

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE TIANGUISTENCO

DIVISIÓN DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES

SISTEMA DETECCIÓN DE INCENDIOS FORESTALES
MEDIANTE INTELIGENCIA ARTIFICIAL

TESIS

Para obtener el título de
INGENIERO EN SISTEMAS COMPUTACIONALES

Presenta:

BRAYAN ORDOÑEZ GONZALEZ

Asesor(a):

MTRA. MARÍA GUADALUPE PINEDA ARIZMENDI

Santiago Tianguistenco, Estado de México, Febrero 2026

HOJA DE PERMISO DE DATOS

Yo, Brayan Ordoñez Gonzalez, titular de los derechos de autor del conjunto de datos y código asociado al proyecto, Sistema detección de incendios forestales mediante 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/motobrayan/deteccion-de-incendios-2.git>

Brayan Ordoñez Gonzalez

RESUMEN

Los incendios forestales constituyen una amenaza constante a nivel mundial. Su impacto afecta la biodiversidad, la economía y la salud pública. En México, la Comisión Nacional Forestal (CONAFOR) reportó en 2023 una superficie afectada superior a 560,000 hectáreas (Comisión Nacional Forestal, 2025). Los métodos tradicionales de detección, como el patrullaje terrestre o la observación satelital, presentan limitaciones en la rapidez de respuesta. Con frecuencia, los incendios se identifican cuando ya alcanzan una extensión considerable.

El presente trabajo tiene como propósito desarrollar e implementar un sistema de detección temprana de incendios forestales. La metodología se basa en el modelo CRISP-DM e incluye tres etapas. La primera consiste en la creación de un conjunto de imágenes con y sin presencia de fuego. La segunda contempla el diseño, entrenamiento y evaluación de arquitecturas de redes neuronales convolucionales (CNN), implementando específicamente el modelo YOLOv5, para la detección y clasificación en imágenes. La tercera corresponde al desarrollo de una plataforma web que integre el modelo de inteligencia artificial y automatice el envío de alertas mediante una API.

Se espera obtener un modelo de CNN basado en la arquitectura YOLOv5 con una precisión aproximada al 95 % y un tiempo de inferencia menor a dos segundos por imagen. Además, se proyecta el desarrollo de un prototipo funcional que permita emitir notificaciones automáticas ante la detección de incendios.

ABSTRACT

Forest fires pose a constant threat worldwide. Their impact affects biodiversity, the economy, and public health. In Mexico, the National Forestry Commission (CONAFOR) reported over 560,000 hectares affected in 2023 (National Forestry Commission, 2025). Traditional detection methods, such as ground patrols or satellite observation, have limitations in response time. Fires are often identified only after they have already reached a considerable extent.

This work aims to develop and implement an early forest fire detection system. The methodology is based on the CRISP-DM model and includes three stages. The first consists of creating a set of images with and without the presence of fire. The second involves the design, training, and evaluation of convolutional neural network (CNN) architectures, specifically implementing the YOLOv5 model, for image detection and classification. The third stage consists of developing a web platform that integrates the artificial intelligence model and automates the sending of alerts via an API.

The goal is to obtain a CNN model based on the YOLOv5 architecture with approximately 95% accuracy and an inference time of less than two seconds per image. Additionally, the development of a functional prototype is planned to enable automatic notifications upon fire detection.

AGRADECIMIENTOS

En primer lugar, agradezco a Dios por darme fuerza y esperanza para lograr meta que me proponga.

A mi director de tesis, Mtra. María Guadalupe Pineda Arizmendi, por su orientación, paciencia y valiosas contribuciones académicas, las cuales fueron fundamentales para la realización de esta investigación.

A los miembros del comité de tesis, Ing. Nayeli Zapata Álvarez y al Ing. Genaro Chávez Leónles agradezco sus comentarios, sugerencias y el tiempo dedicado a revisar este trabajo, lo que permitió enriquecerlo significativamente.

A la Universidad Tecnológico de Estudios Superiores de Tianguistenco, por brindarme los recursos, espacios y condiciones necesarias para llevar a cabo esta investigación.

Al Ing. Luis Valdez Alcántara, por sus consejos académicos y su apoyo en momentos clave.

En el plano personal, agradezco profundamente a mi familia: Sandra Gonzalez Guerrero, Ignacio Ordoñez Blancas y Pedro Ordoñez Gonzalez, por su apoyo emocional incondicional, su paciencia y su comprensión.

A mis amigos cercanos, en especial a Luis Manuel Morales López, Deivid Irineo Rojas Zetina, Aelin Eliuth López Hernández, Ricardo Alejandro Sánchez Gonzalez, Antonio Herrera Pérez, Fidel Oswaldo Castro y a la familia Villegas Zamorano gracias por su ánimo, por escucharme y por estar presentes en los momentos difíciles.

También deseo agradecer a las personas que participaron en este estudio protección civil y brigadistas de probosque del ayuntamiento de Santiago Tianguistenco, por su disposición y por compartir su tiempo y conocimiento, lo cual fue invaluable para los resultados obtenidos.

DEDICATORIA

Dedicada a mi familia: Sandra Gonzalez Guerrero, Ignacio Ordoñez Blancas

Y a mi abuela María Guerrero Pérez por apoyar en la carrera.

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
AGRADECIMIENTOS	IX
ÍNDICE GENERAL	XI
ÍNDICE DE TABLAS	XIII
ÍNDICE DE FIGURAS	XIII
INTRODUCCIÓN	15
1. CAPÍTULO 1	17
1.2 MARCO TEORICO	21
1.2.1 <i>Incendios Forestales</i>	21
Detección de incendios forestales	22
1.2.2 Tipos de Incendio	23
Factores Ambientales de Propagación	24
Impacto de los Incendios Forestales	25
Factores meteorológicos en incendios forestales	26
1.2.3 Estaciones Meteorológicas	27
Sensores Meteorológicos	28
1.2.4 Mecanismos de detección	29
1.2.5 Sistemas embebidos	29
1.2.6 Internet de las cosas	30
1.2.7 Almacenamiento de datos	31
1.2.8 Inteligencia Artificial	31
1.2.9 Python	36
Google Colab	37
Google sheets	38
1.2.10 Procesamiento de Imágenes	38
Segmentación de Imágenes	38
Imágenes térmicas	39
Dron mini	39
1.2.11 Técnicas de minería de datos	40

2.	CAPÍTULO 2	46
2.1	METODOLOGÍA	46
3	CAPÍTULO 3	55
3.4	Validación de Sensores y Alertas	62
3.5	Evaluación del Modelo de Visión Artificial	63
3.6	Pruebas de la red neuronal con yoloV5	63
3.7	Pruebas del entrenamiento de la red neuronal	64
	Prueba 1	65
	Prueba 2	66
	Prueba 3	67
	Prueba 4	68
4	CAPITULO 4	69
4.1	CONCLUSIONES	69
4.5	RECOMENDACIONES	70
5	FUENTES BIBLIOGRÁFICAS	71
	Bibliografía	71
6	ANEXOS	75
	Anexo 1	75
	Anexo 2	83
	Anexo 3	89
	ANEXO 4	90
	ANEXO 5	91
	ANEXO 6	92
	Anexo 7	93
	Anexo 8	94
	Anexo 9	97

ÍNDICE DE TABLAS

Tabla 1 descripción de los sensores que se utilizaron para la estación (elaboracion propia)	76
Tabla 2descripcion de los lenguajes de programacion que se utilizaron (elaboraxion propia)	77
Tabla 3 cronograma de actividades	89

ÍNDICE DE FIGURAS

INTRODUCCIÓN

Los incendios forestales se establecen como un fenómeno de gran relevancia ambiental y social, debido a los daños que ocasionan en los ecosistemas, la biodiversidad y las actividades humanas. Estos eventos pueden originarse por fenómenos naturales, por la acción directa o indirecta del ser humano, dicha propagación depende de múltiples condiciones como las condiciones climáticas, la vegetación y la topografía del terreno. En México, los incendios forestales representan un problema recurrente que requiere estrategias de prevención, monitoreo y detección temprana para reducir sus impactos negativos (Comisión Nacional Forestal, 2025).

Diversos estudios han indicado que los incendios forestales generan impactos significativos en los campos social, ambiental y económico, teniendo en consecuencia el deterioro de la calidad del aire, reduciendo la disponibilidad de recursos ambientales y protección de las comunidades cercanas a las zonas forestales. Del mismo modo, el análisis de sus causas y efectos resulta esencial para el diseño de sistemas que contribuyan a su detección y control oportuno (Aguilera Sánchez, 2015). El presente proyecto se enfoca en optimizar dos procesos específicos en el municipio de Tianguistenco: la detección temprana de condiciones propicias para incendios (temperaturas altas y baja humedad) y la clasificación automática del tipo de incendio forestal (superficie, copa o subsuelo) una vez iniciado. En este contexto, la inteligencia artificial se consolida como una herramienta relevante para el procesamiento automático de información y la resolución de problemas complejos. A través del aprendizaje automático y en especial del aprendizaje profundo, siendo factible el desarrollar sistemas capaces de identificar diferentes muestras y características relevantes gracias a grandes conjuntos de datos, como imágenes digitales. Estas técnicas permiten automatizar procesos de clasificación y reconocimiento, reduciendo la intervención humana y aumentando la eficiencia en tareas de análisis visual (Soria Olivas, Gamero Cruz, Castillo Caballero, & Cano Michelen, 2023). Para ello, en este trabajo se ha desarrollado una estación meteorológica equipada con sensores de humedad y temperatura, cuyos datos se almacenan en una hoja de cálculo de Google Sheets para su monitoreo continuo. Esta estación, junto con

un sistema de red neuronal artificial, permitirá emitir mensajes preventivos ante condiciones de riesgo y clasificar los incendios forestales registrados en el área. Por este medio, la inteligencia artificial se consolida como un apoyo crucial relevante para el procesamiento automático de información y en la resolución de problemas complejos. A través de aprendizaje automático y en especial del aprendizaje profundo, siendo factible el desarrollar sistemas capaces de identificar similitudes y características relevantes a partir de grandes conjuntos de datos, como imágenes digitales. Estas técnicas permiten automatizar procesos de clasificación y reconocimiento, reduciendo la intervención humana y aumentando la eficiencia en tareas de análisis visual (Soria Olivas, Gamero Cruz, Castillo Caballero, & Cano Michelena, 2023). Para ello, en este trabajo se ha desarrollado una estación meteorológica equipada con sensores de humedad y temperatura, cuyos datos se almacenan en una hoja de cálculo de la aplicación Google Sheets para su monitoreo continuo. Esta estación, junto con un sistema de red neuronal artificial, permitirá emitir mensajes preventivos ante condiciones de riesgo y clasificar los incendios forestales.

En el ámbito del aprendizaje profundo y las redes neuronales convolucionales se han posicionado como las arquitecturas más utilizadas en aplicaciones de visión por computadora, gracias a su exactitud para extraer características especiales y visuales presentes en las imágenes. Estas redes están formadas por diferentes tipos de capas que permiten el procesamiento gradual de la información visual, lo que particularmente se vuelve adecuada para tareas de detección y clasificación de imágenes relacionadas con fenómenos naturales, como el fuego (López Sotelo, 2023).

En este proyecto, propone una implementación de una red neuronal convolucional para clasificar automáticamente los tipos de incendios forestales (superficie, copa o subsuelo) a partir de imágenes capturadas en el municipio de Tianguistenco. El funcionamiento consiste en entrenar la red con un conjunto de imágenes etiquetadas de incendios, de modo que la red aprenda a reconocer patrones visuales característicos de cada tipo de fuego.

1. CAPÍTULO 1

1.1 ANTECEDENTES

Los incendios forestales se han posicionado como uno de los principales desafíos ambientales a escala mundial, causando graves problemas ante la biodiversidad, los recursos naturales y las comunidades cercanas a zonas boscosas. En los últimos años, el aumento de temperatura, la sequía y las condiciones climáticas extremas en las que se incrementa la frecuencia e intensidad de dichos eventos, lo que ha impulsado el desarrollo de sistemas tecnológicos orientados a su detección y prevención temprana (Comisión Nacional Forestal, 2025).

Frente a este panorama, la Inteligencia Artificial (IA), en específico el procesamiento digital de imágenes y las redes neuronales (CNN), se ha surgido como una herramienta eficaz para identificar visualmente indicios de humo o fuego en tiempo real.

El autor Aliaga Casaletti, (2020) propone un sistema que ayude a la precaución de incendios forestales utilizando hardware de bajo consumo computación en la nube y software de visión por computadora, a su vez realizó una entrevista con el cuerpo de bomberos para aclarar y explicar cuál es el método que ellos utilizan para la prevención de incendios forestales. La metodología que se implementó para su proyecto fue ágil. A sí mismo para obtener datos realizó una encuesta al cuerpo de bomberos y con ayuda de una cámara inteligente fue recopilando imágenes de incendios creando su conjunto de datos y etiquetando las imágenes como fuego y humo en Robo Flow. Con todo esto logro desarrollar su proyecto, como lo es detectar los incendios mediante las imágenes previamente etiquetadas, la emisión de alertas en tiempo real, obtener la ubicación de los cuerpos de bomberos de manera geográfica, visualización de cámaras con ayuda de una página web, gestión de usuarios, cálculos sobre las rutas más cercanas hacia donde se encuentra el incendio y la visualización de los brigadistas que estén cercanos donde se suscita el incendio.

Las propuestas a futuro que propone son mejorar la precisión en cuanto a la recolección de datos, también indica que las cámaras que se utilizan para el monitoreo quedan fijas por lo que propone anexar vehículos aéreos no tripulados mejorando la visión del área.

En otro estudio Lazo La Rosa, Leandro (2024) propuso un sistema de detección de incendios

forestales mediante redes neuronales convolucionales, asimismo la creación de máscaras mediante una imagen espectral unificándolo con la imagen original para visualizar de manera óptima los incendios provocados con ayuda de la aplicación QGIS y con imágenes satelitales para que también pueda analizar el área afectada por los incendios provocados. La metodología se basa en investigación aplicada con un enfoque cuantitativo y diseño experimental. Para la captación de datos utilizó un conjunto de datos de imágenes satelitales, uno del satélite LandSat 8 (con 227 imágenes) y el segundo Sentinel 2 (con 108 imágenes) obtenidas de una página web (SERFOR). Al realizar las pruebas y el entrenamiento de las imágenes utilizó un modelo llamado UNet, menciona que esta es una red pre entrenada con otra arquitectura denominada VGG 11, en el primer conjunto de datos (227 imágenes) lo entreno con 350 y 150 épocas, aunque explica que al entrenarla con 350 épocas no segmentó ninguna de las áreas, mientras que en el de 150 épocas llegó a segmentar las áreas, pero no son donde se encuentran las áreas incendiadas. Con este resultado da a entender que esto no se debe necesariamente por las épocas en la fueron procesadas las imágenes y tampoco la red neuronal, sino que fue por la página en la que encontró su conjunto de datos que fue a causa del filtro de las fechas y que no podía asegurar con certeza si las imágenes coincidían con los incendios que ya se habían notificado. Así que con todo el trabajo que realiza interpreta que de sus trabajos a futuro plantea que con las imágenes que el capturo se pueda realizar un conjunto de datos para que próximos trabajos puedan conseguir de manera rápida imágenes para el entrenamiento de la red neuronal.

En otro estudio, Thedy Cornejo (2024), creó un vehículo con conducción asistida con instalaciones de cámaras térmicas siendo eficiente aplicando la conducción automática, también creando una red neuronal con el propósito de identificar múltiples objetos y la detección de personas en un entorno real. Utiliza una metodología cuantitativa en la que ilustra como es que fue realizando su proyecto desde como empezaría la calibración de la cámara, siguiendo con la detección de objetos, implementación STT y finalmente concluyendo con la implementación MTT. Para los datos los recopiló de GitHub y de dos videos en YouTube. Para que la cámara infra roja pueda detectar los objetos con sensores como cámaras RGB o LIDARS, así también utilizando MATLAB 2018 para calibrar la cámara. El porcentaje que comenta de resultados es el 49% de mAP de la red neuronal.

También menciona otros valores como probabilidad de detección $PD = 0.8$ (80%), clutter esperado $\lambda_k = 0.2$ (20%), y una simulación de detecciones 100% perfectas como caso base. Así que con todos los avances que obtuvo propuso como trabajos a futuro el instalar una segunda cámara al vehículo para que entregue más profundidad a las imágenes, también propone crear un conjunto de datos para que pueda tener detecciones más precisas.

Asimismo, León Castillo (2024), desarrolló un sistema como implementación para la detección y prevención de incendios forestales por inteligencia artificial basado en visión por computadora. Utilizando la metodología de investigación aplicada y cuantitativo, gracias a los datos recolectados se buscaron las herramientas con las que se tenían que trabajar como lo son cámaras IP, vehículo aéreos no tripulados con cámaras RGB e imágenes satelitales que se extrajeron de Sentinel-2 y Landsat 8, así también equipo de cómputo con unidades GPU para el entrenamiento de los modelos en donde se utilizaron los programas como Python, librerías para el procesamiento de imágenes y etiquetado como TensorFlow Keras y PyTorch y por ultimo QGIS para el mapeo de las zonas vulnerables. El desarrollo fue automatizando datos geoespaciales y climáticos mejorando la eficacia de los programas de gestión ambiental, a su vez las imágenes que se recopilaron fueron etiquetadas de manera manual para tener más precisión en detectar humo, fuego, vegetación afectada y zonas críticas, para después crear una red neuronal convolucional basada en la arquitectura de MobileNetV2 y clasificándola en dos clases como “sin riesgo” y “riesgo potencial”. Al entrenar la red neuronal obtuvo las métricas de precisión de tres lugares en los que puso a prueba su red en el primer caso obtuvo métricas como precisión del 92%, un recall del 90% y un F1-score del 91%, en el segundo caso precisión del 89%, recall del 86% y F1-score de 87,5% y en el tercer caso precisión del 87%, recall del 84% y un F1-score de 85,4%. Con esto pudo obtener los trabajos a futuro como lo son la ampliación del conjunto de datos y aumentar la información de los datos meteorológicos en tiempo real y que no este proyecto también de se debe probar en diferentes ecosistemas.

Por último, Fernández (2024), diseñó un sistema para la predicción de incendios forestales y la predicción de la propagación de este, con ayuda de cámaras inteligentes, sensores y

algoritmos de inteligencia artificial, estos ubicados en lugares estratégicos como en torres de vigilancia para mayor visibilidad del área. Utiliza Metodología experimental de enfoque cuantitativo. Para la recolección de datos que se utilizó fue mediante cámaras de red pública, específicamente HPWREN y ALERTWildfire estas son cámaras que son propiedad del gobierno de los estados unidos americanos, gracias a ello se obtuvieron diversos conjuntos de datos como “SmokeFrames” en el cual contiene un aproximado de 50 mil imágenes de alerta de incendio, “nemo” esta parte de los incendios ya registrados en los que se muestran evidencias de diferentes densidades de humo, otra que se adjuntó de “fuego”, donde se muestra los inicios de incendio a pesar de que cuenta con 8,500 registros solo están disponible 1,661 imágenes públicas. Con toda la información recolectada se creó su base de datos, al haber entrenado su red neuronal para mostrar los resultados que obtuvo utilizo las métricas de precisión, en los que sometió a prueba los tres conjuntos de datos que utilizó para este proyecto: Fuego: Precision 0.69, Recall 0.86, F1-Score 0.76. SmokeFrames: Precision 0.90, Recall 0.83, F1-Score 0.86 y Nemo: Precision 0.86, Recall 0.93, F1-Score 0.90. Las propuestas a futuro que el considera para este proyecto son explorar nuevas arquitecturas, la que el menciona se llama “transformers” en la que puede tener detecciones más precisas y capturar relaciones más complejas entre cuadros consecutivos, ampliar su conjunto de datos, este no pudo entrenar su sistema con las imágenes tan limitadas que encontró además de que fueran más imágenes con diferentes zonas geográficas, vegetaciones y condiciones climáticas y mejorar las etapas de detección para reducir errores o inconsistencias y tener mejores resultados.

