

Trabajo de Grado

SISTEMA IOT DE ALERTA TEMPRANA DE RIESGO DE INCENDIOS
FORESTALES BASADO EN LA NORMA ITU-T Y.4113

Jhon Alejandro Rincón Maldonado

Documento presentado a
Programa de Ingeniería Electrónica
como requisito para optar por el título de
Ingeniero Electrónico

Director: M.Sc. Fabio G. Guerrero

Escuela de Ingeniería Eléctrica y Electrónica (EIEE)
Facultad de Ingeniería



Diciembre 2025

RESUMEN

Este trabajo presenta el diseño e implementación de un sistema IoT de alerta temprana de riesgo de incendios forestales, basado en la recomendación ITU-T Y.4113. La solución integra nodos ESP32, una pasarela de procesamiento en Raspberry Pi y comunicación mediante MQTT. Las mediciones de temperatura, humedad y velocidad del viento se almacenan en InfluxDB y se publican a través de una API REST para el cálculo del índice F como indicador del riesgo.

El sistema se evaluó mediante ensayos controlados que variaron temperatura, humedad y viento. Los resultados muestran sincronización temporal estable, transmisión confiable y una respuesta coherente del índice F frente a los cambios ambientales, lo que confirma la viabilidad operativa de la propuesta.

Palabras clave: IoT, incendios forestales, índice F, ESP32, Raspberry Pi.

ABSTRACT

This work presents the design and implementation of an IoT-based early warning system for forest fire risk, developed according to ITU-T Y.4113. The solution integrates ESP32 sensing nodes, a Raspberry Pi processing gateway, and MQTT-based communication. Measurements of temperature, humidity, and wind speed are stored in InfluxDB and exposed through a REST API for computing the F index as a risk indicator.

The system was evaluated through controlled tests with variations in temperature, humidity, and wind speed. Results show stable synchronization, reliable data transmission, and a consistent response of the F index to environmental changes, demonstrating the feasibility of the proposed system.

Keywords: IoT, forest fires, F index, ESP32, Raspberry Pi.

Glosario

API Interfaz de programación de aplicaciones.

F Index Índice de riesgo de incendios forestales.

FWI Índice meteorológico de incendios.

HMI Interfaz humano-máquina.

HTTP Protocolo de transferencia de hipertexto.

IoT *Internet de las Cosas*.

IT Tecnologías de la información.

ITU-T Unión Internacional de Telecomunicaciones – Sector de normalización.

LAN Red de área local.

MQTT Protocolo de mensajería ligera para telemetría.

NTC Norma técnica colombiana.

RS485 Estándar de comunicación diferencial.

UTC Tiempo universal coordinado.

WAN Red de área amplia.

Wi-Fi Tecnología de comunicación inalámbrica.

Tabla de contenido

	Pag.
Glosario	II
TABLA DE CONTENIDO	III
Lista de figuras	VI
Lista de tablas	VIII
1. Introducción.....	1
1.1 Problema de investigación.....	3
1.2 Objetivo General.....	4
1.3 Objetivos Específicos	4
2. Revisión	6
2.1 <i>Prediction of Peatlands Forest Fires in Malaysia Using Machine Learning</i>	6
2.2 <i>Modelling Long-Term Forest Fire Risk Using Fire Weather Index under Climate Change in Turkey</i>	8
2.3 <i>An Alertness-Adjustable Cloud/Fog IoT Solution for Timely Environmental Monitoring Based on Wildfire Risk Forecasting</i>	10
2.4 <i>Forest Fire Risk Monitoring Using Fuzzy Logic and IoT Technology</i>	10
2.5 Interfaces de usuario.....	11
2.6 Cuadro resumen	12
3. Marco teórico	13
3.1 Índices de riesgo de incendios forestales.....	13
3.1.1 <i>Canadian Forest Fire Danger Rating System</i>	13
3.1.2 <i>National Fire Danger Rating System in US (NFDRS)</i>	14
3.1.3 <i>Forest Fire Danger Index (FFDI)</i>	14
3.1.4 <i>F Index</i>	14
3.2 Microcontroladores y sensores ambientales	18
3.2.1 Microcontroladores	18
3.2.2 Sensores ambientales	19
3.3 Tecnologías de comunicación inalámbrica	20
3.3.1 <i>Wi-Fi</i>	20
3.3.2 <i>LoRa</i>	21

