

- El tracking IoU es eficiente y ligero
- El procesamiento depende de GPU/CPU disponible
- Se recomienda resolución HD para mejor precisión

11. Escalabilidad y mejorar futuras

Integración de Deep SORT o ByteTrack

Base de datos histórica de parcelas

Exportación a PDF

Dashboard agrícola

Integración con sensores IoT

12. Conclusión

El sistema de Detector de Agave es una solución robusta, modular y escalable para la agricultura de precisión, que combina inteligencia artificial, análisis estadístico y visualización interactiva, permitiendo una toma de decisiones informada y basada en datos confiables.

Este manual técnico documenta de forma integral su funcionamiento interno, garantizando su comprensión, evaluación académica y evolución futura.

Autor: Luis Manuel

Carrera: Ingeniería en Sistemas Computacionales

Área: Inteligencia Artificial y Visión por Computadora

ANEXO 2 A



Figura 73 Visita con SECAMPO (Elaboración propia)



Figura 74 Visita con SECAMPO (Elaboración propia)

ANEXO 2 B



Figura 75 Reconocimiento al Tecnológico.



Figura 76 Reconocimiento de Bronce

ANEXO 2 C



Figura 77 Reconocimiento por la participación en el ATICTAC



Figura 78 Constancia por la partición en el ATITAC.