En conjunto, estas investigaciones reflejan una tendencia creciente hacia la integración de IA, visión artificial, sensores IoT y sistemas aéreos autónomos en la detección temprana de incendios forestales. No obstante, aún persisten desafíos importantes, como la necesidad de optimizar el consumo de recursos computacionales, ampliar la cobertura territorial y mejorar la respuesta automatizada ante emergencias.

1.2 MARCO TEORICO

El presente capítulo se tiene como propósito fundamentar conceptualmente el cómo desarrollar un sistema de detección de incendios forestales mediante inteligencia artificial, apoyado por una estación meteorológica y una plataforma web interactiva. Para ello, se abordan los aspectos sociales relacionados con los incendios forestales, los fundamentos computacionales de la inteligencia artificial y los sistemas embebidos, así como el enfoque metodológico empleado para el diseño y desarrollo del proyecto. Esta base teórica permite sustentar científicamente la propuesta tecnológica y contextualizar su aplicación dentro del ámbito académico y ambiental.

1.2.1 Incendios Forestales

Los incendios forestales se constituyen como procesos de combustión de material orgánico no controlada de vegetación, deteriorando ecosistemas terrestres y zonas cercanas a comunidades humanas (Fernández, 2024), Asimismo un incendio se crea de manera natural o con algún propósito en lugares terrestres subterráneo o aéreos en los que se ven afectadas como lo son: arboles, matorrales, humedales, raíces zacate entre otros (Pausas, 2012). En el ámbito ambiental, los incendios causan percances masivos en los ecosistemas, afectando severamente la biodiversidad al aniquilando hábitats y especies. Además, contribuir significativamente al cambio climático, ya que, mediante la emisión de gases de efecto invernadero, como CO₂ (Dióxido de Carbono), CH₄ (Metano) y N₂O (Óxido Nitroso). A su vez degradando significativamente el suelo del área, acelerando la regeneración natural de los bosques (Fernández, 2024).

Detección de incendios forestales

En Mexico la COMISION NACIONAL FORESTAL, implementa varios sistemas de detección de incendios, los cuales pueden considerarse métodos tradicionales. En su página oficial ofrece tres opciones para la prevención de incendios: una de ellas es la captura de datos con imágenes satelitales en tiempo real, con la que hacen una predicción de dónde posiblemente se puede generar un incendio. El segundo apartado es similar, también basado en capturas satelitales, pero esta muestra dónde ya hay incendios que se han propagado o están iniciando. Y en el tercero trabaja en conjunto con CONAGUA (COMISION NACIONAL DEL AGUA), donde está a cargo del cambio climático en general, pero para este caso solo se muestran los datos climatológicos en la que esta alerta si puede surgir alguna alteración para prevenir y alertar tanto a las autoridades correspondientes o a los civiles (Comisión Nacional Forestal, 2025). Otros métodos del tipo tecnológico que son prácticos para una detección temprana de los incendios son con cámaras de monitoreo en las que ya vengán implementadas con un algoritmo de inteligencia artificial con las que se pueden identificar las columnas de humo con precisión, instaladas en torres de vigilancia o en vehículos tripulados. También está la disponibilidad de utilizar sensores ambientales para obtener datos como lo son de temperatura, humedad y la velocidad del viento (Fernández, 2024).

1.2.2 Tipos de Incendio

De acuerdo con la Comisión Nacional Forestal (Comisión Nacional Forestal, 2025) los incendios forestales son un recurso que también pueden ayudar al ecosistema, una perturbación en la que puede mejorar el paisaje y mejorar la calidad de la vegetación, pero si no se supervisa o se maneja de manera adecuada esta puede descontrolarse y propagarse y ocasionando tres tipos de incendios:

Superficiales: Se ocasiona en el terreno de manera horizontal alcanzando una altura de metro y medio (ejemplo en la Ilustración 1), afectando vegetación como lo son pastizales, hojas, ramas secas, troncos, arbustos pequeños entre otros.



Ilustración 1. Incendios forestales de tipo superficie (Comisión Nacional Forestal 2025)

Subterráneos: Esta se generará únicamente si se comienza un incendio superficial, esta al ser intensa se puede filtrar al subsuelo (ejemplo en la ilustración 2) y al igual se puede propagar debido a que puede avivarse con materia orgánica y con raíces, en la mayoría de estos incendios no se encuentra mucho fuego y se encuentra más el humo.



Ilustración 2 incendios forestales de tipo subterráneo (Comisión Nacional Forestal 2025).

Aéreo o de copa: Este tipo de incendio es el más peligroso de los tres, se origina en suelo haciendo que el fuego vaya de manera vertical alcanzando la vegetación de los árboles (ejemplo en la Ilustración 3), haciendo más fácil su propagación y que el poderlo controlar sea aún más complicado.



Ilustración 3 Incendios forestales de tipo copa o aéreo (Comisión Nacional Forestal 2025).

Estos tipos de incendios forestales son grandes factores de daño en los ecosistemas, que debido a las condiciones del lugar como sequías extremas pueden ser más severos como en el caso en chihuahua en 1998, donde los incendios más severos que registraron fueron causados por incendios de copa (Ruiz & Blanco, 2004).

Factores Ambientales de Propagación

En el libro ¿Qué sabemos de? incendios forestales, una introducción a la ecología del fuego, se hace mención que los incendios son provocados en su mayoría por “incendios prescritos” que es la denominación a los incendios que son provocados por la quema de vegetación del lugar y que principalmente es ocasionado por el ser humano (Pausas, 2012). mientras que en el libro “El régimen jurídico de los incendios forestales”, explica que también los incendios forestales también son provocados también por la naturaleza, ocasionadas por el clima o la vegetación (Aguilera Sánchez, 2015), Y se describen los tipos de incendios en los casos que pueden ser ocasionados.

Antropogénicas: Se definen por ser los incendios que el ser humano los inicia principalmente por el área agrícola en las que se busca un beneficio (reducción de combustible, renovación de pastizales o vegetación, modificación de un lugar para traer otro, control de plagas etc.). La autora Aguilera Sánchez menciona que también se debe a una mala práctica de la

población, en la que no es consiente que por mal manejo en lugares recreativos en lugares forestales.

Naturales: Como el nombre lo indica estas con ocasionadas por algún factor climático que no está vinculado al ser humano que puede suceder de manera imprevista. Algunas son ocasionadas por el contenido de CO₂ que puede haber en alguna zona y de algunos otros gases en las que puede aumentar la temperatura. Por otra parte, la vegetación también juega un papel crucial en los incendios forestales teniendo en cuenta que principalmente es el material inicial para provocarlo, estos son blanco fácil en una tormenta eléctrica, por árboles de grandes alturas que pueden atraer rayos y ocasionar el incendio, aunque sea probable que pueda pasar en realidad puede ser mínima a comparación de otras (Aguilera Sánchez, 2015).

Impacto de los Incendios Forestales

En el libro “Sintiendo el calor: Perspectivas internacionales sobre la prevención de la ignición de incendios forestales”, abarca mucha información en la que relata cómo es que en otros países han traído consecuencias devastadoras que incluso pueden llegar a categorizarse como irreparables al iniciarse un incendio en países importantes como Estados Unidos de America aunque no especifica el lugar donde se suscitó, reporta que hay un incendio que desde la década de los setentas se ha iniciado un incendio y hasta la fecha actual no ha sido extinto dejando muchas perdidas (STANLEY, March, & Ogloff, 2025). A continuación, se presentarán la clasificación de impactos de los incendios clasificándolas de la siguiente manera:

Ambiental: en general representa toda la vegetación que existe en el lugar y la biodiversidad de animales que habitan en áreas forestales tan solo en Mexico se han registrado muchos incendios, perdiendo varias hectáreas en 2023 obtuvo un registro final de 1,047,493 hectáreas donde se perdieron millones de árboles y vegetación del suelo, en el 2024 fue de 1,672,216 hectáreas. Aunque esto a veces no signifique que esto sea malo a causa de que existen áreas en donde se necesita incendiarse para que este pueda persistir en el paisaje a esto se adaptó a las altas temperaturas que genera el incendio y a que después puede hacer que florezca su follaje de forma eficaz (Comisión Nacional Forestal, 2025).

Social: Los incendios generan un peligro para la sociedad, debido de que pueden avanzar a

lugares rurales donde familias junto con sus animales pueden ser un blanco fácil del incendio se no evacuan a tiempo, tan solo a nivel mundial se tiene el registro de que en el 2018 murieron 273 personas en donde estados unidos obtuvo más muertes por incendios forestales con 108 personas fallecidas esto también llega a producir enfermedades en las personas como el inhalar el humo de forma directa debido que en el aire viajan partículas que afectan a los pulmones. Así también como lugares ganaderos dado que los incendios pueden afectar el suelo donde se concentraba la producción de alimentos o de la agricultura.

Económico: esto no necesariamente debe inclinarse a pérdidas materiales como casas, estructuras, edificios etc. Sino también una perdida en donde también se debe reforestar los lugares afectados puesto que son miles de hectáreas las que se ven perjudicadas. En el libro “El régimen jurídico de los incendios forestales” (Aguilera Sánchez, 2015), hace inca pie en que en España hacen el conteo de cuanta madera fue afectada por el incendio este registro la han hecho desde hace 50 años con lo que llevan una pérdida de 28.1 millones de euros contando con el gasto de operaciones de extinción contra el fuego.

Factores meteorológicos en incendios forestales

Para que un incendio forestal se propicie o se propague, se debe tomar en cuenta que tenemos que complementar tres cosas importantes, la calidad del viento, temperatura del lugar y la humedad que hay en el entorno. La temperatura es la más importante a mayor temperatura mayor es la perdida de humedad y gran sequia de la vegetación además de que este presente el viento se vuelve más seco provocando que se absorba el vapor de la vegetación, además de que controla la dirección en la que va el fuego y la velocidad siendo un apoyo que actúa también como un fluido continuo aportando más oxígeno al incendio así haciendo que sea complicado la extinción del fuego (Rodríguez Rivas, 2010).

Otro factor que es importante ante los incendios forestales es la evapotranspiración que es la evaporación del agua que se encontraba en el subsuelo y transpiración de plantas o cualquier otra vegetación y que van directo a la atmosfera, esto es crucial a cusa también de que interfiere en el viento y la humedad en lo que al iniciar el incendio pueden conocerse como vientos saturados donde la humedad está presente y puede también ser un apoyo en el

momento de disminuir el incendio pero depende mucho en el área que se ocasiona el fuego al estar en total sequía en cuanto a vegetación y suelo y en donde el sol registre temperaturas altas, de lo contrario esta corriente de viento podría ser una gran ayuda para combatir el incendio forestal (Rodríguez Rivas, 2010).

1.2.3 Estaciones Meteorológicas

Para medir las variables (temperatura, humedad y viento) que son las encargadas de provocar o continuar un incendio forestal se hace mediante una estación meteorológica, que es un dispositivo en el que se guardan diversos materiales para la medición de diversos elementos meteorológicos (Brenes & Saborío, 1995), el cual existen diversos tipos de estaciones y se clasifican dependiendo de los instrumentos como se muestra en la Tabla 1:

Tabla 1 Tipo de Estaciones

Estacion de primer orden	También llamado clase A, es la más equipada, puede medir temperatura, precipitación, nubosidad, viento, presión atmosférica, humedad, evaporación etc.
Estacion de segundo orden	La clase B, esta estación se encarga de solo medir temperatura y precipitación
Estacion de tercer orden	Conocida igualmente como clase C, esta estación solo es encargada de medir la precipitación

Así que la importancia de una estación meteorológica ante un incendio forestal es medir las variables del viento, humedad y temperatura antes y durante del mismo así que para observar los datos de la estación se pueden realizar mediante dos métodos (sensorial e instrumental) que la sensorial se puede medir mediante los sentidos en que prevalece la vista y en la instrumental es a partir de sensores o herramientas especializadas en el lugar, estas deben ser de manera síncrona y sistemáticas para que se pueda obtener de manera muy exacta las métricas que nos pueda dar la estación (Brenes & Saborío, 1995).

Para este proyecto se utilizará la estación de segundo orden, es la que se familiariza con la

estación necesaria para medir los tres tipos de variantes para una detección de incendio forestal, agregando la parte de medición de humedad y viento. También se utilizará la observación instrumental que es más precisa para la medición de las variables necesarias para la prevención y detección de los incendios forestales, además de poder obtener un registro de las mediciones a lo largo del tiempo y de diferentes periodos.

Sensores Meteorológicos

Los sensores meteorológicos son el completo sistema de medición que se integra en las estaciones. Pueden tener diferentes tipos de sensores según la cantidad y tipo de variables que se quieran medir y monitorear. Dichos sensores tienen como tarea el capturar información y mandarlos a un almacenador de datos para que después pueda ser guardada y después monitorear y evaluar los datos obtenidos dependiendo al tiempo al que se requiere.

El uso de sensores meteorológicos también es fundamental en el desarrollo de sistemas automatizados de observación ambiental, donde los datos recolectados permiten evaluar condiciones climáticas, detectar cambios en el ambiente y apoyar sistemas de alerta temprana ante fenómenos naturales como lo puede ser en este caso incendios forestales.

Sensor DHT11: Este es un sensor pequeño (como se muestra en la siguiente ilustración) que tiene una cubierta plástica en forma de rejilla en la que se le permite pasar el aire, además de tener 3 pines en donde se entra energía, tierra y donde se envían los datos que captura el mismo sensor (Rodríguez, León, Bernal, & Gómez, 2025).

Anemómetro: Dispositivo que ayuda a la medición del viento gracias a la ayuda de copas (como se muestra en la ilustración 5) que giran con fuerza a la fuerza del viento haciendo que roten, en este caso para que se pueda medir la velocidad del viento son mediante las pulsaciones electromagnéticas, mediante un imán de neodimio que gira en el eje de las copas que pasa cerca de un Reed switch haciendo que cada que generan un pulso se realiza una ecuación para medir el viento que hay en el área (Mott & Samaniego, 1996).

1.2.4 Mecanismos de detección

Los mecanismos de detección de incendios comprenden un conjunto de técnicas que permiten identificar la presencia de fuego a partir de variables físicas y químicas, como temperatura, radiación infrarroja, gases combustibles y análisis visual. La combinación de múltiples mecanismos incrementa la confiabilidad del sistema y reduce la generación de falsas alarmas (Singh & Kumar, 2025). En el caso de los sistemas basados en inteligencia artificial, el análisis visual mediante procesamiento de imágenes se ha convertido en una de las técnicas más utilizadas, permitiendo identificar características propias del fuego y el humo dentro de una escena capturada por cámaras o sensores ópticos.

1.2.5 Sistemas embebidos

Este sistema se basa principalmente en un microcontrolador para que pueda realizar una o más tareas, pero este no está diseñado para ser programado por el usuario final. A diferencia de una computadora tradicional, donde el usuario tenga la libertad de instalar y reemplazar programas para cambiar su propósito (como pasar de escribir un texto a jugar), un sistema embebido solo tiene una tarea específica definida por su desarrollador. Su implementación permite procesar datos provenientes de distintos dispositivos y ejecutar acciones de control o supervisión en tiempo real, mejorando así el rendimiento de los equipos en los que se incorporan. Gracias a su diseño orientado a tareas específicas, los sistemas embebidos ofrecen soluciones eficientes en términos de consumo de energía, tamaño y costo, lo que ha favorecido su adopción en múltiples áreas de la ingeniería y la tecnología. Este tipo de aplicaciones resulta especialmente útil en áreas como el monitoreo climático y la prevención de riesgos ambientales, facilita la obtención de información relevante para la toma de decisiones y el desarrollo de sistemas de alerta temprana (Cayssials R. , 2014).

Para este proyecto, se emplea un sistema embebido basado en un microcontrolador que permite la recopilación de datos meteorológicos mediante sensores de temperatura, humedad y velocidad del viento, así como su posterior visualización y envío a una plataforma de almacenamiento en línea que es el Esp32.

Es un microcontrolador específico que es considerado como uno de los más económicos y de menor consumo de energía, permitiendo conexión dual mediante Wi-Fi y Bluetooth

trabajando a 32 bits, conteniendo un núcleo es algunos de estos y es muy indicado para la creación de redes de sensores o aplicaciones de IoT además de que también muy utilizado en este tipo de campos, en este caso para la recolección de datos de los sensores procesando la información que capturan mediante la memoria RAM que tiene integrada pudiendo almacenar datos de manera temporal y después enviándola a una base de datos en este caso, mediante programación mandando órdenes para que procese donde enviar los datos que anteriormente tenía (Romero & Janeiro, 2025). Este microcontrolador es muy importante para este trabajo de estación, por las facilidades que tiene de conseguirlo y trabajar con él para la estación meteorológica hace más práctico el trabajar con los sensores y se puede tener mayor comunicación.

1.2.6 Internet de las cosas

Se define como sistema de dispositivos de computación relacionados, maquinas mecánicas, objetos o cualquier cosa que tenga un identificador único y que este sea capaz de poder transferir datos a través de una red, con el fin de automatizar múltiples tareas y procesos de la vida cotidiana (Muñoz, 2019). A través de redes de comunicación, los sensores pueden medir diferentes variables ambientales como temperatura, humedad, velocidad del viento, presión atmosférica y calidad del aire, enviando estos datos a plataformas digitales donde pueden ser almacenados y analizados. Este tipo de sistemas facilita la observación continua de las condiciones ambientales y permite detectar cambios en el entorno de manera oportuna. Gracias a estas características, el IoT se ha convertido en una herramienta importante en áreas como la gestión ambiental. A través del microcontrolador ESP32, se recopila información proveniente de sensores encargados de medir variables ambientales como la temperatura, la humedad y la velocidad del viento. Posteriormente, estos datos son transmitidos mediante conexión a internet hacia una plataforma de almacenamiento en línea, donde pueden ser registrados y visualizados para su análisis. De esta manera, el sistema permite el monitoreo de las condiciones ambientales del entorno, lo que contribuye a la identificación de factores que pueden favorecer la aparición de incendios forestales (Muñoz, 2019).

1.2.7 Almacenamiento de datos

SE considera como el registro de datos de alta capacidad ya sea de forma permanente o temporal que son gestionados por sistemas operativos, aplicaciones o páginas web en este caso, cuando el ESP32 recibe los datos de los sensores conectados, migra a estos hacia Google sheets que aunque es una hoja de cálculo basada en la nube para este proyecto es factible utilizarlo, por su facilidad de editar y recibir datos ayuda bastante (Rubio Ahumada, 2021), además de que se puede utilizar como una base de datos en la que solo son pocos datos que reciben de los sensores siendo más accesibles, seguros a que solo se puede entrar a los datos mediante un correo específico que se redirige los datos a la hoja de cálculo y ventaja es que se puede acceder en cualquier dispositivo y en cualquier momento (Chicano Tejada, 2025).

1.2.8 Inteligencia Artificial

El autor (Novera, 2025), lo describe cómo un concepto que tiene similitud sobre cómo funciona el cerebro humano en capturar datos, reconocer patrones y tomar decisiones como si se tratase de un ser humano. La única diferencia de una y otra es el sentir las emociones que uno puede expresar, que solo puede reconocer y recopilar esa información haciendo un aprendizaje automático esto se le conoce como aprendizaje automático. Aunque sabemos que aún no hay una definición específica en que defina realmente que es una inteligencia artificial, aunque explican que presentan técnicas y decisiones complejas que para el ser humano sean difíciles y complejas de realizar (Perez Ugena, 2024).