3.3.3	<i>Zigbee</i>	21
3.4	Normas de telecomunicaciones y estándares ITU-T	22
3.4.1	<i>ITU-T Y.2060</i>	22
3.4.2	<i>ITU-T Y.4113</i>	23
3.5	Protocolos de comunicación en <i>IoT</i>	25
3.5.1	<i>Message Queuing Telemetry Transport (MQTT)</i>	26
3.5.2	<i>Constrained Application Protocol (CoAP)</i>	27
3.5.3	<i>Hypertext Transfer Protocol (HTTP)</i>	27
3.6	Servicios de aplicación y visualización de datos	28
3.6.1	<i>Interfaces de programación de aplicaciones (API)</i>	28
3.6.2	Aplicaciones web y móviles	29
4.	Metodología	30
4.1	Capa de dispositivos	31
4.1.1	Arquitectura física de los nodos	32
4.1.2	ESPHome y archivos YAML: concepto, implementación y control de calidad de datos	35
4.1.3	Diseño del circuito impreso (PCB)	38
4.2	Capa de procesamiento	40
4.2.1	Arquitectura de procesos en la Raspberry Pi	40
4.2.2	Adquisición de datos del anemómetro	41
4.2.3	Sincronización temporal, ingesta de datos y almacenamiento en InfluxDB	42
4.2.4	Cálculo del índice F	43
4.2.5	Sistema de alerta por SMS	44
4.2.6	Diagrama de flujo de datos	45
4.2.7	Visualización local con Grafana	45
4.2.8	Resumen de servicios y ejecución como <i>daemons</i>	46
4.3	Capa de red	47
4.3.1	Infraestructura y topología de la red local	47
4.3.2	Conectividad externa y política de exposición	49
4.3.3	Seguridad y operación del punto de acceso	49
4.4	Capa de aplicación	50
4.4.1	Convenciones transversales de la API	51
4.4.2	Endpoints expuestos y su semántica	52
4.4.3	Opciones de visualización y análisis	52
4.4.4	Notificaciones locales en <i>Android</i> por umbral	56
4.4.5	Repositorio del proyecto	57
5.	Resultados y discusión	58
5.1	Escenarios experimentales	58
5.2	Resultados experimentales	59

5.3 Resumen del comportamiento del índice F	68
6. Conclusiones	69
7. Recomendación para trabajos futuros	71
Referencias	73
Apendice. A. Figuras	77

ÍNDICE DE FIGURAS

FIGURA	Pag.
2.1 Arquitectura del sistema IoT utilizado en Prediction of Peatlands Forest Fires in Malaysia Using Machine Learning	7
2.2 Diagrama de secuencia utilizado en Prediction of Peatlands Forest Fires in Malaysia Using Machine Learning	8
2.3 Valores actuales y futuros del F Index sobre el mapa de Turquía	9
2.4 Arquitectura general del sistema en Forest Fire Risk Monitoring Using Fuzzy Logic and IoT Technology	11
3.1 Rangos de peligro del Forest Fire Danger Index (FFDI)	15
3.2 Modelo de referencia del Internet de las Cosas según la norma ITU-T Y.2060	24
3.3 Modelo básico de red para el Internet de las Cosas propuesto en la norma ITU-T Y.4113	25
4.1 Arquitectura general del sistema de alerta temprana de incendios forestales	31
4.2 Esquemático de conexión del nodo ESP32 con el sensor DHT22	33
4.3 Montaje físico del nodo sobre PCB	33
4.4 Conexión del anemómetro RS485 (DFRobot SEN0483) a la Raspberry Pi mediante conversor RS485-USB y fuente de 12 VDC	34
4.5 Flujo de datos e interacción entre servicios: sincronización temporal, adquisición, cálculo de índice F y alerta por SMS	45
4.6 Interfaz de monitoreo local en Grafana	46
4.7 Página web: vista de la sección <i>Summary</i>	54
4.8 Página web: vista de la sección <i>Details</i>	54
4.9 APP móvil: vista de la sección <i>Summary</i>	55

4.10	APP móvil: vista de la sección <i>Details</i>	55
4.11	Aplicación móvil: sección de notificaciones por umbral	57
5.1	Montaje del escenario favorable	60
5.2	Registro en Grafana (escenario favorable)	61
5.3	Registro web del índice F (favorable)	62
5.4	Registro en la app móvil (favorable)	62
5.5	Montaje del escenario desfavorable	64
5.6	Registro en Grafana (escenario desfavorable)	65
5.7	Registro web del índice F (desfavorable)	65
5.8	Registro en la app móvil (desfavorable)	66
5.9	SMS de la primera etapa (desfavorable)	67
5.10	SMS de la segunda etapa (desfavorable)	67
A.1	Arquitectura <i>IoT</i> propuesta en An Alertness-Adjustable Cloud/Fog IoT Solution for Timely Environmental Monitoring Based on Wildfire Risk Forecasting.	77
A.2	Esquema general de interfaces de usuario para proyectos de estas características.	78
A.3	Diseño de PCB, vista superior (capa F.Cu).....	79
A.4	Diseño de PCB, vista inferior (capa B.Cu).....	80

ÍNDICE DE CUADROS

TABLA	Pag.
2.1 Cuadro resumen de antecedentes	12
3.1 Escala de peligro del índice F.	17
4.1 Conexiones principales del nodo sensor basado en ESP32.	35
4.2 Conexiones principales del nodo anemómetro RS485.....	35
4.3 Servicios ejecutados en la Raspberry Pi y su función principal.....	46
4.4 Parámetros efectivos de la capa de red	48
5.1 Escenarios experimentales y condiciones ambientales	59
5.2 Relación entre las condiciones ambientales y el índice F	68

1. Introducción

Las áreas boscosas representan aproximadamente el 31 % de la superficie del planeta y juegan un papel fundamental no solo en el equilibrio ambiental y ecológico sino también en el bienestar humano. Los bosques no solo actúan como reguladores del clima al absorber dióxido de carbono y liberar oxígeno, indispensable para toda la vida en el planeta, sino que también ofrecen hábitats y sustento para las numerosas especies animales y vegetales que los habitan. Además, los bosques protegen el suelo contra la erosión y brindan recursos naturales valiosos para la sociedad humana como lo son la madera, los alimentos y las plantas medicinales, relevantes a nivel cultural, ambiental e industrial de manera global [1].