Clasificación de la IA

La inteligencia artificial se puede clasificar de diferentes maneras como lo puede ser funcionalidad, capacidad o nivel de autonomía como se muestra en la siguiente tabla.

Tabla 2 clasificación de Inteligencia artificial

Débil o estrecha	Fuerte o General	Superinteligencia
Esta es la que está en el mercado la que todas las personas usan en su día a día, está diseñada para tareas simples y con un conjunto de datos mediana. Las IA's que se encuentran en este apartado son Chat Gpt, Gemini, Perplexity, Deep seek, copilot etc.	Este es más como un concepto teórico que aún no hay registros de una IA existente en la que sea capaz de aprender como un humano como creatividad, sentir emociones y tener un juicio moral.	Al igual que la anterior aún se refiere a la teoría en que sería una inteligencia que fuese superior a inteligencia que la del humano, sea capaz de resolver problemas autónomamente (Novera, 2025).

Redes neuronales artificiales

Se creó a base de modelos computacionales en los que se buscaban formalizar el cerebro humano a base de matemáticas haciendo que su aprendizaje fuese aumentando con base su experiencia basándose en la imitación de neuronas en la que anteriormente fueron entrenadas o que ahí mismo se entrena haciendo que procese dicha información mostrando como salida múltiples respuestas (Flórez López & Fernández Fernández, 2008), mencionan que estas son una de las mejores IAs que están en el mercado mejorando su amplitud en lenguaje natural en el procesamiento y en la que nacieron modelos con aprendizaje de generar imágenes y videos a partir de texto esta es la que tiene una similitud con la del ser humano hacen conexiones entre de las neuronas artificiales donde la primera capa recibe los datos los procesa y los pasa a otras donde se les conoce como capas profundas hasta llegar a la salida ya con toda la información recopilada dando resultados específicos (Caballero & Martín, 2025) así mostrando un ejemplo de cómo se estructura una red neuronal artificial.

Redes neuronales convolucionales

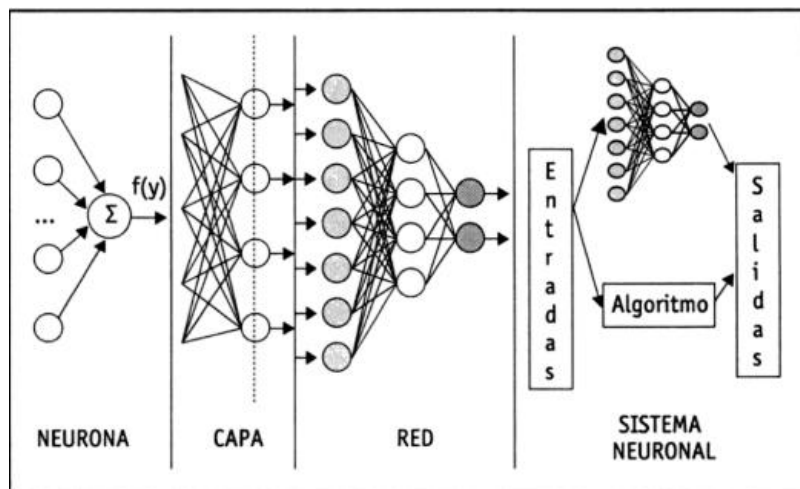


Ilustración 4 estructura basada en una red neuronal artificial

Es una de las arquitecturas más avanzadas y utilizadas, su arquitectura se basa en procesar las cosas haciendo investigaciones muy precisas que pueden ayudar a extraer información exacta de cualquier conjunto de datos. Es muy utilizado para la detección de imágenes en la que tiende a procesar incluso los detalles más sutiles de una imagen para facilitar el reconocimiento de lo que se muestra para poder predecir mejor que es lo que se observa (López Sotelo, 2023), también muy similar trabaja por capas la diferencia es en codificar las

imágenes un claro ejemplo es que para la imagen de un incendio con seres humanos se percibe a simple vista la llama y así también diferenciar el humo, así que eso realiza la red neuronal codifica las imágenes aprendiendo anteriormente saber cómo es un incendio como se ve hasta identificar si en la imagen existe la presencia de un incendio (Torres, 2018).

Conjunto de Datos

Un conjunto de datos es una colección estructurada de información utilizada para el entrenamiento, validación y prueba de modelos de aprendizaje automático. En visión por computadora, los conjuntos de datos suelen estar conformados por imágenes etiquetadas que permiten al algoritmo aprender patrones visuales específicos, como formas, colores y texturas asociadas a un fenómeno determinado (Raschka, 2019), para este caso se utilizaran múltiples imágenes con las que serán entrenadas las red neuronal en la que habrá evidencia de fuego, humo y lugares en donde puede haber semejanzas como nubes grandes o que bajan a la superficie (neblina), y también el registrar los datos de los sensores que es la temperatura, el viento y la humedad del entorno para estudiar si se 'pudiese generar un posible incendio o durante para conocer si aún hay más riesgo de que se pueda propagar hacia más lugares.

Métricas para la evaluación de algoritmos

Permite medir el desempeño de un modelo en el momento de realizar una predicción sobre un conjunto de datos, en este proceso se trata de analizar si el modelo es capaz de cumplir las clases de los datos evaluando las predicciones del algoritmo con los valores reales del conjunto de prueba, la evaluación de modelos es un paso esencial en el reconocimiento de patrones y el aprendizaje automático, debido a que permite determinar la capacidad del algoritmo para generalizar su aprendizaje hacia nuevos datos que no fueron utilizados durante el entrenamiento (Bishop, 2006).

Para obtener estas métricas, comúnmente se emplea una herramienta conocida como matriz de confusión, la cual clasifica los resultados de las predicciones del modelo en diferentes categorías: verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos. Entre las métricas más utilizadas para evaluar modelos de clasificación se encuentran la precisión, exactitud, sensibilidad y la puntuación F1, las cuales permiten obtener una visión más completa del comportamiento del algoritmo (Bishop, 2006).

Precisión

Mide la proporción de predicciones positivas correctas respecto al total de predicciones positivas realizadas por el modelo, es decir que mide la habilidad de clasificar un no etiquetar como positivo, Sin embargo, en escenarios con datos desequilibrados, la precisión por sí sola es insuficiente para calificar el desempeño global. Por ello, es fundamental analizarla en conjunto con indicadores como la sensibilidad, la especificidad y el valor F1 para obtener una visión completa del comportamiento del clasificador, así que se muestra la formula con la que se mide el algoritmo de precisión (Contreras, Fuentes, & Rodríguez, 2024).

Ecuación 1 fórmula para medir la precisión

$$presicion = \frac{VP}{VP + FP}$$

Sensibilidad

Calcula como la relación del número de instancias ya predichas sobre el número total de positivos siendo que para esta métrica para obtener un resultado positivo tiene que ser cercano el porcentaje a 100%, tiene que utilizarse en el conjunto de datos para evaluar la capacidad del modelo para identificar de manera correcta todas las instancias positivas con relación a los verdaderos positivos. En este caso podemos utilizarlo para detectar fuego en una imagen en un lote de imágenes detectando en cuantas de estas imágenes podemos encontrar rastros de o evidencia de que realmente hay fuego (Contreras, Fuentes, & Rodríguez, 2024), utilizando la siguiente formula.

Ecuación 2 fórmula para medir la sensibilidad

$$sensibilidad = \frac{VP}{VP + FN}$$

Exactitud

Es el número de predicciones que se realizan de manera correcta realizadas sobre las predicciones, para esta métrica siempre se suele utilizar una escala de 0% a 100% donde el 0% se representa como una pésima precisión y 100% para una buena precisión, esta métrica es aconsejable utilizarla en un conjunto de datos en donde las clases están equilibradas en cuanto a la importancia con el problema que se está abordando , como tener el mismo número de conjunto como en imágenes de fuego, humo y donde no se encuentren incendios para este caso (Contreras, Fuentes, & Rodríguez, 2024), utilizando la siguiente formula.

Ecuación 3 fórmula para medir la exactitud

$$Exactitud = \frac{VP + VN}{VP + FP + FN + VN}$$

F1-score

Esta ecuación se utiliza para sacar el promedio de precisión y sensibilidad utilizando los falsos positivos y los falsos negativos utilizada cuando en el conjunto de datos las clases no están equilibradas y sobre sale más una que las demás (Contreras, Fuentes, & Rodríguez, 2024), utilizando la siguiente formula.

Ecuación 4 fórmula para para medir el puntaje f1

$$puntaje\ f1 = 2 * \frac{presicion * sensibilidad}{presicion + sensibilidad}$$

1.2.9 Python

Python es un lenguaje de programación multiplataforma que se puede utilizar en distintos entornos operativos (imagen del logo del programa Python), este se encuentra ampliamente utilizado en ciencia de datos, inteligencia artificial y desarrollo de sistemas debido a su simplicidad sintáctica y amplio ecosistema de bibliotecas especializadas (Hernández Bejarano, 2025). Algunos de las funcionalidades que tienen para los usuarios para utilizarlo de forma libre es que pueden ejecutar el programa como lo desee uno, al tener acceso al código fuente se puede hacer que el programa funcione de otra manera o descubrir cómo es que funciona, compartir los programas con otros usuarios (Hinojosa Gutiérrez, 2016), para este proyecto se decidió utilizar Python, esta tiene librerías con las que se puede realizar una

red neuronal debido a que contiene una librería en la ayuda a la creación de Deep learning (Bottini, 2025).

Tensor Flow

Es una biblioteca de código abierto utilizada para el desarrollo y entrenamiento de modelos de aprendizaje profundo, permitiendo la implementación eficiente de redes neuronales mediante el uso de gráficos computacionales (Egmond, 2018), una de las herramientas más fuertes para la ayuda de la inteligencia artificial puesto que una de sus características es permitir pequeños proyectos además de entrar los modelos utilizando el algoritmo backpropagation ayuda a ajustar el peso de las conexiones neuronales automatizando una parte del trabajo en donde se centre más en la arquitectura del modelo y no preocuparse por el área matemática, así que se utilizó para la creación de las clases para red neuronal también una ventaja para el conjunto de datos es poder realizar las etiquetas de las imágenes de forma manual para poder observar si en la imagen nos solo existe la variante de fuego sino también de humo y en los casos en donde las imágenes no cuenten con ninguno de estos etiquetarlo para que no tenga problemas o dar falsos positivos al momento de ejecutar la red neuronal (Rodriguez, 2025).

Google Colab

Está basada en la nube que permite ejecutar código Python mediante notebooks (localizada en las herramientas con el icono que se muestra en la Ilustración 10)A, facilitando el entrenamiento de modelos de aprendizaje automático al proporcionar acceso a recursos computacionales como GPU sin necesidad de infraestructura local (Liu & Mehta, 2019), no requiere de instalaciones grandes solo de las librerías que se utilizaran en el código además, de tener la facilidad de guardar nuestras notas en la nube y poder ingresar en ellas en cualquier momento y en cualquier dispositivo además de poder compartir nuestras notas con terceros (Galeano Prieto & Pastor Ruiz, 2022).

Google sheets

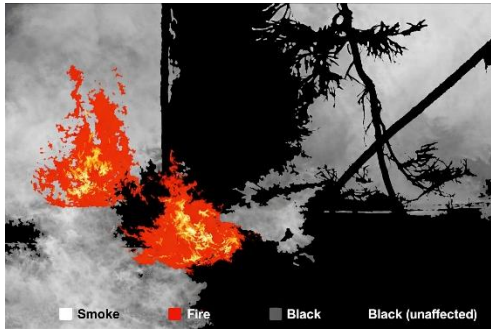
Google Sheets es una herramienta de hojas de cálculo en la nube que permite el almacenamiento, visualización y análisis de datos en tiempo real, (localizada en las herramientas con el icono que se muestra en la Ilustración 11), siendo utilizada como base de datos ligera en sistemas IoT (Piora, 2021), para tener registro al momento de capturar datos mediante los sensores y ser capturados gracias al esp32 nos permite crear funciones como crear graficas de los datos y tener un registro de los datos ni tan limitado lo que ayuda a la comparación de datos mientras la red y estación mejoran, además de poder tener conexiones con microprocesadores teniendo una conexión inalámbrica solo con tener una ide y que ambos estén conectados a una red WI-fi (autores, 2023).

1.2.10 Procesamiento de Imágenes

Las imágenes se definen como una función bidimensional (en el que normalmente es de sistema binario) de intensidad de la luz $f(x,y)$ en donde x y y marcan coordenadas específicas y el valor de f pueda ser cualquier punto de (x, y) que sea proporcional al brillo (Zaragoza & Bartolucci, 2008). Dependiendo del modelo de representación, una imagen puede almacenarse en escala de grises o en espacios de color como RGB, donde cada píxel contiene información correspondiente a distintos canales cromáticos (De la Fraga, 2001), así que con ello podemos extraer características asociadas con patrones de color, textura y forma, estos son muy necesarios para detectar humo y fuego que se encuentre en la imagen.

Segmentación de Imágenes

La segmentación de imágenes es importante para la detección de imágenes, con este apoyo se puede identificar y delimitar mejor el área en una imagen, en este caso saber exactamente se encuentra fuego o humo en los pixeles en las imágenes como se parecía en las siguientes Ilustraciones (13 y 12).



*Ilustración 6 identificación del Fuego y Humo
(Elaboración Propia).*



*Ilustración 5 captura de un incendio forestal
(conjunto de datos).*

Imágenes térmicas

Las imágenes térmicas representan la distribución de temperatura de una escena mediante la captación de radiación infrarroja. En el contexto de la detección de incendios forestales, esta tecnología permite identificar focos de calor incluso en condiciones de baja visibilidad, como humo denso o durante la noche, lo que las convierte en una herramienta complementaria a los sistemas de visión convencional (Vollmer & Möllmann, 2010).

Dron mini

Los drones son plataformas aéreas no tripuladas que permiten la adquisición de imágenes y datos en zonas de difícil acceso. En la detección de incendios forestales, los drones facilitan el monitoreo aéreo, la identificación temprana de focos de calor y la evaluación de áreas afectadas (Unmanned Aircraft Systems: UAVS Design, 2010). así que se utilizó el dron DJI MINI 3 que es muy ligero, capaz de aguantar ráfagas de viento de hasta 38 km/h, contando con una cámara de 48 megapíxeles en imagen fija y grabar en 4k con zoom digital 4x ,teniendo estabilizadores para la captura de imagen y video y sensores de a proximidad para evitar choques esto (DJI, 2026), optando este dron como útil para la captura de imágenes y videos de incendios sin correr tanto riesgo en tener perdida de señal o que pueda confundirse y generar algún desperfecto en la hora del vuelo.

1.2.11 Técnicas de minería de datos

La minería de datos da un conjunto de técnicas destinadas a descubrir patrones, relaciones, útil a partir de grandes cantidades de datos. En sistemas inteligentes, estas técnicas permiten analizar información histórica y optimizando la precisión de los modelos predictivos (Witten, Frank, & Hall, 2011).

1.3 PLANTEAMIENTO DEL PROBLEMA

En México, los incendios forestales han registrado un aumento preocupante en los últimos años (gráfica en la Ilustración 7). En 2023 se contabilizaron 7 611 siniestros que afectaron 1 047 492 hectáreas; para 2024, la cifra ascendió a 8 002 incendios con un total de 1 672 216 hectáreas afectadas (Comisión Nacional Forestal, 2025).

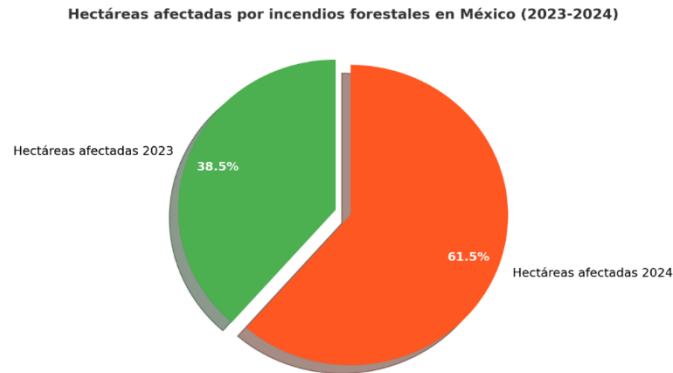


Ilustración 7 grafica de pastel del aumento de incendios forestales (Comisión Nacional Forestal 2025).

El gráfico muestra un incremento significativo en la superficie afectada por incendios forestales en México entre 2023 y 2024: mientras que en 2023 se registró el 38.5% del total, en 2024 la cifra aumentó al 61.5%. Esto representa un crecimiento de 23 puntos porcentuales, lo que se entiende de una intensificación de los incendios en solo un año.

A pesar de que ya existen múltiples métodos para la prevención y el cuidado del área llevan consigo muchos límites para poder realizar su tarea de manera efectiva y evitar la propagación de fuego. En el municipio de Tianguistenco en sinergia con la Coordinación de Protección Ambiental los registros de incendios forestales son abundantemente en superficie y en periodos de marzo- junio donde las temperaturas son altas, y los patrones más frecuentes son la quema descontrolada, por pastores en zonas abiertas.

Este problema no solo daña la biodiversidad y la vegetación, sino que también afecta la salud y seguridad de las comunidades, así como actividades económicas como la agricultura y el turismo. Los sistemas actuales de monitoreo son insuficientes para evitar la propagación.

1.4 JUSTIFICACIÓN

Los incendios forestales representan uno de los principales problemas ambientales en México y a nivel mundial, debido a su impacto en la pérdida de biodiversidad, la degradación de suelos, las emisiones de gases de efecto invernadero y las afectaciones a las comunidades humanas cercanas a las zonas boscosas. En el 2024 se registraron 8002 incendios teniendo un área total de 1 672 216 hectáreas debido a los fuertes cambios climáticos que se registran en el país, como la elevación de temperaturas provocando sequías más rápidas, lo que eleva las posibilidades de inicios de incendios y propagación de esta (Comisión Nacional Forestal, 2025).

Debido a los riesgos que generan los incendios forestales, se establecieron estaciones de monitoreo para la prevención de México con ayuda de la comisión nacional de incendios forestales maneja un sistema de predicción de incendios con imágenes satelitales e imágenes térmicas en donde señalan incendios que anteriormente ya se habían provocado (Comisión Nacional Forestal, 2025).

La innovación y ventaja de este proyecto frente a los sistemas existentes radica en tres aspectos clave:

- Integración de tecnologías de bajo costo y acceso local: a diferencia de los sistemas satelitales, se propone una página web con inteligencia artificial entrenada para reconocer incendios forestales a partir de imágenes terrestres.
- Datos en tiempo real mediante sensores meteorológicos (IoT): se incorpora una estación meteorológica conectada que mide humedad, temperatura y viento almacenando datos en Google Sheets que nos facilita la visión de los datos y registrarlos en tiempo real sin necesidad de crear una base de datos en aplicaciones especializadas en ello.
- Etiquetado automático de tipos de incendio: a diferencia de los sistemas actuales que solo señalan la presencia de fuego, este proyecto clasifica el tipo de incendio (superficie, copa, subsuelo), lo que permite a las autoridades elegir la estrategia de combate más adecuada.

Por lo tanto, el proyecto es importante porque reduce los tiempos de detección, ofrece una solución adaptada a las condiciones específicas de Tianguistenco, y complementa los sistemas nacionales con datos locales de alta frecuencia, permitiendo pasar de una gestión reactiva a una preventiva. Esto protege la biodiversidad, la salud de las comunidades y las actividades económicas como la agricultura y el turismo.

1.5 OBJETIVOS

1.5.1 Objetivo general

Desarrollar una estación de monitoreo que emita mensajes preventivos ante temperaturas ambientales elevadas y que clasifique los tipos de incendios forestales mediante una red neuronal artificial.