En América Latina, una región conocida por sus extensos y diversos bosques como la selva amazónica, los incendios forestales han aumentado significativamente en los últimos años. Silveira et al. analizan en su artículo la actividad de incendios en todos los países de la región amazónica desde 2003 hasta 2020 y encontraron que, para 2020, el área total quemada en la Amazonía fue la mayor desde el año 2010 debido a la intensificación de los incendios agrícolas y la deforestación de los ecosistemas amazónicos [2]. Por esta razón, los incendios forestales suponen la mayor amenaza para estos valiosos ecosistemas.

En Colombia, la situación es igualmente preocupante. Del 3 de noviembre del 2023 hasta el 20 de enero del 2024, en el marco del fenómeno El Niño, se registraron 237 incendios forestales que afectaron 131 municipios y 3,523 hectáreas de vegetación [3]. Esto representa un aumento en la frecuencia y la intensidad de los incendios y se debe a una combinación de factores, incluyendo el cambio climático, la deforestación y las prácticas agrícolas insostenibles [4]. Los incendios no solo destruyen valiosos ecosistemas, sino que también tienen un impacto negativo en la salud pública [5] y en la economía del país [6], lo que pone en evidencia la urgente necesidad de implementar estrategias más efectivas para la prevención y atención de estos eventos.

En los últimos años, Colombia ha enfrentado grandes desafíos en la gestión de incendios forestales. Por ejemplo, el uso de técnicas convencionales de control del

fuego, la falta de inversión en desarrollo tecnológico y diversas dificultades administrativas han limitado la capacidad de respuesta del país a estos siniestros [7]. Debido a esto, en diversas poblaciones del país aún no se cuenta con equipos tecnológicos para la prevención o atención de incendios; siguen dependiendo de métodos convencionales y limitados, lo que retrasa la respuesta al evento y aumenta el impacto negativo de los incendios forestales.

En este contexto, surge la necesidad de explorar e implementar una solución de ingeniería accesible a las comunidades más remotas del país, que permita cuantificar el riesgo de incendio en las zonas boscosas de Colombia. De este modo será posible generar alertas tempranas a las autoridades competentes o la comunidad sin necesidad de contar con equipos o formación especializada. Para esto, resulta esencial comprender los factores climáticos y ambientales que determinan el riesgo de incendios forestales. La cuantificación de este riesgo permite transformar variables meteorológicas en información útil para la toma de decisiones y la prevención de desastres.

La medición del riesgo de incendios forestales otorga un valor cuantitativo al posible peligro que representa un incendio forestal dadas las condiciones ambientales actuales. Esta medición se conoce como índice de riesgo de incendio (en inglés, *Fire Weather Index, FWI*) y es una poderosa herramienta para prevenir y mitigar los daños provocados por estos, ya que proporciona un valor que puede ser fácilmente interpretado por cualquier persona. Normalmente esta medición se evalúa a través de un índice integrado que incorpora variables meteorológicas como la temperatura del aire, la humedad relativa, la velocidad y dirección del viento, y las precipitaciones [8], factores que en conjunto determinan las condiciones climáticas propicias para el inicio y propagación de incendios forestales. Estas variables climáticas son fundamentales e intuitivas ya que, por ejemplo, altas temperaturas y baja humedad aumentan la probabilidad de incendios, mientras que la velocidad del viento influye significativamente en la velocidad de propagación del fuego [9].

En el mundo existen diversos sistemas de clasificación del peligro de incendios forestales que fueron desarrollados por las agencias ambientales de algunos países como Canadá, Estados Unidos o Australia. Estos índices se basan en modelos matemáticos empíricos adaptados a las condiciones ambientales del país donde se

desarrollaron. A continuación, se listan los más relevantes: *Canadian Forest Fire Danger Rating System* (conocido como *CFFDRS*, cuyo componente meteorológico es el *Fire Weather Index, FWI*) [10], *National Fire Danger Rating System (NFDRS)* [11] y *Forest Fire Danger Index (FFDI)* [12]. En la actualidad, la mayoría de los países usan versiones adaptadas de uno de estos sistemas usando las condiciones propias de sus ecosistemas, geografía y clima.

Estos sistemas suelen ser complejos y requieren de información histórica de las variables ambientales del lugar donde se aplica. Por esta razón, el cálculo de estos índices suele estar solo al alcance de expertos en el tema o de la autoridad ambiental de la nación. Para abordar este problema, Sharples et al. [13] proponen en su artículo la aplicación de un índice de riesgo simplificado que tiene un comportamiento similar al de los índices más usados actualmente. Este indicador, conocido como *F Index*, está diseñado para ser calculado a través de una sencilla ecuación usando solo tres variables clave: temperatura ambiente, humedad relativa y velocidad del viento. Estas variables son medibles en cualquier lugar con sensores y microcontroladores comerciales como Arduino o Raspberry Pi [14, 15], lo que lo hace más accesible y fácil de implementar en áreas con recursos tecnológicos limitados.