1.5.2 Objetivos específicos

- Analizar el estado del arte para identificar aspectos como la relevancia de estrategias de solución, métodos.
- Diseñar un modelo de red neuronal artificial que permita analizar imágenes sobre tipos de incendios forestales
- Crear una estación meteorológica que nos permita identificar temperaturas ambientales altas
- Generar un envío de notificaciones SMS sobre la detección de temperaturas altas
- Enviar datos meteorológicos a una hoja de datos en Google Sheets
- Entrenar la red neuronal con datos reales y validar el funcionamiento del modelo
- Analizar los datos como factores de predicción

1.6 HIPÓTESIS

La implementación de redes neuronales convolucionales aplicadas para el análisis de imágenes permitirá clasificar los tipos de incendios forestales en tiempo real con una precisión mayor a la de los métodos tradicionales.

2. CAPÍTULO 2

2.1 METODOLOGÍA

La metodología CRISP-DM desarrollada por las empresas SPP, NCR Y DAMILER CHRYSLER es la más utilizada en proyectos para la minería de datos, este está organizado en una secuencia de jerárquica de procesos a diferentes niveles en que se realizan por 6 etapas: Comprensión del negocio (o análisis del problema), comprensión de los datos, preparación de datos, modelado, evaluación e implementación como se muestra en Ilustración 17 donde se muestran por orden las fases a realizar de la metodología (Vicente Vírveda, González Arias, Parra Rodríguez, & Beltrán Pascual, 2019).

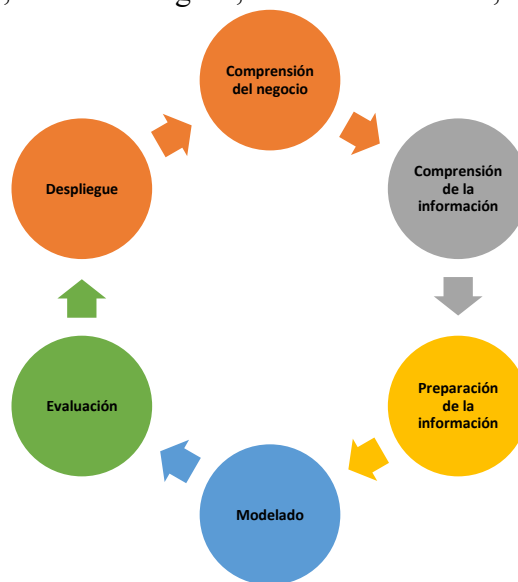


Ilustración 8 diagrama de las fases del método CRISP-DM (Elaboración propia)

Para el desarrollo del presente proyecto se adoptó un enfoque metodológico mixto, combinando investigación aplicada con desarrollo tecnológico experimental, orientado a la solución de una problemática ambiental específica (Dulhare & Houssein, 2025).

El alcance del proyecto se define como tecnológico y experimental, de acuerdo con la siguiente Tabla:

Tabla 3. Alcance del proyecto Tecnológico y Experimental

Tecnológico	Experimental
Implica la integración de sistemas embebidos, microcontroladores y plataformas de software para el desarrollo de un artefacto funcional. El uso del ESP32 como núcleo del sistema responde a sus capacidades de conectividad y procesamiento, ampliamente documentadas en sistemas embebidos modernos (Cayssials R. , 2014).	El prototipo es sometido a pruebas controladas en las que se manipulan variables independientes, como la exposición a fuentes de calor simuladas, para observar el comportamiento de las variables dependientes, tales como el tiempo de respuesta del sistema, la precisión en el registro de datos y la confiabilidad de las alertas generadas.

Asimismo, se muestra en la Ilustración 9 la arquitectura del sistema basado en sistemas embebidos.



Ilustración 9 Arquitectura del sistema con sistemas embebidos (elaboración propia)

Comprensión del Negocio

Tanto en México como en el resto del mundo, los incendios forestales son una de las mayores amenazas ecológicas. Para ello se implementó la idea de crear una página web sobre la detección de incendios forestales con inteligencia artificial junto con una estación meteorológica con el propósito de estudiar la calidad del aire, temperatura y humedad antes y durante el incendio.

Comprensión de los Datos

Se realizó la recopilación y análisis preliminar de los datos utilizados. Para el módulo de inteligencia artificial, se recolectaron conjuntos de imágenes de incendios forestales y escenarios naturales sin fuego gracias al apoyo de la brigada de protección civil y probosque con el ayuntamiento de Tianguistenco en una prueba de fuego controlado.

Para el módulo IoT, se analizaron los rangos de operación de sensores de temperatura, humedad y flama, con el fin de comprender la naturaleza de los datos a procesar, tal como se recomienda en proyectos de aprendizaje automático supervisado.

Preparación de los Datos

Con las imágenes obtenidas se etiquetan con la aplicación roboflow con un formato YOLOv5 (Ilustración 10 conjunto de imágenes para el etiquetado).

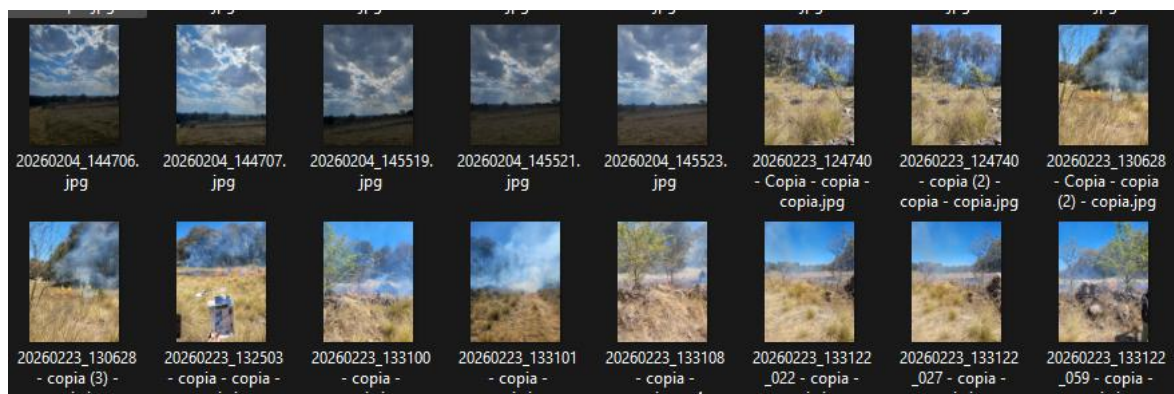


Ilustración 10 Conjunto de imágenes que se utilizaran (elaboración propia).

Posteriormente se ingresa a la aplicación roboflow con un correo y contraseña que anteriormente se solicita, para ingresar a la página principal y crear un nuevo proyecto donde se ingresaran las imágenes para el etiquetado de estas como se muestra en la Ilustración 11 donde se muestra la página principal de la aplicación.

Posteriormente se configura el proyecto a las especificaciones necesarias para la creación del proyecto como se muestra en la siguiente ilustración.

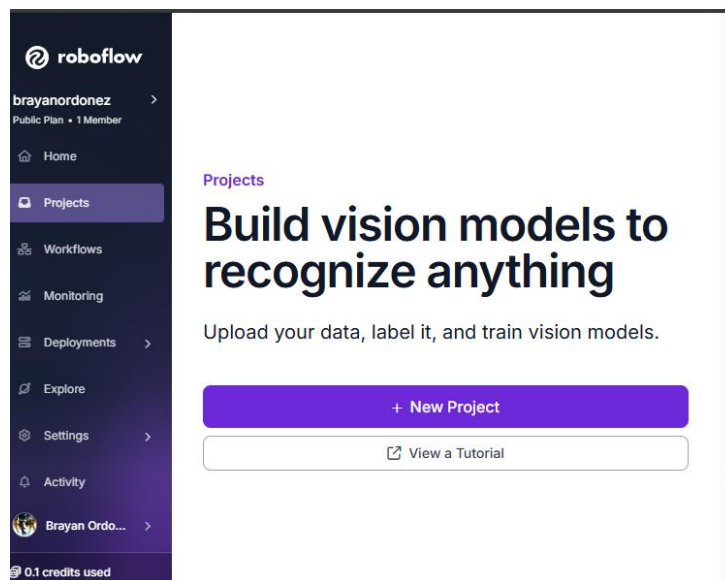


Ilustración 11 página principal de roboflow (Roboflow).

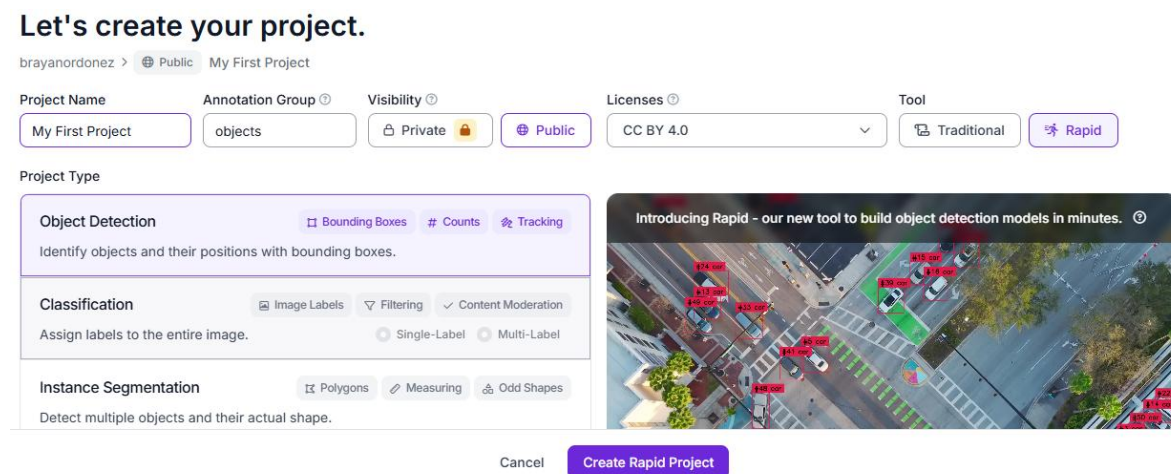


Ilustración 12 especificaciones para el proyecto en roboflow (roboflow).

Al haber creado nuestra clase para el etiquetado de las imágenes subimos nuestro conjunto de datos como se muestra en la Ilustración 13.

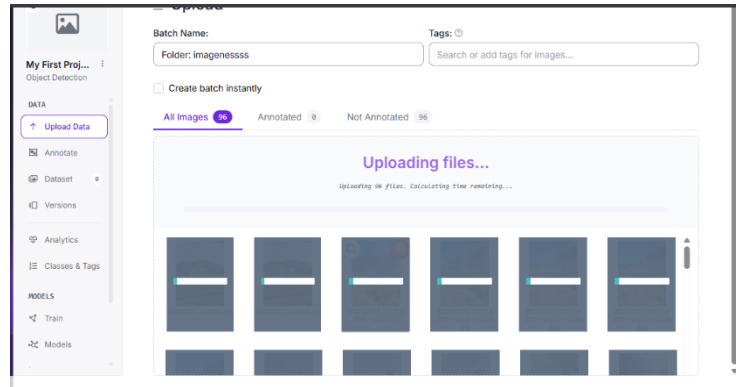


Ilustración 13 Carga del conjunto de datos (elaboración propia)

Una vez cargada las imágenes que se utilizarán se deberán etiquetar cada una de las imágenes con el nombre de “Fuego_copa” “Fuego_superficial” “Fuego_subteraneo” “humo” y “No fuego” como se muestra en la Ilustración 14.

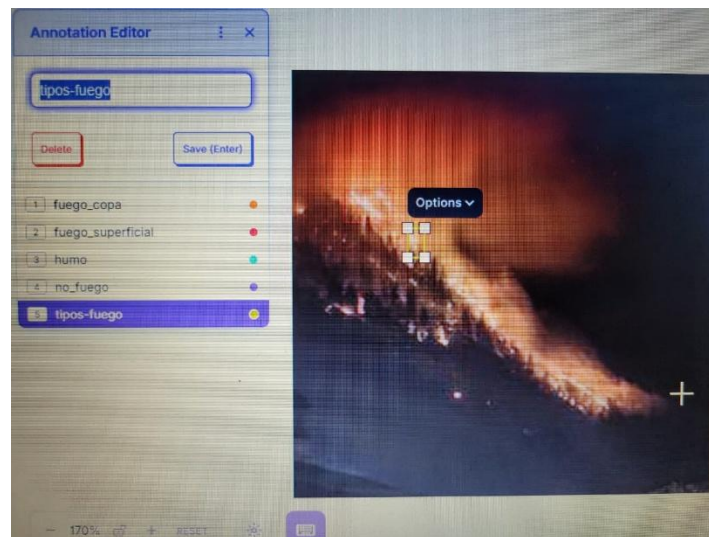


Ilustración 14 etiquetado de imágenes (elaboración propia)

Después de etiquetar todas las imágenes, se tienen que equilibrar entre tren, validez y testeo como se muestra en la Ilustración 15 para obtener buenos resultados y que no tengan problemas en entre clases

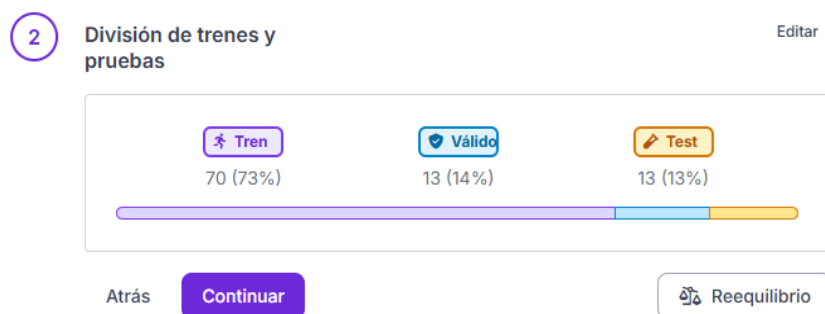


Ilustración 15 reequilibrio de las imágenes para el conjunto de datos (elaboración propia)

Posteriormente se descarga el conjunto de datos en formato YOLOv5 en .zip como se muestra en la Ilustración 17, donde deberá tener los siguientes conjuntos de datos para que funcione la red neuronal como se muestra en la Ilustración 16.

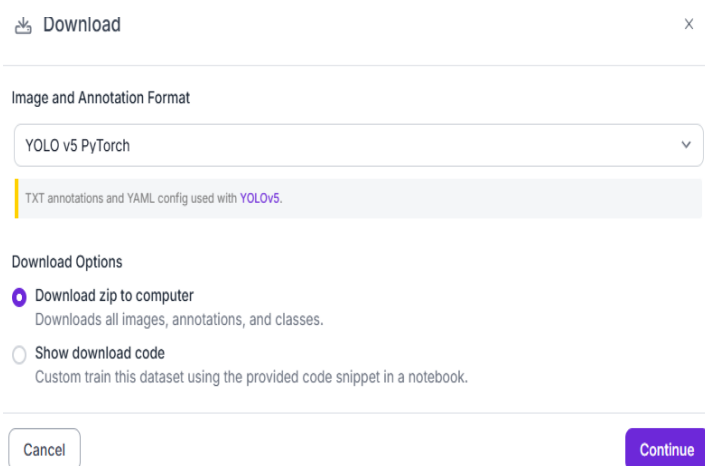


Ilustración 17 descarga del conjunto de datos en roboflow (elaboración propia)

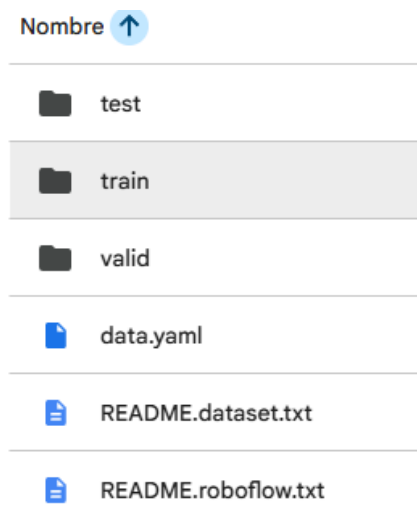


Ilustración 16 conjunto de datos para utilización de red neuronal (elaboración propia)

Modelado

Para el modelado se utilizará un modelo de una red neuronal basada en YOLOv5 esto con apoyo de la plataforma Google colab donde en el lenguaje de programación utilizado será Python, generando entorno GPU T4, mostrado en la Ilustración 18.

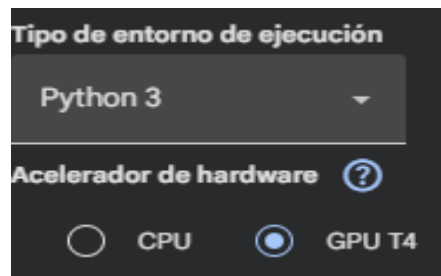


Ilustración 18 entorno de ejecución en google colab (elaboración propia)

Posteriormente se crea el código para desarrollar la red neuronal a utilizar en este proyecto, se inicia con vincular Google drive para conectar el conjunto de datos que se extrajeron desde roboflow, después se inicia con los repositorios de YOLOv5 desde GitHub (Ilustración 19).

Después se crea un archivo data yaml (como se muestra en Ilustración 20).

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

[ ] !git clone https://github.com/ultralytics/yolov5.git
    # Descarga el repositorio de YOLOv5 desde GitHub.
    %cd yolov5
    # Cambia el directorio de trabajo a la carpeta yolov5.
    !pip install -r requirements.txt
    #Instala todas las bibliotecas necesarias que YOLOv5 necesita para funcionar (como PyTorch, OpenCV, etc.)
```

Ilustración 19 código para inicio de sesión en google drive y git hub (elaboración propia)

```
data_yaml = """
train: /content/drive/MyDrive/incendios/data_incendios.vii.yolov5pytorch/train/images
val: /content/drive/MyDrive/incendios/data_incendios.vii.yolov5pytorch/valid/images

nc: 1
names: ['fuego']

"""

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

Ilustración 20 Código para la creación de la data yaml (creación propia).

Ahora se ejecuta la línea de Código donde se entrena la red neuronal realizando 50 ciclos completos donde cada época es entrenada 16 veces (como se muestra en la Ilustración 21).

```
!python train.py --img 640 --batch 16 --epochs 50 --data /content/yolo_incendios.yaml --weights yolov5s.pt --cache
```

Ilustración 21 código que permite el entrenamiento de la red neuronal (creación propia).

Finalmente, en este Código se utiliza para realizar pruebas de que la red neuronal fue eficiente y cargar una imagen en donde nos demuestra el resultado de la imagen (como se muestra en la Ilustración 22).

```
!python detect.py --weights runs/train/exp3/weights/best.pt --img 640 --conf 0.4 --source /content/drive/MyDrive/incendio.png
```

Ilustración 22 Código para evaluar resultados de la red neuronal (creación propia).

Resultado del entrenamiento de la red neuronal (Ilustración 23).



Ilustración 23 resultado del entrenamiento de la red (creación propia).

Evaluación

Al entrenar esta red neuronal convolucional (YOLOv5) se muestra que contiene 157 capas, lo que quiere decir que esta red permite un equilibrio y estabilidad al momento de utilizarla, además de mostrar las métricas y dar los resultados de haber entrenado la red neuronal con las imágenes etiquetadas y saber si fueron bien etiquetadas y como obtuvo los resultados (como se muestran en la Ilustración 24).

```
Validating runs/train/exp3/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
Class      Images  Instances  P      R      mAP50  mAP50-95: 100% 72/72 [00:33<00:00, 2.17it/s]
all        2280    7099      0.968  0.939  0.98   0.677
Results saved to runs/train/exp3
```

Ilustración 24 resultados de las métricas utilizadas para la evaluación de la red neuronal (creación propia).

También se implementó la conexión de la estación meteorológica con Google sheets para poder almacenar los datos que capturados y posteriormente guardarlos en un archivo .csv y poder graficarlos (visualizados en la Ilustración 25).