1.1 Problema de investigación

A pesar de los avances teóricos en la evaluación del riesgo de incendios forestales, en Colombia persiste una brecha significativa entre el conocimiento científico disponible y la aplicación práctica de herramientas de monitoreo accesibles para las comunidades rurales. Las limitaciones en infraestructura tecnológica, conectividad y recursos económicos dificultan la implementación de sistemas predictivos convencionales. Por ello, resulta necesario desarrollar una solución que, apoyándose en el *F Index* propuesto por Sharples et al. [13], permita medir las condiciones ambientales mediante sensores de bajo costo y generar alertas tempranas en tiempo real.

En este sentido, el presente proyecto busca formular un sistema *IoT* (Internet de las cosas) que integre sensores ambientales, procesamiento local de datos y transmisión inalámbrica de la información, de manera que se facilite la detección

oportuna de condiciones que incrementen el riesgo de incendio. Este enfoque pretende reducir la vulnerabilidad de las zonas boscosas del país y fortalecer la capacidad de respuesta ante emergencias ambientales.

En consecuencia, se plantea la siguiente pregunta problematizadora:

- ¿Cómo calcular un índice de riesgo de incendio simplificado en las zonas boscosas de Colombia, usando tecnologías de bajo costo y de software libre, que permitan cuantificar este riesgo y generar una alerta temprana para las comunidades o la autoridad competente?

1.2 Objetivo General

- Desarrollar un sistema *IoT* de alerta temprana de riesgo de incendios forestales basado en la norma ITU-T Y.4113.

1.3 Objetivos Específicos

- Implementar un sistema de comunicaciones inalámbrico con sensores *IoT* basado en la arquitectura definida en la norma ITU-T Y.4113.
- Implementar una aplicación móvil y una web para el análisis de la información meteorológica del sistema, visualización del *F Index* y generación de alertas tempranas.
- Evaluar el funcionamiento del sistema empleando condiciones controladas de ajuste de las variables medidas.

Este trabajo de grado está compuesto por seis capítulos. En el capítulo 1 se presenta el planteamiento del problema de investigación, los objetivos y el alcance del trabajo, además de una breve introducción a los temas más relevantes que sustentan el proyecto. El capítulo 2 describe los antecedentes y los trabajos más significativos

considerados en la revisión del estado del arte. El capítulo 3 desarrolla el marco teórico, profundizando en los conceptos, modelos y fundamentos técnicos relacionados con el estudio. En el capítulo 4 se detalla la solución propuesta, incluyendo su diseño, componentes y arquitectura de implementación. El capítulo 5 presenta los resultados de las pruebas y validaciones realizadas al sistema. Finalmente, el capítulo 6 expone las conclusiones generales del trabajo y las recomendaciones para desarrollos futuros.

2. Revisión

La medición y monitoreo del riesgo de incendios forestales es un tema que ha sido ampliamente estudiado alrededor del mundo, ya que los incendios forestales representan una de las principales amenazas a los ecosistemas boscosos de cualquier lugar del planeta. A lo largo del tiempo, se han desarrollado diversas metodologías y tecnologías para medir y predecir este riesgo, que van desde el cálculo de índices hasta el uso de *Machine Learning* para predecir incendios. A continuación, se presentan cuatro estudios representativos que abordan enfoques diferentes para la medición del riesgo de incendios forestales, cada uno con su propia metodología y contexto ambiental, y que proporcionan una base teórica y técnica importante para el desarrollo del presente proyecto.

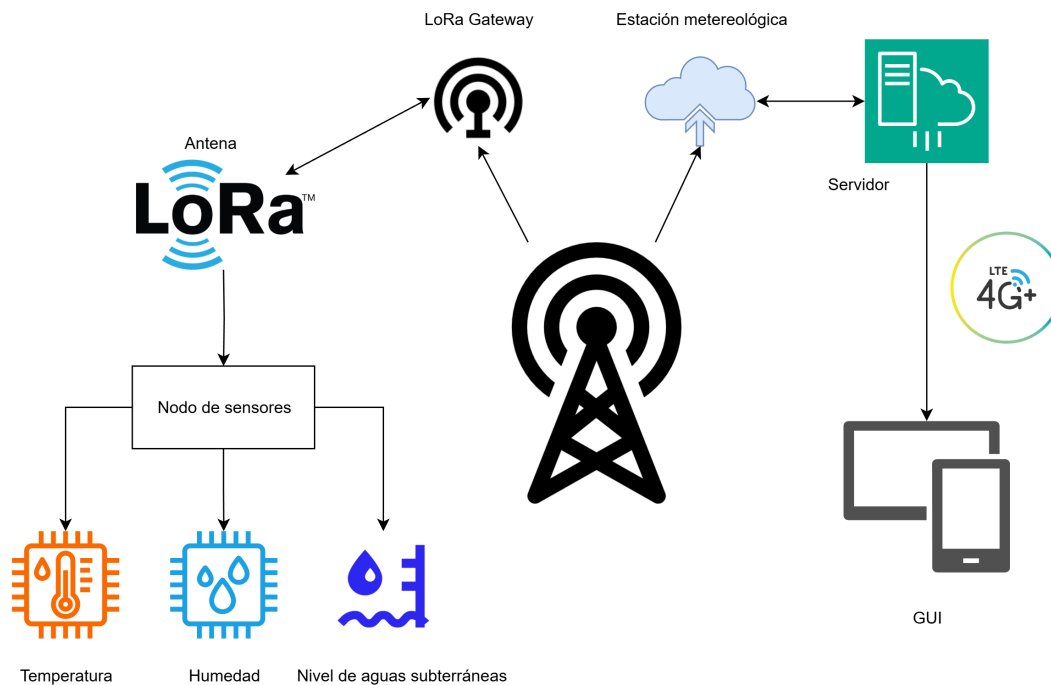
2.1 *Prediction of Peatlands Forest Fires in Malaysia Using Machine Learning*

El estudio realizado por Li et al. [16] explora el uso de algoritmos de *Machine Learning* para la predicción de incendios en áreas de turberas en Malasia. Las turberas son un tipo de humedal donde se produce la acumulación superficial por capas de material orgánico en un estado de descomposición conocido como turba. Debido a esto, es un ecosistema especialmente vulnerable a los incendios forestales.

Este trabajo se basa en el análisis de datos históricos de incendios forestales y condiciones meteorológicas al momento de incendios en las turberas, combinándolos con modelos predictivos que permiten identificar zonas de alto riesgo con anticipación. Los autores emplean datos de sensores y estaciones meteorológicas para entrenar modelos de *Machine Learning*, que luego son capaces de hacer predicciones sobre la ocurrencia de incendios a corto plazo. Como se muestra en la Figura 2.1, estas estaciones meteorológicas usan sensores *IoT* para medir la temperatura, humedad y el nivel del agua del subsuelo; después usan *LoRa (Long Range)* como tecnología de comunicación inalámbrica para enviar la información a servicios en la nube donde esta se procesa y finalmente se visualiza la información recolectada a través de las

aplicaciones móvil y web del proyecto.

Figura 2.1. Arquitectura del sistema *IoT* utilizado en *Prediction of Peatlands Forest Fires in Malaysia Using Machine Learning* [16].

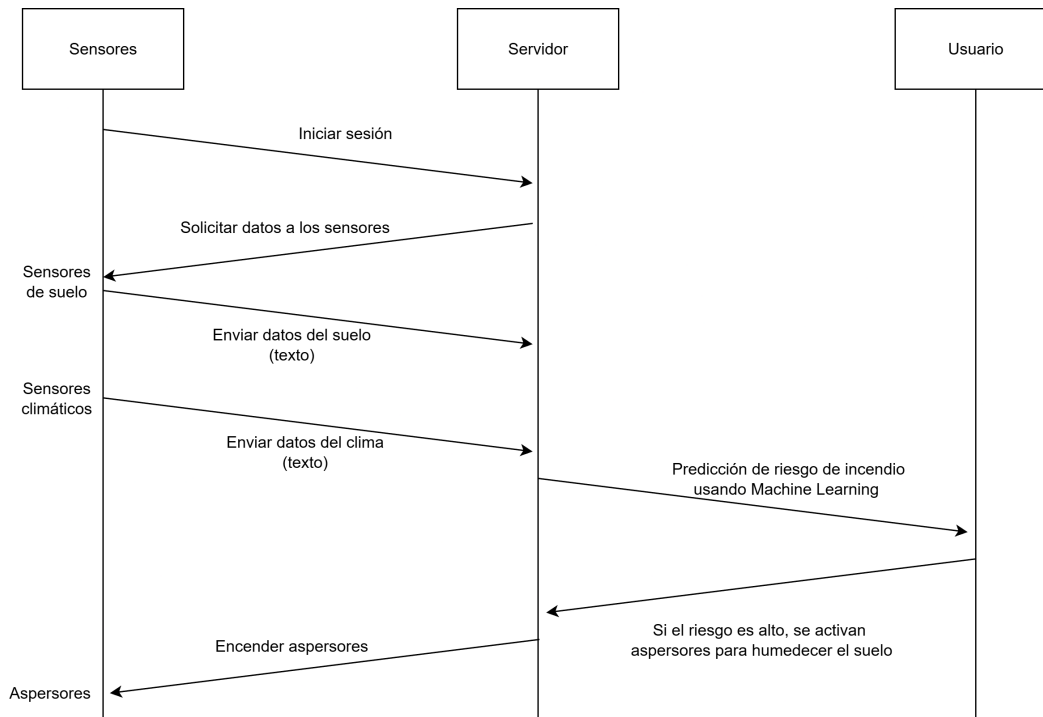


Fuente: Elaboración propia.

Aunque este enfoque de *Machine Learning* ofrece resultados impresionantes en términos de precisión y capacidad predictiva, su aplicación requiere grandes volúmenes de datos históricos y capacidad computacional avanzada, lo que lo hace restrictivo en áreas donde los recursos tecnológicos y de infraestructura son limitados.

El sistema planteado en este anteproyecto se distancia del uso de *Machine Learning* en favor de un enfoque más simple y directo, pero este estudio proporciona un marco de referencia sobre la importancia de los datos históricos y las variables ambientales en la predicción del riesgo de incendios. Si bien no se pretende predecir incendios mediante IA, el manejo y monitoreo de las variables ambientales en tiempo real usado en *Prediction of Peatlands Forest Fires in Malaysia Using Machine Learning* sigue una estructura similar a la planteada en este proyecto, como se puede apreciar en la Figura 2.2.