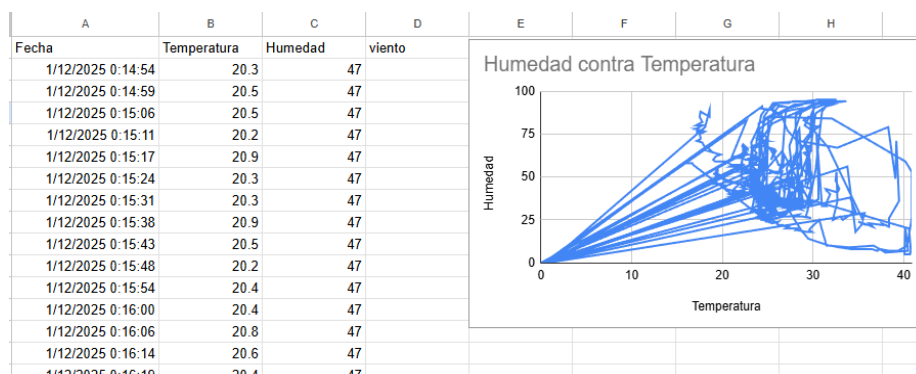


Ilustración 25 resultados de la conexión meteorológica con google sheets (creación propia).

Implementación

Para el funcionamiento requerido tanto de la estación meteorológica como para la página web este tiene que estar a una altura considerable en lugares donde hay muchos árboles esto para capturar de mejor manera los datos de viento, humedad y temperatura del entorno, pero en lugares amplios donde es escaso el árbol y hay mucha vegetación que se puede convertir en material de combustible puede estar a una altura más baja aproximadamente un metro y medio del suelo (como se muestra en la Ilustración 26) para posteriormente capturar datos con la estación e imágenes para el entrenamiento de la red neuronal.



Ilustración 26 demostración de la estación meteorológica (creación propia).

3 CAPÍTULO 3

3.1 Pruebas y Resultados

Para dar inicio al desarrollo del prototipo, se realizó un análisis de los componentes electrónicos necesarios para cumplir con los objetivos de autonomía y conectividad. Se seleccionó el microcontrolador ESP32, como unidad central de procesamiento debido a su módulo Wi-Fi integrado y su bajo consumo energético en comparación con otras placas de desarrollo.

Para la adquisición de datos físicos, se determinó el uso del sensor DHT11 para el monitoreo de variables climáticas (temperatura y humedad) y un módulo de sensor de flama infrarrojo para la detección inmediata de fuego (ejemplo en la Ilustración 27). Con el fin de garantizar

la autonomía energética en campo, se diseñó un subsistema de alimentación basado en una batería de polímero de litio (Li-Po) de 3.7V, gestionada por un módulo de carga TP4056 y respaldada por un panel solar fotovoltaico de 6V.



Ilustración 27 sensor DHT11 (ElectroCrea.com)

3.2Diseño de la Arquitectura Electrónica

Se procedió al diseño del diagrama esquemático de conexiones (Topología en Estrella). Se implementó un mecanismo de protección y regulación de voltaje utilizando un módulo regulador de voltaje MT3608 (Ilustración 28) en la línea positiva de la batería, aumentando el voltaje de salida de 3.7V a un nivel seguro de 5V para la alimentación del ESP32.

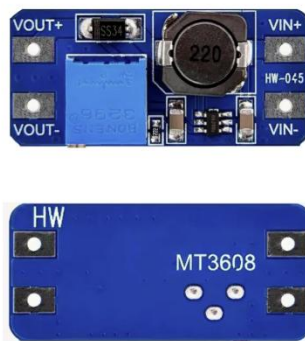


Ilustración 28 módulo de regulador de voltaje MT3608 (SIEEG 2025)

Para la interfaz visual local, se integró una pantalla LCD 16x2 (Ilustración 29) mediante el protocolo I2C. Se optó por una configuración de alimentación híbrida, conectando el VCC de la pantalla directamente a la salida del interruptor principal para maximizar el contraste, mientras que los sensores y el microcontrolador se conectaron a la línea regulada.



Ilustración 29 Pantalla LCD 16x2 (Mercado Libre).

También con ayuda del programa proteus se diseñó las conexiones de todos los componentes que se utilizaran en la estación meteorológica, para saber con más precisión y evitar posibles falsos en las conexiones o perdidas de datos como lo puede ser en la pantalla lcd y el sensor dht11 al entregarlos

al esp32 como se muestra en la Ilustración 15 de cómo se deberán conectar, así también se creó un diagrama esquemático (Ilustración 30), para las demás conexiones para darle energía a la estación meteorológica como lo es con el panel solar el switch, el módulo tp4056 para cargar la batería de litio y el regulador de voltaje mt3608.

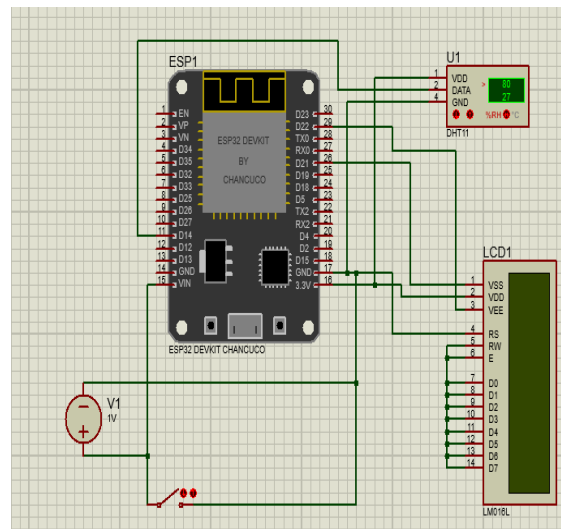


Ilustración 30 Componentes electrónicos utilizados en el sistema de detección de incendios (elaboración propia en proteus)

Así que para revisar como serían las conexiones sobre la estación meteorológica se creó un diagrama para dar un ejemplo de cómo son las conexiones correctas y esta tenga una función correcta (como se muestra en la Ilustración 31).

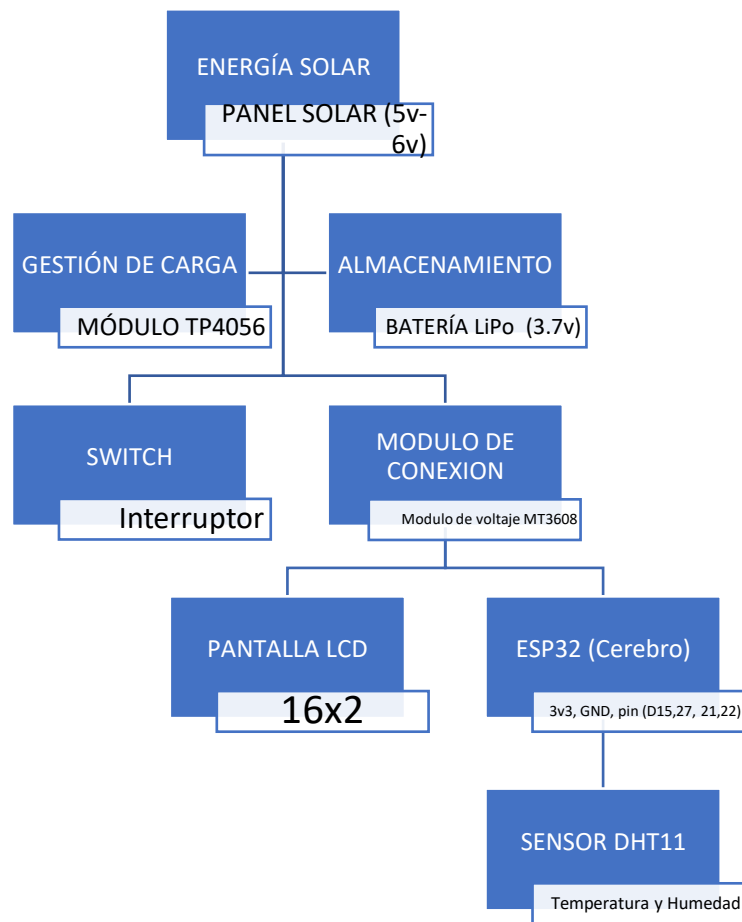


Ilustración 31 Diagrama esquemático de conexiones del sistema autónomo.

3.3 Desarrollo e Implementación

Desarrollo del Firmware (ESP32)

La programación del microcontrolador se llevó a cabo utilizando el IDE de Arduino en lenguaje C++. Se desarrollaron algoritmos para la lectura cíclica de los sensores cada 5 segundos (como se muestra en la ilustración 32). Para la transmisión de datos, se implementó una rutina de conexión Wi-Fi que gestiona el enlace con un punto de acceso móvil.

Se configuró el cliente HTTP para realizar peticiones tipo GET hacia la API de Google, enviando las variables de temperatura, humedad y estado del fuego incrustadas en la URL. Se incluyó el manejo de redirecciones HTTPS mediante la instrucción setFollowRedirects

para garantizar la compatibilidad con los servidores de Google.

```
incenfor.ino
1  #include <WiFi.h>
2  #include <HTTPClient.h>
3  #include <DHT.h>
4  #include <LiquidCrystal_I2C.h>
5
6  // =====
7  // 1. CONFIGURACIÓN CAMPO (HOTSPOT CELULAR)
8  // =====
9  const char* ssid = "Motorrouter";
10 const char* password = "12345678";
11
12 // TU LINK DE GOOGLE SCRIPT (El que termina en /exec)
13 String googleScriptURL = "https://script.google.com/macros/s/AKfycbwV59EgazoqToKvLUCF2J1hDnyF8QvaiM1FW9tSfD76oSDtoFukPqaR3jmlxfR6vSG2oQ/exec";
14
15 // =====
16 #define DHTPIN 15
17 #define DHTTYPE DHT11
18 DHT dht(DHTPIN, DHTTYPE);
19 LiquidCrystal_I2C lcd(0x27, 16, 2);
20
21 void setup() {
22   Serial.begin(115200);
23   dht.begin();
24   lcd.init();
25   lcd.backlight();
26
27   WiFi.begin(ssid, password);
```

Ilustración 32 Fragmento del firmware desarrollado para el ESP32 (creación propia).

Al cargar los datos al ESP32 y conectar los componentes a él se muestran ya los datos reflejados en el LCD (evidencia en la Ilustración 33), validando que el código que se le agrego al ESP32 responde de manera correcta recibiendo los datos del sensor DHT11 y guardando los registros en Google sheets.



Ilustración 33 Captura de datos del sensor Dht11 mostrados en un LCD (Elaboración propia)

Implementación de la Infraestructura en la Nube

Se configuró una hoja de cálculo en Google Sheets para fungir como hoja de registros de datos en la nube (como se muestra en la Ilustración 34). Mediante Google Apps Script, se desarrolló y desplegó un servicio web encargado de recibir las peticiones del ESP32.

Dentro de este entorno serverless, se programó la lógica de monitoreo crítico: el script evalúa cada dato entrante y, si la temperatura supera el umbral de 30°C o se recibe una señal de fuego activo, ejecuta una llamada a la API de Twilio. Esto permite el envío automatizado de alertas SMS al número telefónico del personal encargado, independientemente de la ubicación del dispositivo.

	A	B	C
1	Fecha	Temperatura	Humedad
2	1/12/2025 0:14:54	20.3	47
3	1/12/2025 0:14:59	20.5	47
4	1/12/2025 0:15:06	20.5	47
5	1/12/2025 0:15:11	20.2	47
6	1/12/2025 0:15:17	20.9	47
7	1/12/2025 0:15:24	20.3	47
8	1/12/2025 0:15:31	20.3	47
9	1/12/2025 0:15:38	20.9	47
10	1/12/2025 0:15:43	20.5	47
11	1/12/2025 0:15:48	20.2	47
12	1/12/2025 0:15:54	20.4	47
13	1/12/2025 0:16:00	20.4	47
14	1/12/2025 0:16:06	20.8	47
15	1/12/2025 0:16:14	20.6	47
16	1/12/2025 0:16:19	20.4	47
17	1/12/2025 0:16:26	20.3	47
18	1/12/2025 0:16:30	20.2	47
19	1/12/2025 0:16:37	20.8	47

Ilustración 34 Registro de datos ambientales en Google Sheets (creación propia).

Desarrollo de la Interfaz de Usuario e IA

Para la etapa de visualización y diagnóstico, se desarrolló una aplicación web utilizando el framework Streamlit en lenguaje Python (desarrollo de la página en la ilustración 35). Se implementó la librería panda para la lectura en tiempo real del archivo CSV exportado por Google Sheets, permitiendo el monitoreo remoto de las variables ambientales, integrando el modelo de visión artificial YOLOv5 (previamente entrenado) mediante la librería PyTorch.

Además de añadir un módulo de carga de imágenes que procesa fotografías capturadas externamente, identificando patrones de fuego y humo. El sistema fue programado para emitir un diagnóstico técnico basado en la detección, clasificando el siniestro como superficial, subterráneo o de copa, y calculando una probabilidad de propagación estimada.



Ilustración 35 Interfaz web del sistema de monitoreo ambiental (creación propia).

3.4 Validación de Sensores y Alertas

Alerta Remota: Al superar los 30°C, el sistema en la nube procesó el dato y el mensaje SMS como se muestra en la ilustración 21, fue recibido en el dispositivo móvil con una latencia promedio de 5 a 10 segundos, dependiendo de la calidad de la red celular como se muestra en la ilustración 36.

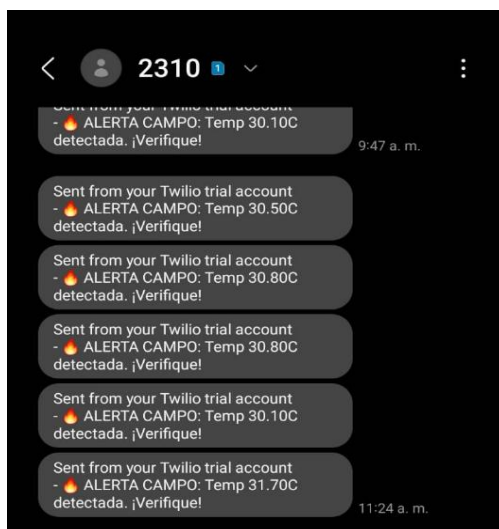


Ilustración 36 Recepción de alerta SMS en dispositivo móvil. (creación propia).

3.5 Evaluación del Modelo de Visión Artificial

Se probaron distintos escenarios visuales cargando imágenes de incendios reales y falsos positivos en la plataforma (evidencia en la Ilustración 37). El modelo YOLOv5 logró diferenciar correctamente entre fuego y humo en el 90% de los casos de prueba, proporcionando la clasificación del tipo de incendio y la barra de probabilidad de propagación de manera consistente con la teoría de incendios forestales.

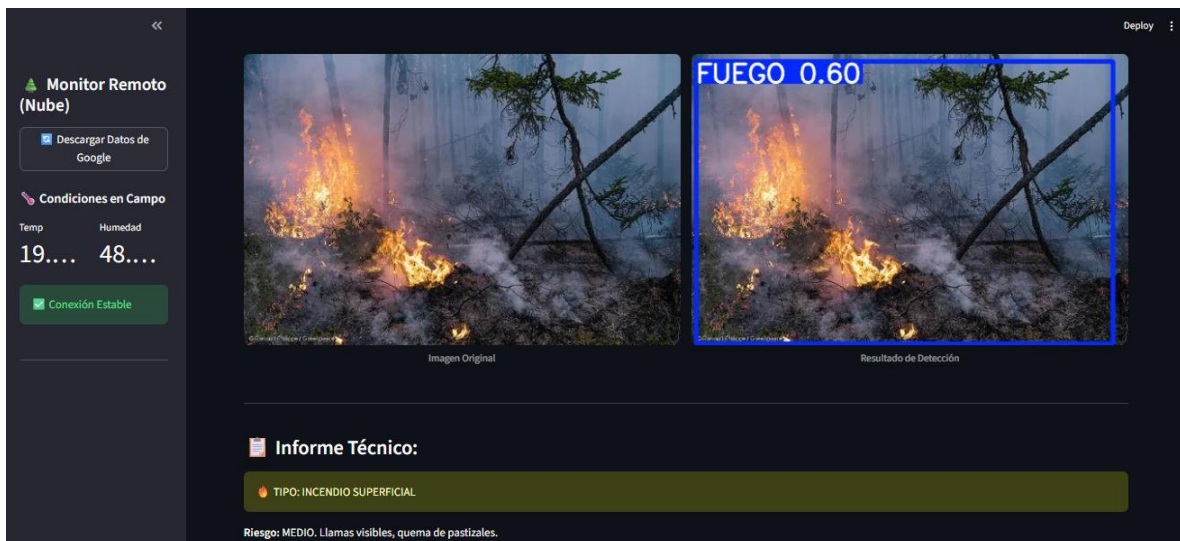


Ilustración 37 Diagnóstico de incendio forestal mediante modelo YOLOv

3.6 Pruebas de la red neuronal con yoloV5

Se realizó una red neuronal en Google colab que contiene muchas facilidades para la creación de la red neuronal como sus librerías que utilizan ya están previamente descargadas, además de que ofrece GPU (Unidad de Procesamiento Gráfico) gratuitos que son muy necesarios, ayudan al aceleramiento de la red neuronal en caso de que nuestro equipo no cuente con un GPU esto reduce por mucho el tiempo de entrenamiento (Goodfellow, Bengio, & Courville, 2016).

3.7 Pruebas del entrenamiento de la red neuronal

En estas pruebas se realizaron demostraciones con estos tipos de hiperparámetros para saber cómo se comportaba la red y estas son los resultados que se esperan para demostrar que es 100% efectiva para ser utilizada.

Modelo: YOLOv5s, esta versión es la más optimizada, utiliza menos recursos en el momento de entrenar la red haciendo que esta sea más eficiente de y veloz de entrenar, aunque como lo menciona (Sirisha, Praveen, & Srinivasu, 2023), se debe tener un poco de cuidado debido que al tener un conjunto de datos (50 imágenes etiquetadas para este caso) funciona bastante bien, pero para al obtener un conjunto de datos con más imágenes tienden a sobre ajustarse y que no reconozca bien las imágenes.

Capas 157 y Parámetros, (7.2 millones): Se define esto como la representación suficiente para aprender características visuales sin ser excesivamente compleja (Probst, Bischl, & Boulesteix, 2018), al tener demasiados parámetros tienden a memorizar así que para este conjunto de datos al ser pequeño esta equilibrado y para las tres pruebas realizadas fueron las mismas esto a que son muy eficientes y dan buenos resultados.

Épocas (entre 50 y 100): Es notorio, es un buen comienzo de partida para la red ya que al ser menor la red no aprenderá bien y no reconocerá las imágenes, en cambio al tener más épocas se obtendría un sobreajuste, así que también debe tener similitud con el conjunto de datos para correr algún riesgo de ser entrenado de manera no adecuada o tener un sobreajuste como lo comenta (Li, y otros, 2020), al hacerlo de 100 épocas tardaba más en procesar las imágenes o al estar mucho tiempo cargando Google colab automáticamente lo detenía esto debido al procesador del equipo y la ram del mismo, pero otras veces lo realizaba de manera correcta.

Lote (16): ayuda a obtener un balance entre estabilidad y velocidad para el entrenamiento de la red así que para el conjunto de datos que se utiliza para esta prueba de la red no es tan recomendada hacerla con ese número de datos porque está muy limitado, pero aun así se puede realizar y también al aumentar el lote este pueden dar muchos errores en la imagen entrenada, como lo menciona (Sirisha, Praveen, & Srinivasu, 2023).

Imágenes totales: Al tener esta cantidad de imágenes en un conjunto de datos no es recomendable, está limitado a la combinación de aprendizaje y métricas, pero este también

ayuda a la comprensión de generalizar en las imágenes (Li, y otros, 2020), le permite comprender más las imágenes etiquetadas arriesgando el balance que se debe de tener en la red neuronal.