Figura 2.2. Diagrama de secuencia utilizado en *Prediction of Peatlands Forest Fires in Malaysia Using Machine Learning* [16].



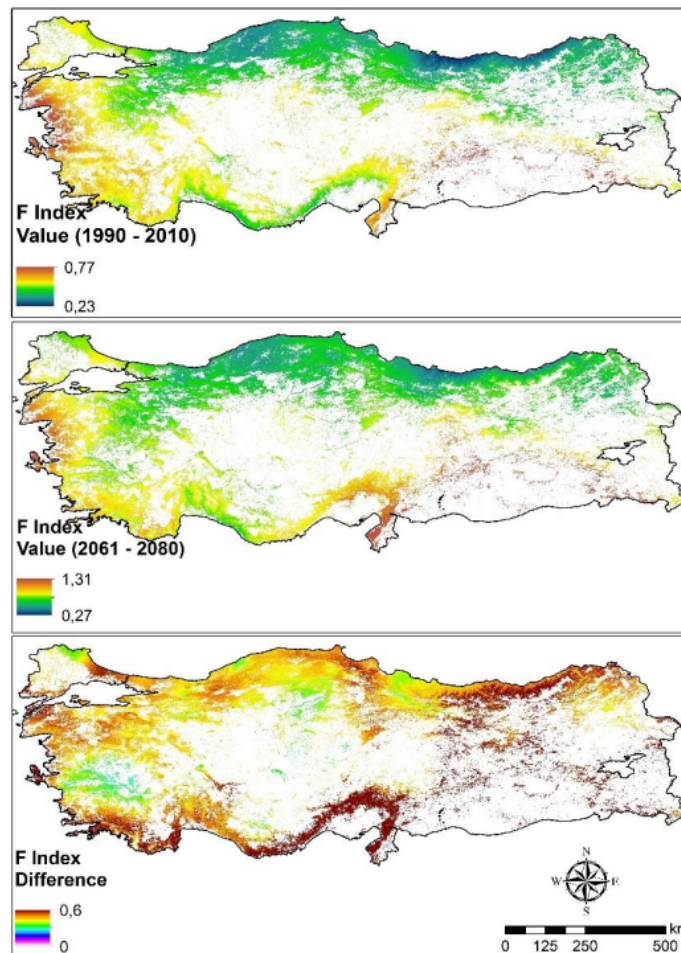
Fuente: Elaboración propia.

2.2 *Modelling Long-Term Forest Fire Risk Using Fire Weather Index under Climate Change in Turkey*

En este estudio centrado en Turquía, Satir et al. [17] utilizan el *F Index* propuesto por Sharples et al. [13] para modelar el riesgo de incendios forestales en condiciones actuales y futuras bajo escenarios de cambio climático. Como ya se mencionó, el *F Index* simplifica el cálculo del índice al depender solamente de tres variables climáticas: temperatura ambiente, humedad relativa y velocidad del viento. Este modelo tiene una ventaja significativa al requerir menos datos y ser más fácil de implementar en regiones con limitaciones tecnológicas o con menor acceso a grandes volúmenes de datos históricos. Los resultados de este estudio son útiles para gestionar incendios forestales y planificar estrategias a largo plazo en un contexto de cambio climático. Se destaca también la posibilidad de usar estos resultados para integrar tecnologías de monitoreo en tiempo real, como drones y sensores remotos.

Este artículo obtuvo como conclusión que el *F Index* es una técnica eficaz de evaluación del riesgo de incendios forestales. Como se muestra en la Figura 2.3, también determinó que la tasa de riesgo de incendios forestales en Turquía aumentará alrededor de un 21 % en la década de 2070 según el escenario planteado en el estudio. Esto pone en evidencia que en el futuro, gracias al cambio climático, serán necesarias nuevas estrategias para reducir los efectos de los incendios forestales.

Figura 2.3. Valores actuales y futuros del *F Index* sobre el mapa de Turquía [17].



Fuente: Tomado de *Modelling Long-Term Forest Fire Risk Using Fire Weather Index under Climate Change in Turkey* [17].

2.3 An Alertness-Adjustable Cloud/Fog IoT Solution for Timely Environmental Monitoring Based on Wildfire Risk Forecasting

Otro ejemplo relevante es el trabajo presentado por Tsipis et al. [18], en el cual se propone una red de sensores *IoT* que integra una arquitectura híbrida de *Cloud/Fog computing* (computación en la nube y en la niebla). Este sistema se validó intensivamente en las instalaciones de la Universidad Jónica en Grecia y demostró su eficiencia para ajustar de manera autónoma su comportamiento de sensado en función de las condiciones ambientales críticas, con especial enfoque en el riesgo de incendios. Un aspecto clave de este sistema es la capacidad de reducir significativamente la latencia, manteniendo un alto nivel de precisión y control del consumo energético. Adicionalmente, ofrece una interfaz web que permite la visualización en tiempo real y la predicción del riesgo de incendio, basándose en el *Chandler's Burning Index* (índice de quema de Chandler), propuesto por Chandler et al. [19]. Este índice también se trata de un modelo simplificado como el *F Index*, con la diferencia de que solo requiere de la temperatura ambiente y la humedad relativa para su cálculo.

Este enfoque destaca la relevancia de integrar tecnologías como la computación en la niebla, que podrían ser aplicables para mejorar la eficiencia y capacidad de respuesta en sistemas de monitoreo de incendios como el que se propone en este proyecto. La arquitectura de este proyecto se muestra en la Figura A.1 y sirve como base para la arquitectura que se plantea en este anteproyecto.

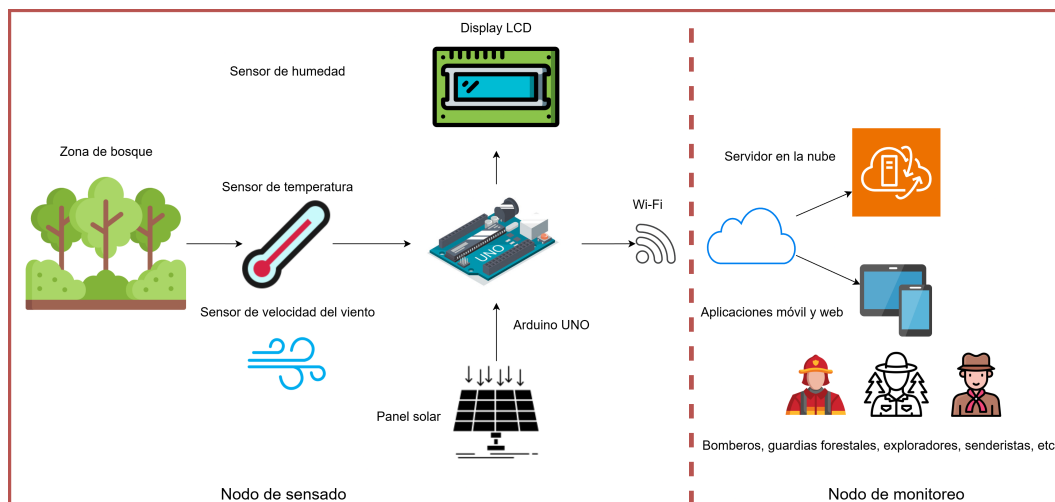
2.4 Forest Fire Risk Monitoring Using Fuzzy Logic and IoT Technology

Por último, en el artículo de Sahour et al. [20] se introduce un enfoque innovador al combinar la tecnología *IoT* con el cálculo del *F Index*. En este estudio se desarrolló un sistema de monitoreo en tiempo real para calcular el riesgo de incendio forestal en regiones vulnerables. Como se observa en el esquema de la Figura 2.4, los sensores *IoT* están junto al microcontrolador, conformando una estación meteorológica independiente y portátil gracias a que funciona con paneles solares. En esta estación se miden las variables necesarias para el cálculo del *F Index*: temperatura ambiente, humedad relativa y velocidad del viento, que son enviadas a una plataforma de servicios en la nube que utiliza lógica difusa para interpretar los datos que finalmente

se muestran al usuario a través de las aplicaciones móvil y web.

La ventaja de este enfoque es que permite un monitoreo continuo y en tiempo real, lo que mejora significativamente la capacidad de respuesta ante un potencial incendio. Además, la lógica difusa permite manejar la incertidumbre inherente a las mediciones, ofreciendo una herramienta precisa y adaptable a distintos ecosistemas forestales. Como punto a mejorar, la estación meteorológica tiene datos limitados del entorno al estar todos los sensores concentrados en un solo lugar; por lo tanto, el índice que se calcula solo es válido en las cercanías de la estación.

Figura 2.4. Arquitectura general del sistema en *Forest Fire Risk Monitoring Using Fuzzy Logic and IoT Technology* [20].



Fuente: Elaboración propia.

2.5 Interfaces de usuario

Cada uno de los artículos mencionados anteriormente cuenta con su propia *GUI* (interfaz gráfica de usuario), la cual busca organizar y mostrar eficazmente la información a los usuarios. En la Figura A.2 se presenta un esquema general que resume los elementos principales que suelen tener las interfaces de usuario para proyectos de estas características.