Métrica principal (Precisión media promedio): Es importante que nos ayuda a tener múltiples umbrales de confianza antes de que la respuesta sea arrojada y así obtener una confianza exacta de la imagen analizada (Sirisha, Praveen, & Srinivasu, 2023).

Evaluación (Matriz de confusión + Precisión + Recall + F1): Al tener como métrica principal “precisión media promedio” nos aseguramos de tener una evaluación correcta para tener solo dos clases funciona como sistema binario de objetos así permitiéndonos describir (Verdaderos positivos, Falsos positivos, Falsos negativos, Verdadero negativos), (como se muestra en la ilustración 38).

```
print("\n--- Iniciando el entrenamiento ---")

# Usamos el archivo .yaml que acabamos de corregir
!python train.py \
  --img 640 \
  --batch 16 \
  --epochs 100 \
  --data /content/dataset_incendios/incendios_config.yaml \
  --weights yolov5s.pt \
  --cache
```

Ilustración 38 Captura del apartado del entrenamiento de la red neuronal (creación propia).

Confianza:

Prueba 1

Para la primera prueba se utilizó un conjunto de 10 imágenes, con la finalidad de conocer cómo funciona la red, ver la forma del etiquetado si es de manera correcta ya con los parámetros ajustados.

En la Ilustración 39, se muestra la detección y muestra la predicción de que realmente hay fuego en la imagen cargada, a pesar de que en las métricas (tabla 4) arroja resultados buenos para una red, en la práctica muestra solo un 60% de confianza al detectar fuego.



Ilustración 39 resultados de la confianza de la red neuronal (elaboración propia).

Tabla 4 resultados de las métricas de la prueba 1 (elaboración propia).

Precisión	Sensibilidad	Precisión media promedio	F1-Score	Confianza
0.948	1	0.995	0.973	60

Prueba 2

En la segunda prueba se muestran mejores avances que el primero esto a que en este se aumentan las imágenes etiquetadas que son 210, para el entrenamiento de esta red (mostrado en la Ilustración 40), pero la confianza varía mucho debido a que tuvo que ver mucho en



Ilustración 40 resultados de la confianza de la red neuronal (elaboración propia).

cómo se etiquetaron las imágenes visto que solo se encerrabas en cajas los lugares donde solo había llamas ya sea de manera cercana y lejana es por ello por lo que varía el umbral de confianza en cada etiqueta.

Tabla 5 resultados de las métricas de la prueba 2 (elaboración propia).

Precisión	Sensibilidad	Precisión media promedio	F1-Score	Confianza
0.979	0.925	0.97	0.968	55-75

Prueba 3

En esta prueba nuevamente se aumentaron las imágenes siendo un total de 2280 imágenes y se modificó también el etiquetado, no solo se encuentra la caja en el fuego, sino que también donde se encuentra el incendio en la copa de un árbol, tronco, o en la tierra donde está el pastizal o cualquier material que se pueda convertir en combustible, teniendo un umbral de confianza mayor y siendo más preciso al detectar un incendio (como se muestra en la Ilustración 41).



Ilustración 41 resultados de la confianza de la red neuronal (elaboración propia).

Tabla 6 resultados de las métricas de la prueba 3 (elaboración propia).

Precisión	Sensibilidad	Precisión media promedio	F1-Score	Confianza
0.968	0.939	0.98	0.95	88-89

Prueba 4

En la prueba seguimos con la cantidad de imágenes iguales de un total de 200 mil imágenes, pero con un cambio enorme en el que se modificaron los tipos de etiquetas los cuales fueron “fuego_copa”, “fuego_superficial”, “fuego_subterraneo”, “humo” y “no_fuego” manteniendo el entrenamiento que se utilizó en las pruebas pasadas así obteniendo los resultados siguientes en la ilustración 42 y mostrando las métricas de evaluación en la tabla.



Ilustración 42 resultados de la red neuronal entrenada (elaboración propia).

Tabla 7 Resultados de las metricas de la prueba 4 (elaboracion propia).

Nombre de la clase	presicion	recall	mAP50	mAP50-95
fuego_copa	0.895	0.837	0.927	0.497
fuego_superficial	0.83	0.889	0.917	0.459
humo	0.669	0.559	0.623	0.246
no_fuego	0.795	0.873	0.894	0.471
Fuego_subterraneo	0.797	0.789	0.84	0.418

4 CAPITULO 4

4.1 CONCLUSIONES

El desarrollo del presente proyecto permitió cumplir de manera satisfactoria con el objetivo general planteado en los antecedentes, el cual consistió en diseñar e implementar un sistema inteligente para la detección temprana de incendios forestales mediante el uso de tecnologías de Internet de las Cosas (IoT) e inteligencia artificial. A lo largo del trabajo se logró integrar de forma eficiente tanto el componente físico de monitoreo ambiental como el componente lógico de análisis y visualización de la información, demostrando que la combinación de estas tecnologías representa una alternativa viable y funcional para la prevención de este tipo de desastres naturales.

En relación con los objetivos específicos, se logró la selección adecuada de los sensores ambientales, permitiendo la medición confiable de variables como temperatura y humedad, las cuales son determinantes en la propagación de incendios forestales. Asimismo, se cumplió con la implementación del microcontrolador ESP32 como núcleo del sistema, aprovechando sus capacidades de conectividad inalámbrica para el envío de datos a la nube en tiempo real.

Otro logro relevante fue el desarrollo de una interfaz web que facilita el monitoreo remoto de las condiciones ambientales, así como la visualización de los resultados generados por el modelo de visión artificial. El uso de una plataforma accesible permitió que el sistema pudiera ser interpretado fácilmente por el usuario final, reduciendo la complejidad técnica y favoreciendo su posible adopción en escenarios reales. De igual manera, la implementación del modelo de inteligencia artificial para la clasificación de incendios demostró un desempeño adecuado, reflejado en las métricas obtenidas durante la etapa de evaluación.

Del análisis conjunto de los datos ambientales y los resultados del modelo de visión artificial, se concluye que ambos enfoques se complementan de manera efectiva. Mientras que el monitoreo ambiental proporciona un contexto preventivo, el análisis visual permite confirmar la presencia de un incendio, lo que contribuye a generar alertas más confiables y oportunas.

4.5 RECOMENDACIONES

Si bien el sistema desarrollado cumple con los objetivos propuestos, existen diversas oportunidades de mejora que podrían ser consideradas como trabajo futuro. Una de las principales recomendaciones es la incorporación de un mayor número de sensores, tales como sensores de gases o sensores de partículas, los cuales permitirían obtener información adicional sobre la presencia de humo o combustión, incrementando la precisión del sistema en sus primeras etapas de detección.

Asimismo, se recomienda ampliar el conjunto de datos utilizado para el entrenamiento del modelo de visión artificial, incorporando imágenes capturadas en diferentes condiciones ambientales y geográficas. Esto permitiría mejorar la capacidad de generalización del modelo y reducir posibles errores de clasificación en escenarios reales. De igual forma, podría evaluarse la implementación de modelos más avanzados o la optimización del actual para su ejecución en dispositivos embebidos, reduciendo el consumo de recursos.

Otra recomendación importante es la integración del sistema con plataformas oficiales de protección civil o autoridades ambientales, de manera que las alertas generadas puedan ser canalizadas directamente a instancias responsables de la atención de emergencias. Finalmente, se sugiere realizar pruebas de campo a mayor escala, con el fin de validar el desempeño del sistema en entornos reales y bajo condiciones variables, fortaleciendo así su viabilidad como una solución aplicada.

5 FUENTES BIBLIOGRÁFICAS

Bibliografía

(s.f.).

Aguilera Sánchez, M. (2015). *El régimen jurídico de los incendios forestales: Especial referencia al contexto jurídico y territorial de Cataluña*. España: Publicacions URV.

Aliaga Casaletti, A. (2020). *Sistema para la detección temprana de incendios forestales utilizando visión artificial*. Universidad Siglo 21.

autores, V. (2023). *Fórmulas y funciones matemáticas con excel*. Colombia: Ediciones de la U.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Bottini, C. (2025). *Python para IA - Vol.4: Redes neuronales y Deep Learning. procesamiento del lenguaje natural. Visión por computadora*. (n.p.): RedUSERS Premium.

Brenes, Á., & Saborío, V. F. (1995). *Elementos de climatología: su aplicación didáctica a Costa Rica*. Costa Rica: Editorial Universidad Estatal a Distancia (EUNED).

Caballero, R., & Martín, E. (2025). *40 conceptos clave sobre inteligencia artificial*. . Los Libros de La Catarata.

Cayssials, R. (2014). *Sistemas embebidos en FPGA*. Alpha Editorial.

Cayssials, R. (2014). *Sistemas embebidos en FPGA*. Colombia: Alpha Editorial.

Chicano Tejada, E. (2025). *Gestión de servicios en el sistema informático*. España: IC Editorial.

Comisión Nacional Forestal. (2025). Incendios. Sistema Nacional de Información Forestal. *Gobierno de México*.

Contreras, L., Fuentes, H., & Rodríguez, J. (2024). *Algoritmos supervisados y de ensamble con python: Implementación y estrategia de optimización*. Colombia: Ediciones de la U.

De la Fraga, L. G. (2001). *Introducción Curso: Procesamiento Digital de Imágenes*. Instituto Tecnológico de Estudios Superiores de Monterrey.

DJI. (10 de march de 2026). *DJI Mini 3*. Obtenido de DJI Official.

Dulhare, N., & Houssein, E. H. (2025). *Deep Learning and Computer Vision: Models and Biomedical Applications: Volume 1*. Springer Nature Singapore.

Egmond, M. (2018). *Wim Kieft: De terugkeer*. Overamstel Uitgevers.

Fernández, I. G. (2024). *Detección temprana de incendios forestales con el uso de cámaras utilizando redes neuronales profundas*. Centro Nacional de Inteligencia Artificial (CENIA), Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas, Departamento de Ingeniería Eléctrica.

- Flórez López, R., & Fernández Fernández, J. (2008). *Las Redes Neuronales Artificiales*. España: Netbiblo.
- Galeano Prieto, J., & Pastor Ruiz, J. (2022). *Laboratorio de Computación para profesores de Física y Química*. España: U.P.M. Press.
- Gambín Martínez, J. A. (2023). *so de drones para el control y extinción de incendios forestales*. Universidad Politécnica de Cartagena, Escuela Técnica Superior de Ingeniería Industrial.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. doi:9780262035613
- Granados, M. (2024). *Microcontroladores PIC en lenguaje BASIC*.
- Hernández Bejarano, M. (2025). *Python: con orientación a objetos y al análisis de datos*. Ediciones de la U.
- Hinojosa Gutiérrez, Á. (2016). *Python Paso a paso*. España, RA-MA Editorial.
- Howse, J., & Minichino, J. (2020). *Learning OpenCV 4 Computer Vision with Python 3: Get to Grips with Tools, Techniques, and Algorithms for Computer Vision and Machine Learning*. Packt Publishing.
- Lahoz-Beltra, R. (2010). *Bioinformática: Simulación, vida artificial e inteligencia artificial*. . Editorial Díaz de Santos, S.A..
- Leandro, L. L. (2024). *Modelo para la segmentación y análisis de gravedad de áreas afectadas por incendios forestales usando redes neuronales convolucionales e imágenes satelitales*. Pontificia Universidad Católica del Perú Tesis de licenciatura en Ingeniería Informática. Obtenido de <http://hdl.handle.net/20.500.12404/28020>
- León Castillo, Y. (2024). *Aplicación de visión por computadora e inteligencia artificial para la prevención de incendios forestales en áreas protegidas del litoral ecuatoriano*. Revista Internacional de Investigación y Desarrollo Global.
- Li, H., Chaudhari, P., Yang, H., Lam, M., Ravichandran, A., Bhotika, R., & Soatto, S. (18 de february de 2020). *Rethinking the Hyperparameters for Fine-tuning*. Obtenido de arXiv : <https://arxiv.org/abs/2002.11770>
- Liu, Y., & Mehta, S. (2019). *Hands-On Deep Learning Architectures with Python: Create Deep Neural Networks to Solve Computational Problems Using TensorFlow and Keras*. Packt Publishing.
- López Sotelo, J. (2023). *Deep Learning: teoría y aplicaciones*. . Marcombo. doi:9788426736079
- López, C. P. (2024). *Ciencia de datos a través de Python*. Garceta Grupo Editorial.
- Morais, J. (2023). *ESP32 con IDF*. Editora NCB.
- Mott, R. L., & Samaniego, A. H. (1996). *Mecánica de fluidos aplicada*. México: Pearson Educación.
- Muñoz, O. Q. (2019). *Internet de las Cosas (IoT)*. . Estados Unidos: Ibukku, LLC.

- Murphy, K. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.
- Novera, K. (2025). *Inteligencia Artificial: De Cero Al Uso Práctico*. . Clube de Autores.
- OTAÑO, J. (1868). *Estación Meteorológica de Burgos. Resultados de las observaciones del año 1867*. España: (n.p.).
- Pausas, J. G. (2012). *Incendios forestales: Una introducción a la ecología del fuego*. Los Libros de la Catarata / CSIC. Madrid, España. doi:ISBN: 978-84-8319-692-7
- Pedre, S. (2017). *Sistemas embebidos*. Departamento de computación FCEN UBA.
- Perez Ugena, M. (2024). *La inteligencia artificial: definición, regulación y riesgos para los derechos fundamentales*. Facultad de Derecho, Universidad de Deusto.
- Piora, C. (2021). *Google Classroom*. RedUsers.
- Probst, P., Bischl, B., & Boulesteix, A.-L. (26 de february de 2018). *Aprendizaje automático* . Obtenido de arXiv : <https://arxiv.org/abs/1802.09596>
- R, S. E., & Vega, M. N. (2025). *Diseño gráfico y fábrica visual. El poder de la comunicación visual en la industria*. Qartuppi.
- Ramírez Gil, C. (2023). *Programación de Inteligencia Artificial. Curso Práctico*. . Ra-Ma S.A. Editorial y Publicaciones.
- Raschka, S. (2019). *Python machine learning: aprendizaje automático y aprendizaje profundo con Python, scikit-learn y TensorFlow*. . Marcombo.
- Rivas, A. R. (2010). *Estudios de valoración energética de combustibles forestales para la prevención de incendios forestales en la Sierra de la Primavera (Jalisco, México) mediante calorimetría de combustión y ensayos de inflamabilidad*. Universidade de Santiago de Compostela, Servizo de Publicacións e Intercambio Científico.
- Rodríguez Rivas, A. (2010). *Estudios de valoración energética de combustibles forestales para la prevención de incendios forestales en la Sierra de la Primavera (Jalisco, México) mediante calorimetría de combustión y ensayos de inflamabilidad*. España: Universidade de Santiago de Compostela, Servizo de Publicacións e Intercambio Científico.
- Rodriguez, D. (2025). *APRENDE TENSORFLOW: Construye Modelos Reales con Aprendizaje Profundo. Desde los Fundamentos hasta las Aplicaciones Prácticas*. . (n.p.): StudioD21.
- Rodríguez, E. B., León, E. L., Bernal, I. G., & Gómez, H. R. (2025). *Aplicación de inteligencia artificial y tecnologías para la transformación digital: Pymes, desarrollo sostenible e inclusión social*. . Editorial Fontamara S. A. de C. V.
- Romero, B. P., & Janeiro, A. D. (2025). *MicroPython en proyectos. Una introducción práctica a la programación de microcontroladores*. . España: Marcombo.

- Rubio Ahumada, F. (2021). *Curso práctico de Google Data Studio: Guía interactiva para crear informes personalizados y creativos*. España: ANAYA MULTIMEDIA.
- Ruiz, L. V., & Blanco, J. L. (2004). *Incendios Forestales en Mexico*. México: Universidad Nacional Autónoma. doi:9789703218431
- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. Pearson.
- Singh, H., & Kumar, N. (2025). *Forest Fire and Climate Change: Insights Into Science*. . Springer Nature Switzerland.
- Sirisha, U., Praveen, S., & Srinivasu, P. (2023). *Statistical Analysis of Design Aspects of Various YOLO-Based Deep Learning Models for Object Detection*. Sistema de inteligencia computacional Int J 16.
- Soria Olivas, E., Gamero Cruz, R., Castillo Caballero, B., & Cano Michelena, P. (2023). *Sistemas de Aprendizaje Automático*. Ra-Ma S.A. Editorial y Publicaciones.
- Soriano Monserrate, D. X. (2024). *Modelo de gestión para sistemas de vigilancia por amenazas de incendios forestales del cantón Guayaquil basado en inteligencia artificial e IoT*. Universidad Politécnica Salesiana, Sede Guayaquil.
- STANLEY, J., March, A., & Ogloff, J. (2025). *Sintiendo el calor: Perspectivas internacionales sobre la prevención de la ignición de incendios forestales*. Vernon Art and Science Incorporated.
- Thedy Cornejo, B. I. (2024). *Implementación de un sistema de multitarget tracking con una cámara infrarroja de alta resolución*. Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas, Departamento de Ingeniería Eléctrica. Obtenido de <https://repositorio.uchile.cl/xmlui/bitstream/handle/2250/200834/Implementacion-de-un-sistema-de-multitarget-tracking-con-una-camara-infrarroja-de-alta-resolucion.pdf?sequence=1&isAllowed=y>
- Torres, J. (2018). *DEEP LEARNING Introducción práctica con Keras*. España: Lulu.com.
- Unmanned Aircraft Systems: UAVS Design, D. a. (2010). *Unmanned Aircraft Systems*. Wiley.
- Vicente Vírveda, J., González Arias, J., Parra Rodríguez, F., & Beltrán Pascual, M. (2019). *Métodos de Data Science aplicados a la Economía y a la Dirección y Administración de Empresas*. España: UNED.
- Vollmer, M., & Möllmann, K. (2010). *Infrared Thermal Imaging: Fundamentals, Research and Applications*. . Wiley.
- Witten, I., Frank, E., & Hall, M. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.
- Zaragoza, M. I., & Bartolucci, G. (2008). *La imagen digital: Expresión artística y modalidades de producción a través de la utilización de herramientas tecnológicas*. Facultad de Artes y Diseño, Universidad Nacional de Cuyo Mendoza, Argentina.

6 ANEXOS

Anexo 1

MANUAL TÉCNICO

MANUAL TÉCNICO SISTEMA DE DETECCIÓN DE INCENDIOS FORESTALES.

1. Introducción

El presente Manual Técnico tiene como finalidad documentar de manera clara, ordenada y detallada el funcionamiento, la arquitectura y los componentes del sistema de detección temprana de incendios forestales desarrollado durante la residencia profesional.

2. Descripción general del sistema

El sistema de detección de incendios forestales está diseñado como una solución modular y escalable, capaz de monitorear variables ambientales críticas y analizar imágenes para identificar indicios de fuego. Su funcionamiento se basa en la interacción de tres bloques principales: el sistema embebido de monitoreo, la infraestructura de comunicación y almacenamiento, y la plataforma de análisis y visualización.

Desde el punto de vista funcional, el sistema realiza la lectura continua de variables como temperatura y humedad, transmite estos datos a la nube mediante conexión Wi-Fi y, de forma paralela, permite el análisis de imágenes mediante un modelo de inteligencia artificial



Ilustración 43 Arquitectura del sistema con sistemas embebidos (elaboración propia)

entrenado para detectar incendios forestales. Cuando se detecta una condición de riesgo, el sistema emite alertas automáticas al usuario.

3. Arquitectura del sistema

La arquitectura del sistema se encuentra dividida en capas, lo que facilita su comprensión, mantenimiento y posible expansión futura. Cada capa cumple una función específica dentro del proceso de detección y alerta.

3.1 Capa de adquisición de datos

Esta capa está conformada por los sensores ambientales y el microcontrolador ESP32. Los sensores se encargan de medir las variables físicas del entorno, mientras que el ESP32 actúa como unidad central de procesamiento, recolectando y organizando la información.

3.2 Capa de comunicación

La transmisión de los datos se realiza mediante conectividad Wi-Fi, permitiendo el envío de la información a servicios en la nube. Esta capa es fundamental para garantizar el monitoreo remoto y el acceso en tiempo real a los datos recolectados.

3.3 Capa de procesamiento y visualización

En esta capa se integran los servicios de almacenamiento, la interfaz web y el modelo de inteligencia artificial. Aquí se procesan los datos recibidos, se ejecuta el análisis visual y se presentan los resultados al usuario de forma clara e intuitiva.

4. Componentes del sistema

4.1 Componentes de hardware

El sistema físico está compuesto por dispositivos electrónicos de bajo consumo, seleccionados por su confiabilidad y facilidad de integración. Estos componentes permiten la adquisición de datos y la operación autónoma del sistema.

Tabla 8 descripción de los sensores que se utilizaron para la estación (elaboración propia)

Componente	Características	Función dentro del sistema
-------------------	------------------------	-----------------------------------

Sensor de temperatura DHT11	Sensor digital de temperatura y humedad	Mide condiciones ambientales para apoyar la detección temprana de incendios.
ESP32	Microcontrolador de bajo consumo con Wi-Fi y Bluetooth integrados; alto rendimiento	Controla los sensores, procesa datos y envía información a la nube
Pantalla LCD	Pantalla alfanumérica de bajo consumo energético	Visualiza temperatura, humedad y estado del sistema
Batería de litio	Alta densidad energética y recargable	Alimenta el sistema en ausencia de radiación solar
Fototransistor	Sensor sensible a radiación luminosa e infrarroja	Detecta la presencia de flama mediante cambios de luz
Zumbador de 5 V	Dispositivo de alerta acústica	Emite señal sonora ante detección de incendio
Panel solar	Fuente de energía renovable de corriente continua	Suministra energía y recarga la batería

4.2 Componentes de software

El software del sistema incluye el firmware del ESP32, la plataforma de almacenamiento en la nube, la interfaz web y el modelo de inteligencia artificial. Cada uno de estos elementos cumple un rol específico dentro del flujo de información.

Tabla 9 descripción de los lenguajes de programación que se utilizaron (elaboración propia)

Software / Herramienta	Lenguaje	Función dentro del sistema
Arduino IDE	C / C++	Programación y carga del firmware en el ESP32 para la lectura de sensores y control del sistema

Python	Python	Procesamiento de datos, entrenamiento del modelo de inteligencia artificial y desarrollo de la aplicación
Plataforma web (Streamlit)	Python	Interfaz gráfica para la visualización de datos y detección de incendios mediante imágenes
Modelo de IA (YOLOv5)	Python	Detección automática de fuego y humo mediante visión por computadora

5. Funcionamiento del sistema

El sistema opera de manera continua y automática. En primer lugar, los sensores ambientales capturan las variables del entorno y envían la información al microcontrolador. Posteriormente, el ESP32 procesa los datos y los transmite a la nube, donde son almacenados y visualizados.

De forma paralela, el usuario puede cargar imágenes en la interfaz web para su análisis mediante el modelo de visión artificial. En caso de detectar un incendio o condiciones de riesgo, el sistema genera una alerta inmediata (ejemplo en la Ilustración 26).

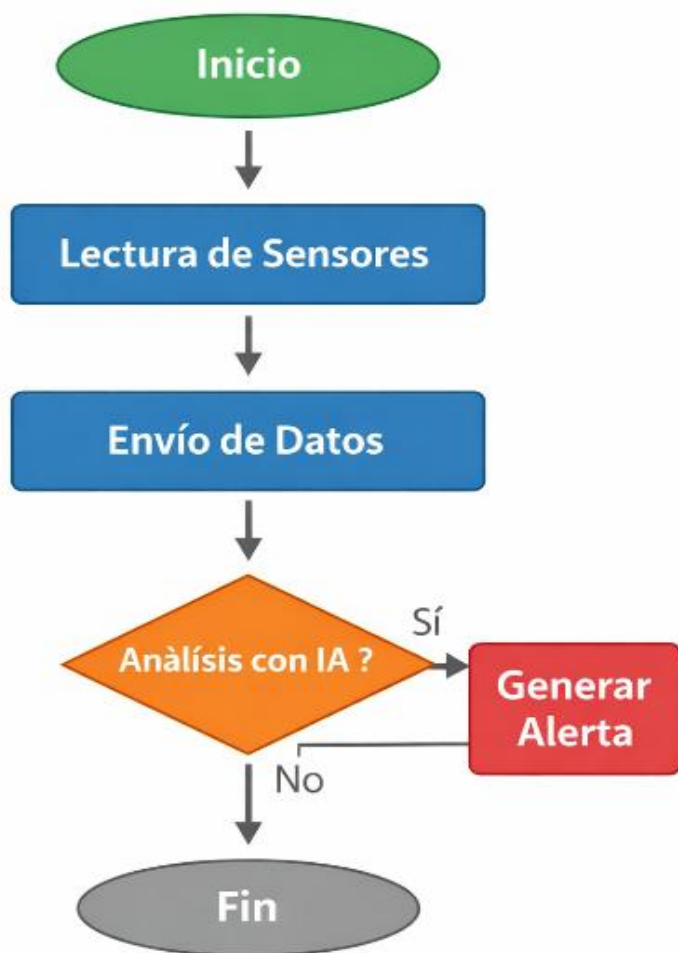


Ilustración 44 Diagrama de flujo del sistema (elaboracion propia)

6. Instalación y configuración

6.1 Instalación del hardware

La instalación del hardware debe realizarse asegurando una correcta conexión entre los sensores, el microcontrolador y el sistema de alimentación. Es importante verificar la polaridad de los dispositivos y asegurar conexiones firmes como se muestra en la ilustración 52.

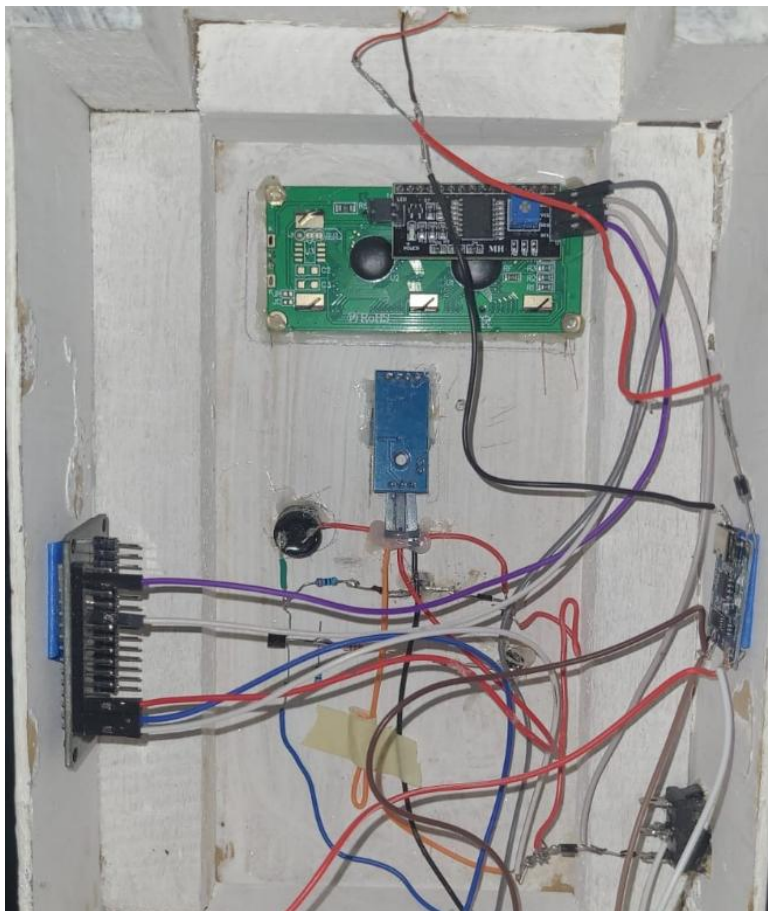


Ilustración 45 ensamblaje del circuito (elaboración propia)

6.2 Configuración del software

La configuración del software incluye la carga del firmware en el ESP32, la conexión a la red Wi-Fi y la configuración de los servicios en la nube. Esta etapa es clave para garantizar el correcto funcionamiento del sistema.

7. Interfaz de usuario

La interfaz web permite al usuario monitorear en tiempo real las variables ambientales, visualizar registros históricos y analizar imágenes mediante inteligencia artificial (Ilustración 28). Su diseño prioriza la claridad de la información y la facilidad de uso.

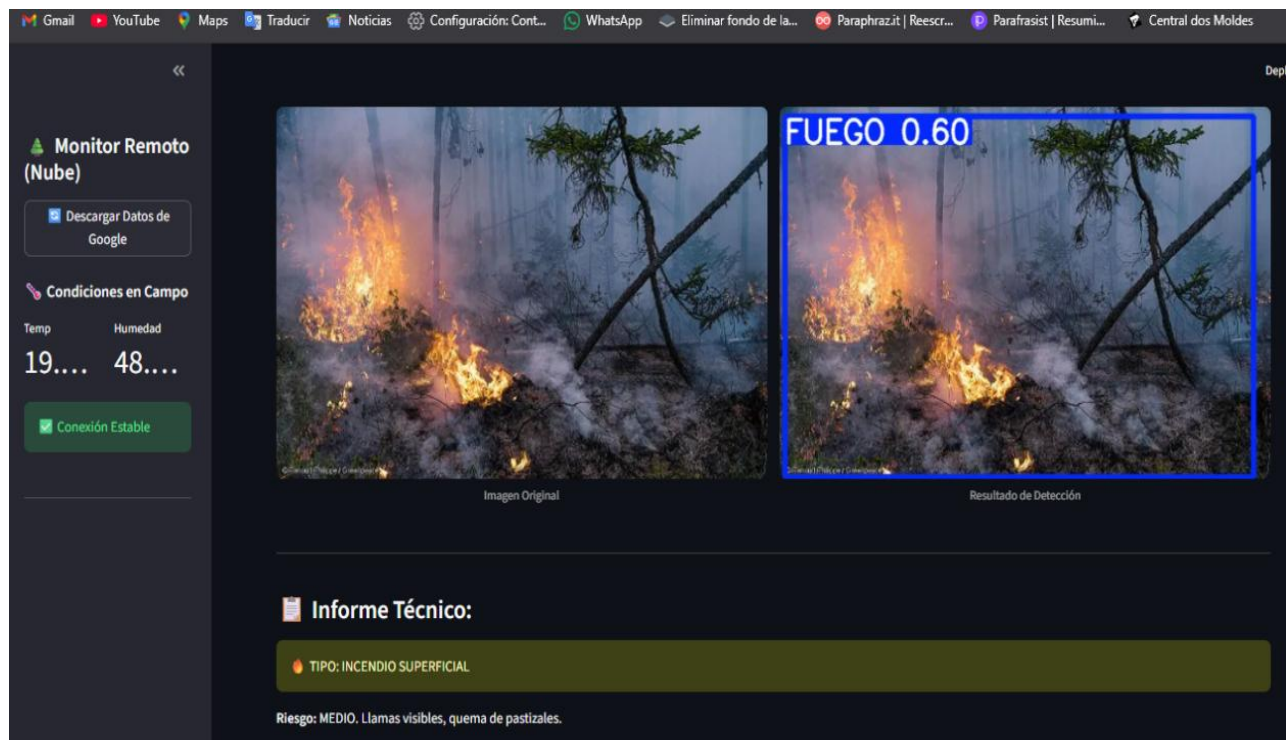


Ilustración 46 pagina web conectada con la estación meteorológica funcionando (elaboración propia)

8. Alertas y notificaciones

El sistema cuenta con un mecanismo de alertas que notifica al usuario cuando se detectan condiciones críticas. Estas alertas pueden enviarse mediante mensajes SMS, permitiendo una respuesta oportuna ante un posible incendio como se muestra en la Ilustración 29.

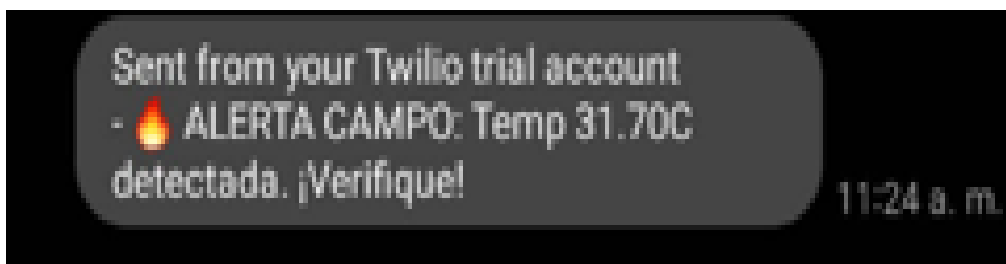


Ilustración 47 prueba de envío de notificaciones de alerta a dispositivos móviles (elaboración propia)

9. Mantenimiento y consideraciones técnicas

Para asegurar un funcionamiento óptimo del sistema, se recomienda realizar revisiones periódicas de los sensores, verificar la alimentación eléctrica y mantener actualizado el software. Asimismo, es importante proteger el sistema de condiciones ambientales extremas que puedan afectar su operación.

10. Posibles mejoras y escalabilidad

El sistema fue diseñado con un enfoque modular, lo que permite la incorporación de nuevos sensores, la mejora del modelo de inteligencia artificial y la integración con plataformas externas. Estas mejoras pueden incrementar la precisión, confiabilidad y alcance del sistema en aplicaciones reales.

11. Conclusión

El presente manual técnico documenta de forma integral el sistema de detección de incendios forestales, proporcionando una base sólida para su operación, mantenimiento y futura expansión. La combinación de IoT e inteligencia artificial demuestra ser una alternativa viable y con alto potencial de impacto en la prevención de incendios forestales.

Anexo 2

MANUAL DE USUARIO DEL SISTEMA DE DETECCIÓN DE INCENDIOS FORESTALES

1. introducción

El presente Manual de Usuario tiene como objetivo guiar al usuario final en el uso correcto del sistema de detección temprana de incendios forestales. Este documento está dirigido a personas con conocimientos básicos en el uso de computadoras y dispositivos móviles, sin necesidad de contar con formación técnica especializada.

El sistema fue diseñado para facilitar el monitoreo de variables ambientales y el análisis de imágenes mediante inteligencia artificial, permitiendo al usuario identificar posibles riesgos de incendio de manera oportuna y sencilla.

2. Requisitos para el uso del sistema

Para el correcto funcionamiento del sistema, el usuario debe contar con los siguientes elementos:

Dispositivo con acceso a Internet (computadora, tablet o teléfono inteligente).

Navegador web actualizado.

Conexión activa a la red donde se encuentra el sistema.

Credenciales de acceso a la plataforma (en caso de aplicar).

3. Acceso al sistema

El acceso al sistema se realiza a través de la interfaz web. Para ello, el usuario debe ingresar la dirección correspondiente en el navegador y, en caso de ser necesario, introducir sus credenciales.

Una vez dentro del sistema, el usuario podrá visualizar la información principal del monitoreo ambiental y acceder a las diferentes funciones disponibles.

4. Pantalla principal

La pantalla principal concentra la información más relevante del sistema. En esta sección el usuario puede observar, en tiempo real, los valores de las variables ambientales monitoreadas.

Los datos se muestran de forma clara mediante indicadores numéricos y, en algunos casos, gráficas que permiten interpretar fácilmente el comportamiento del entorno.



Ilustración 48 pantalla principal de la página web (elaboración propia)

5. Monitoreo de variables ambientales

En este apartado el usuario puede consultar de manera detallada las variables ambientales registradas por el sistema, tales como temperatura y humedad. La información se actualiza automáticamente conforme se reciben nuevos datos.

El usuario puede identificar tendencias o condiciones anómalas que indiquen un posible riesgo, lo cual resulta fundamental para la prevención de incendios

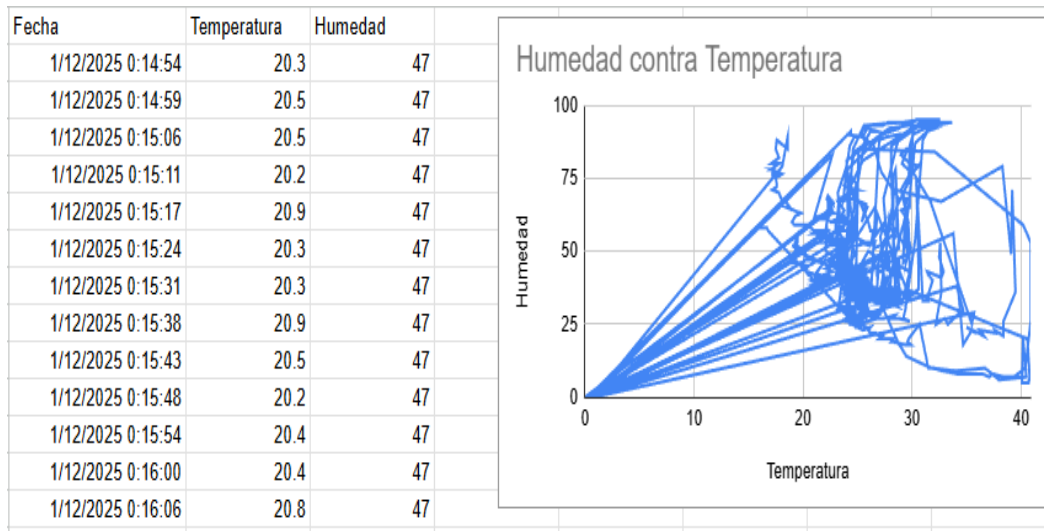


Ilustración 49 captura de datos de los sensores conectados (elaboración propia)

forestales.

6. Análisis de imágenes con inteligencia artificial

El sistema permite al usuario cargar imágenes para su análisis mediante un modelo de inteligencia artificial especializado en la detección de incendios forestales. Para realizar esta acción, el usuario debe seleccionar una imagen desde su dispositivo y esperar el resultado del análisis.

Una vez procesada la imagen, el sistema muestra el resultado indicando si se detecta o no la presencia de un incendio, así como el nivel de confianza del modelo.



Ilustración 50 análisis del incendio con la red neuronal



Ilustración 51 Resultado del análisis con la red neuronal

6. Alertas y notificaciones

Cuando el sistema detecta condiciones críticas o la posible presencia de un incendio, se genera una alerta automática. Estas alertas permiten al usuario actuar de manera inmediata y tomar las medidas correspondientes.

Las notificaciones pueden recibirse directamente en la plataforma o a través de mensajes enviados al dispositivo móvil del usuario.

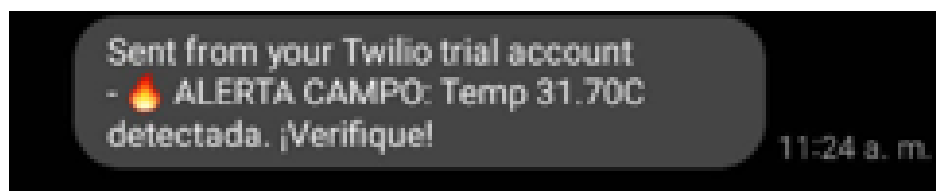


Ilustración 52 prueba de envío de notificaciones de alerta a dispositivos móviles (elaboración propia)

7. Recomendaciones de uso

Para un uso adecuado del sistema, se recomienda revisar periódicamente la información mostrada en la plataforma, verificar la calidad de las imágenes cargadas y mantener una conexión a Internet estable.

Asimismo, es importante considerar que el sistema funciona como una herramienta de apoyo para la detección temprana, por lo que las alertas generadas deben ser evaluadas y confirmadas por personal capacitado.

8. Solución de problemas comunes

En caso de que el sistema no muestre información actualizada, se recomienda verificar la conexión a Internet. Si no es posible acceder a la plataforma, se debe revisar que la dirección ingresada sea correcta.

Ante cualquier falla persistente, se sugiere contactar al personal responsable del mantenimiento del sistema.

9. Conclusión

El Manual de Usuario proporciona las instrucciones necesarias para operar el sistema de detección de incendios forestales de forma correcta y segura. Su uso adecuado contribuye a una mejor interpretación de la información y a una respuesta más oportuna ante posibles riesgos ambientales.

Anexo 3

CRONOGRAMA

Tabla 10 cronograma de actividades

Actividad	Septiembre				Octubre				Noviembre				Diciembre				Enero			
SEMANAS	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Investigación teórica y documental sobre incendios forestales																				
Recolección y análisis de datasets para entrenar la red neuronal																				
Diseño de la arquitectura del sistema (hardware + software)																				
Configuración de cámaras y pruebas preliminares de captura																				
Desarrollo y entrenamiento de la red neuronal para detección de fuego																				
Clasificación de tipos de incendios según parámetros visuales																				
Desarrollo de la página web y panel de monitoreo																				
Integración de alertas en tiempo real (notificaciones)																				
Pruebas de integración del sistema completo																				
Validación con posibles usuarios (autoridades o comunidad piloto)																				
Entrega final del proyecto y documentación																				

ANEXO 4



ANEXO 5



ANEXO 6



Anexo 7



ATICTAC

Academia Tamaulipeca de Investigación Científica y Tecnológica, A.C.

Otorga la presente

Constancia

a: **ORDOÑEZ GONZALEZ BRAYAN**

Que le acredita con la Membresía de la Sección Juvenil de esta Academia por el periodo del 1 de enero al 31 de diciembre de 2026.

Número de Membresía
JA26007



ATICTAC
ACADEMIA TAMAULIPECA DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA A.C.

Dr. Fabio Chalé Lara
Presidente de la Academia Tamaulipeca de Investigación Científica y Tecnológica, A.C.

Anexo 8

Código del esp32

```
#include <WiFi.h>

#include <HTTPClient.h>

#include <DHT.h>

#include <LiquidCrystal_I2C.h>

// 1. CONFIGURACIÓN CAMPO (HOTSPOT CELULAR)

const char* ssid = "Motorrouter";

const char* password = "12345678";

String googleScriptURL =
"https://script.google.com/macros/s/AKfycbwVS9EgazoqToKvLUCF2J1hDnyF8QvaiM1F
W9tSfD76oSDtoFUkPqaRJjmlxfR6vSG2oQ/exec";

#define DHTPIN 15

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {

  Serial.begin(115200);

  dht.begin();

  lcd.init();

  lcd.backlight();
```

```

WiFi.begin(ssid, password);

lcd.print("Conectando red..");


while(WiFi.status() != WL_CONNECTED) {

    delay(500);

    Serial.print(".");

}

lcd.clear();

lcd.print("Conectado!");

}


void loop() {

    delay(5000); // Enviar cada 5 seg


    float h = dht.readHumidity();

    float t = dht.readTemperature();

    if (isnan(h) || isnan(t)) return;


    lcd.setCursor(0,0);

    lcd.print("T:" + String(t,1) + "C H:" + String(h,0) + "%");

```

```

if(WiFi.status() == WL_CONNECTED){

    HTTPClient http;

    http.setFollowRedirects(HTTPC_FORCE_FOLLOW_REDIRECTS);

    http.setTimeout(15000);


    String urlFinal = googleScriptURL + "?temperatura=" + String(t) + "&humedad=" +
String(h);


    http.begin(urlFinal);

    int httpCode = http.GET(); // Usamos GET para evitar error 400

    lcd.setCursor(0,1);

    if(httpCode == 200 || httpCode == 302) {

        lcd.print("Enviado OK  ");

    } else {

        lcd.print("Err:" + String(httpCode) + "  ");

    }

    http.end();

}

}

```

Anexo 9

Código de la página web

```
import streamlit as st

import subprocess

import pathlib

from PIL import Image

import sys

import pandas as pd

import io

import json

import shutil

import time

from collections import Counter

from datetime import datetime

import tempfile

# =====

# CONFIGURACIÓN

# =====

MODEL_FILE = "best.pt"

OUTPUT_DIR = pathlib.Path("runs/detect_app")

UPLOAD_DIR = pathlib.Path("temp_uploads")
```

```
RESULT_DIR = OUTPUT_DIR / "results"
```

```
# Crear directorios si no existen
```

```
UPLOAD_DIR.mkdir(parents=True, exist_ok=True)
```

```
RESULT_DIR.mkdir(parents=True, exist_ok=True)
```

```
# =====
```

```
# GOOGLE SHEETS FETCHER
```

```
# =====
```

```
SHEET_ID = "1iM8BypTFDacmp43Keq-G9w3JziY4c wd7St5cfKw8h5w"
```

```
URL_DATOS
```

=

```
f"https://docs.google.com/spreadsheets/d/{SHEET_ID}/export?format=csv"
```

```
@st.cache_data(ttl=60)
```

```
def obtener_datos_nube():
```

```
    try:
```

```
        df = pd.read_csv(URL_DATOS)
```

```
        temp = float(df.iloc[-1, 1]) if pd.notna(df.iloc[-1, 1]) else None
```

```
        hum = float(df.iloc[-1, 2]) if pd.notna(df.iloc[-1, 2]) else None
```

```
        return temp, hum
```

```
    except:
```

```
        return None, None
```

```

# =====

# DETECTOR (YOLOv5)

# =====

class FireDetector:

    def __init__(self, model_path=MODEL_FILE, output_dir=RESULT_DIR):

        self.model_path = model_path

        self.output_dir = output_dir

        self.class_map = {

            0: "FUEGO_COPA",

            1: "FUEGO_SUPERFICIAL",

            2: "HUMO",

            3: "NO_FUEGO"

        }

        self.risk_messages = {

            0: "🔥 **Riesgo extremo** – Incendio de copa, evacuación inmediata.",

            1: "⚠️ **Riesgo alto** – Quema de pastizales, propagación rápida.",

            2: "🌀 **Presencia de humo** – Monitorear posibles focos de incendio.",

            3: "✅ **Zona segura** – No se detectó fuego."

        }

    def detect(self, image_path):

```

```
"""Ejecuta YOLO y devuelve la ruta de la imagen resultado y las detecciones."""
```

```
temp_image_path = image_path
```

```
result_name = temp_image_path.stem
```

```
project_dir = self.output_dir / result_name
```

```
project_dir.mkdir(parents=True, exist_ok=True)
```

```
command = [
```

```
    sys.executable, "detect.py",
```

```
    "--weights", str(self.model_path),
```

```
    "--source", str(temp_image_path.resolve()),
```

```
    "--project", str(self.output_dir),
```

```
    "--name", str(result_name),
```

```
    "--exist-ok",
```

```
    "--save-txt",
```

```
    "--save-conf"
```

```
]
```

```
result = subprocess.run(command, capture_output=True, text=True)
```

```
if result.returncode != 0:
```

```
    raise RuntimeError(f"YOLO error: {result.stderr}")
```

```
result_image = project_dir / temp_image_path.name
```

```
labels_file = project_dir / 'labels' / (temp_image_path.stem + '.txt')
```

```
return result_image, labels_file
```

```
def parse_labels(self, labels_path, img_width, img_height):
```

```
    """Lee el archivo de etiquetas y devuelve lista de detecciones."""
```

```
    detecciones = []
```

```
    if not labels_path.exists():
```

```
        return detecciones
```

```
    with open(labels_path, 'r') as f:
```

```
        for line in f:
```

```
            parts = line.strip().split()
```

```
            if len(parts) < 5:
```

```
                continue
```

```
            clase = int(parts[0])
```

```
            x_c = float(parts[1])
```

```
            y_c = float(parts[2])
```

```
            w_norm = float(parts[3])
```

```
            h_norm = float(parts[4])
```

```
            conf = float(parts[5]) if len(parts) >= 6 else None
```

```
            w_px = w_norm * img_width
```

```
            h_px = h_norm * img_height
```

```
            area = w_px * h_px
```

```

    detecciones.append({
        'clase': clase,
        'conf': conf,
        'area': area,
        'bbox': (x_c, y_c, w_px, h_px)
    })

return detecciones

```

```

def get_summary(self, detecciones):

```

```

    """Devuelve un resumen estadístico de las detecciones."""

```

```

    if not detecciones:

```

```

        return None, None, None, None

```

```

    clases = [d['clase'] for d in detecciones]

```

```

    conf_por_clase = {}

```

```

    area_por_clase = {}

```

```

    for d in detecciones:

```

```

        c = d['clase']

```

```

        area_por_clase[c] = area_por_clase.get(c, 0) + d['area']

```

```

        if d['conf'] is not None:

```

```

            if c not in conf_por_clase:

```

```

                conf_por_clase[c] = []

```

```

        conf_por_clase[c].append(d['conf'])

# Clase principal: prioriza confianza promedio, si no, la más frecuente
if conf_por_clase:

    conf_promedio = {c: sum(vals)/len(vals) for c, vals in conf_por_clase.items()}

    clase_principal = max(conf_promedio, key=conf_promedio.get)

else:

    contador = Counter(clases)

    clase_principal = contador.most_common(1)[0][0]

total_area = sum(d['area'] for d in detecciones)

return clase_principal, total_area, area_por_clase, clases

# =====

# UTILIDADES

# =====

def clean_temp_files(directory=UPLOAD_DIR):

    """Elimina todos los archivos temporales."""

    if directory.exists():

        shutil.rmtree(directory)

        directory.mkdir(parents=True, exist_ok=True)

```

```
def save_report(detecciones, image_name, total_area, porcentaje, clase_principal,
class_map):
```

```
    """Genera un informe en formato JSON."""
```

```
    report = {
```

```
        "imagen": image_name,
```

```
        "timestamp": datetime.now().isoformat(),
```

```
        "total_detecciones": len(detecciones),
```

```
        "area_total_px": total_area,
```

```
        "porcentaje_area": porcentaje,
```

```
        "clase_principal": {
```

```
            "id": clase_principal,
```

```
            "nombre": class_map.get(clase_principal, "DESCONOCIDO")
```

```
        },
```

```
        "detecciones": detecciones
```

```
    }
```

```
    return json.dumps(report, indent=2)
```

```
# =====
```

```
# INTERFAZ DE USUARIO (UI COMPONENTS)
```

```
# =====
```

```
def mostrar_sidebar():
```

```
    with st.sidebar:
```

```

st.markdown("## 🔥 **FireScope**")

st.markdown("##### *Monitoreo piro-inteligente*")

st.markdown("---")

if st.button("🔄 **Refrescar datos**", use_container_width=True):

    st.cache_data.clear()

    st.success("¡Datos actualizados!")

st.markdown("### 🚒 **Condiciones en Campo**")

temp_actual, hum_actual = obtener_datos_nube()

col_a, col_b = st.columns(2)

if temp_actual is not None:

    col_a.metric("🚒 **Temp**", f"{temp_actual} °C")

    col_b.metric("💧 **Humedad**", f"{hum_actual} %")

    if temp_actual > 40:

        st.error("⚠️ ¡TEMPERATURA CRÍTICA!")

    else:

        col_a.metric("🚒 **Temp**", "❌ Sin dato")

        col_b.metric("💧 **Humedad**", "❌ Sin dato")

    st.warning("⚠️ No hay datos válidos.")

```

```

st.markdown("---")

if st.button(" 🧹 **Limpiar archivos temporales**", use_container_width=True):

    clean_temp_files()

    st.success("Archivos temporales eliminados.")

st.markdown("---")

st.caption(" 📊 Datos vía Google Sheets | 🔥 YOLOv5")

def mostrar_card(titulo, contenido):

    st.markdown(f"""

    <div class="card">

        <h3>{titulo}</h3>

        {contenido}

    </div>

    """, unsafe_allow_html=True)

# =====

# MAIN APP

# =====

st.set_page_config(

    page_title=" 🔥 FireScope - Detector de Incendios",

    page_icon=" 🔥 ",

```

```

        layout="wide",

        initial_sidebar_state="expanded"

    )

# CSS (igual que antes, lo omito por brevedad, pero debe estar aquí)

# ... (incluir el CSS que ya tenías)


mostrar_sidebar()


# =====

# SECCIÓN PRINCIPAL

# =====

st.markdown('<div class="card">', unsafe_allow_html=True)

st.markdown("## 🔥 **FireScope** – *Visión artificial contra incendios*")

st.markdown("_Detección en tiempo real + sensores remotos_")

st.markdown('</div>', unsafe_allow_html=True)


# Estado de la sesión para historial

if 'historial' not in st.session_state:

    st.session_state.historial = []


# Subida de imágenes

```

```

st.markdown('<div class="card">', unsafe_allow_html=True)

st.markdown("### 🖼️ **Detección en Imagen**")

st.markdown("Sube una o varias imágenes (JPG, JPEG, PNG).")


uploaded_files = st.file_uploader(

    "📁 **Arrastra o selecciona imágenes**",

    type=["jpg", "jpeg", "png"],

    accept_multiple_files=True,

    help="La IA procesará cada imagen y mostrará los resultados."

)

```

```

if uploaded_files:

    # Botón para procesar todas

    if st.button("🚀 **Procesar todas las imágenes**", use_container_width=True):

        detector = FireDetector()

        progress_bar = st.progress(0)

        status_text = st.empty()


        resultados = []

        total = len(uploaded_files)


        for idx, uploaded_file in enumerate(uploaded_files):

```

```

status_text.text(f"Procesando {uploaded_file.name} ({idx+1}/{total})...")

# Guardar imagen temporal

temp_path = UPLOAD_DIR / uploaded_file.name

with open(temp_path, "wb") as f:

    f.write(uploaded_file.getbuffer())

# Detectar

try:

    result_image, labels_file = detector.detect(temp_path)

    image = Image.open(temp_path)

    img_w, img_h = image.size

    detecciones = detector.parse_labels(labels_file, img_w, img_h)

    clase_principal, total_area, area_por_clase, clases =
detector.get_summary(detecciones)

    porcentaje = (total_area / (img_w * img_h)) * 100 if (img_w * img_h) > 0 else 0

# Generar informe

report_json = save_report(

    detecciones, uploaded_file.name, total_area,

    porcentaje, clase_principal, detector.class_map

)

```

```

# Guardar en historial

st.session_state.historial.append({

    "nombre": uploaded_file.name,

    "clase": detector.class_map.get(clase_principal, "DESCONOCIDO"),

    "area": total_area,

    "porcentaje": porcentaje,

    "timestamp": datetime.now().isoformat()

})


resultados.append({

    "imagen": uploaded_file,

    "result_image": result_image,

    "detecciones": detecciones,

    "clase_principal": clase_principal,

    "nombre_clase": detector.class_map.get(clase_principal, "DESCONOCIDO"),

    "total_area": total_area,

    "porcentaje": porcentaje,

    "report_json": report_json,

    "labels_file": labels_file

})

except Exception as e:

```

```

st.error(f"Error en {uploaded_file.name}: {e}")

# Actualizar barra de progreso

progress_bar.progress((idx + 1) / total)

status_text.text("✅ Procesamiento completado.")

# Mostrar resultados en columnas

for res in resultados:

    col1, col2 = st.columns(2)

    with col1:

        st.image(res["imagen"], caption=f"🖼 Original: {res['imagen'].name}",
use_column_width=True)

    with col2:

        st.image(str(res["result_image"]), caption="🎯 Resultado IA",
use_column_width=True)

# Informe técnico

st.markdown("---")

st.markdown("### 📄 Informe Técnico")

clase = res["clase_principal"]

```

```

nombre_clase = res["nombre_clase"]

if clase == 0:

    st.error(f"🔥 **FUEGO DETECTADO – {nombre_clase}**")

elif clase == 1:

    st.error(f"⚠️ **FUEGO DETECTADO – {nombre_clase}**")

elif clase == 2:

    st.warning(f"🕒 **DETECCIÓN – {nombre_clase}**")

elif clase == 3:

    st.success(f"✅ **{nombre_clase}**")

else:

    st.info(f"🔍 **Detección: {nombre_clase}**")


st.info(f"🇮🇹 Área: **{res['total_area']:.0f} px²** ({res['porcentaje']:.1f}%) | Focos: {len(res['detecciones'])}")

st.markdown(detector.risk_messages.get(clase, "⚠️ Sin clasificación de riesgo."))


# Botones de descarga

col_a, col_b = st.columns(2)

with col_a:

    st.download_button(

        label="📄 Descargar informe (JSON)",

```

```

        data=res["report_json"],

        file_name=f'informe_{res['imagen'].name}.json',

        mime="application/json"

    )

    with col_b:

        # Mostrar etiquetas en detalle

        with st.expander("🔧 Ver etiquetas"):

            if res["labels_file"].exists():

                with open(res["labels_file"], 'r') as f:

                    st.code(f.read())

            # Limpiar archivos temporales después de procesar

            clean_temp_files()

    else:

        st.info("👉 **Arrastra imágenes o haz clic en 'Browse files'**")

    st.markdown('</div>', unsafe_allow_html=True)

    # =====

    # HISTORIAL Y ESTADÍSTICAS

    # =====

    if st.session_state.historial:

        st.markdown('<div class="card">', unsafe_allow_html=True)

```

```

st.markdown("### 🇮🇹 Historial de detecciones")

df_hist = pd.DataFrame(st.session_state.historial)

st.dataframe(df_hist, use_container_width=True)


# Gráfico simple de tipos detectados

st.markdown("#### Distribución de tipos")

tipo_counts = df_hist['clase'].value_counts()

st.bar_chart(tipo_counts)

st.markdown('</div>', unsafe_allow_html=True)


# =====

# CÁMARA (mejorada)

# =====

st.markdown('<div class="card">', unsafe_allow_html=True)

st.markdown("### 📷 **Detección en Tiempo Real**")

st.markdown("Usa la cámara de tu dispositivo para capturar y analizar al instante.")


camera_image = st.camera_input("📷 Tomar foto")

if camera_image:

    # Guardar imagen temporal

    temp_path = UPLOAD_DIR / "camera_capture.jpg"

    with open(temp_path, "wb") as f:

```

```
f.write(camera_image.getbuffer())
```

```
detector = FireDetector()
```

```
with st.spinner("🌀 Procesando..."):
```

```
    try:
```

```
        result_image, labels_file = detector.detect(temp_path)
```

```
        image = Image.open(temp_path)
```

```
        img_w, img_h = image.size
```

```
        detecciones = detector.parse_labels(labels_file, img_w, img_h)
```

```
        clase_principal, total_area, _, _ = detector.get_summary(detecciones)
```

```
        porcentaje = (total_area / (img_w * img_h)) * 100 if (img_w * img_h) > 0 else 0
```

```
    col1, col2 = st.columns(2)
```

```
    with col1:
```

```
        st.image(image, caption="🖼️ Captura original", use_column_width=True)
```

```
    with col2:
```

```
        st.image(str(result_image), caption="🎯 Resultado IA", use_column_width=True)
```

```
    nombre_clase = detector.class_map.get(clase_principal, "DESCONOCIDO")
```

```
    st.markdown(f"**Tipo detectado:** {nombre_clase}")
```

```
    st.markdown(detector.risk_messages.get(clase_principal, ""))
```

```
except Exception as e:
```

```

st.error(f"Error: {e}")

clean_temp_files()

st.markdown('</div>', unsafe_allow_html=True)

# =====

# FOOTER

# =====

st.markdown("""
<div class="footer">

    <p> 🔥 FireScope v4.0 | YOLOv5 + Google Sheets | Procesamiento por lotes | Informes
    JSON</p>

</div>

""", unsafe_allow_html=True)

```