2.6 Cuadro resumen

Cuadro 2.1. Cuadro resumen de antecedentes

	<i>Prediction of Peatlands Forest Fires in Malaysia Using Machine Learning</i>	<i>Modelling Long-Term Forest Fire Risk Using Fire Weather Index under Climate Change in Turkey</i>	<i>An Alertness-Adjustable Cloud/Fog IoT Solution for Timely Environmental Monitoring Based on Wildfire Risk Forecasting</i>	<i>Forest Fire Risk Monitoring Using Fuzzy Logic and IoT Technology</i>
Objetivo	Predecir incendios en la zona de turberas de Malasia.	Evaluar el riesgo de incendios forestales a largo plazo en Turquía.	Monitoreo en tiempo real del riesgo de incendios, probado en Grecia.	Monitoreo del riesgo de incendios forestales en cualquier lugar.
Metodología	Uso de algoritmos de <i>Machine Learning</i> , basados en datos históricos de la zona estudiada.	Cálculo del <i>F Index</i> de Sharples usando información histórica de Turquía y teniendo en cuenta el impacto del cambio climático en el riesgo de incendios forestales.	Cálculo del índice de <i>Chandler</i> usando una red de sensores <i>IoT</i> , procesado de datos con una arquitectura híbrida de computación en la nube y la niebla, así como el uso de aplicaciones móvil y web como <i>GUI</i> .	Cálculo del <i>F Index</i> de Sharples usando una red de sensores <i>IoT</i> , procesado de datos en la nube y aplicaciones móvil y web como <i>GUI</i> .
Variables Clave	Datos históricos de temperatura, humedad, precipitaciones y la cantidad de materia orgánica contenida en el suelo.	Temperatura ambiente, humedad relativa, velocidad del viento y datos históricos relacionados al cambio climático.	Temperatura ambiente y humedad relativa.	Temperatura ambiente, humedad relativa y velocidad del viento.
Ventajas	Capacidad de predecir incendios basándose en patrones históricos complejos.	Adaptación del <i>F Index</i> a los efectos del cambio climático específico de la región.	La computación en la niebla mejora la eficiencia y reduce la latencia en el procesamiento de datos.	Permite el cálculo del <i>F Index</i> en cualquier lugar y el monitoreo en tiempo real de las variables más relevantes.
Limitaciones	Dependencia de la calidad y cantidad de datos históricos para entrenar el modelo de <i>Machine Learning</i> .	Dependencia de datos climáticos históricos específicos para el modelado del efecto del cambio climático.	Requiere infraestructura más compleja y costosa para implementar computación en la niebla y nube.	El diseño de estación meteorológica centralizada limita el área efectiva que se puede monitorear.

Fuente: Elaboración propia.

3. Marco teórico

En el siguiente capítulo se presentan los conceptos y tecnologías fundamentales que sustentan el desarrollo del proyecto, con el propósito de establecer un marco conceptual sólido para su comprensión e implementación. Se abordan los principales índices de riesgo de incendios forestales, los sensores ambientales y microcontroladores utilizados en la adquisición de datos, las tecnologías de comunicación inalámbrica empleadas para la transmisión de la información, las normas y estándares de telecomunicaciones propuestos por la *Unión Internacional de Telecomunicaciones* (ITU-T), así como los protocolos de comunicación y los servicios de aplicación propios de los sistemas *IoT*.

3.1 Índices de riesgo de incendios forestales

Los índices de riesgo de incendios forestales son modelos empíricos que permiten cuantificar la probabilidad de que se produzca un incendio en función de las condiciones meteorológicas y ambientales. Estos índices integran múltiples variables que afectan el comportamiento del fuego, como la temperatura ambiente, la humedad relativa, la velocidad del viento y las precipitaciones [8]. Algunos de los índices más conocidos y utilizados a nivel mundial son:

3.1.1 *Canadian Forest Fire Danger Rating System*

El *Canadian Forest Fire Danger Rating System (CFFDRS)* es uno de los sistemas más detallados y precisos para evaluar el riesgo de incendios. Este sistema se desarrolló para medir el riesgo de incendios forestales en Canadá. Se basa en cuatro componentes principales: el *Fire Weather Index (FWI)*, que combina la temperatura ambiente, la humedad relativa, la velocidad del viento y las precipitaciones para determinar la gravedad del riesgo de incendio. Además, el *Drought Code (DC)* evalúa la sequedad del suelo y su capacidad para sostener el fuego, mientras que el *Fine Fuel Moisture Code (FFMC)* mide la humedad de los combustibles finos sobre el suelo, como hojas, hierba o material orgánico. Por último, el *Buildup Index (BUI)* estima la

cantidad de combustible disponible que puede consumir un incendio [10].

3.1.2 *National Fire Danger Rating System in US (NFDRS)*

Este sistema se desarrolló para medir el riesgo de incendios forestales en los Estados Unidos. Este índice está diseñado para evaluar el peligro de incendios en función de varios factores meteorológicos y del estado del combustible disponible. Incluye tres componentes: el *Energy Release Component (ERC)*, que mide la cantidad de energía liberada si el combustible se quema completamente, lo que proporciona un indicador sobre la intensidad del incendio; el *Burning Index (BI)*, que combina información sobre la velocidad del viento, la humedad relativa y el tipo de combustible disponible para evaluar la dificultad de control del fuego; y finalmente, el *Spread Component (SC)*, que evalúa la tasa a la cual un incendio podría propagarse, basado en las condiciones meteorológicas de ese momento [11].

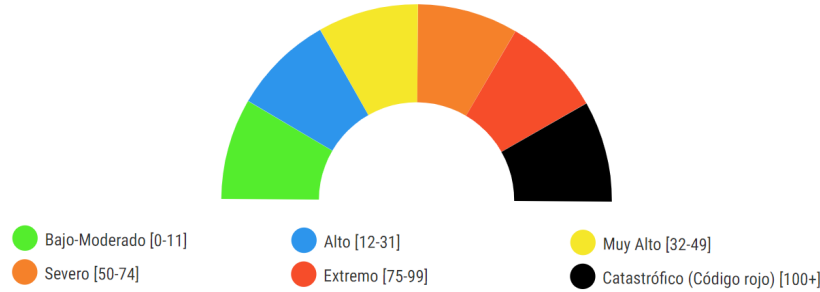
3.1.3 *Forest Fire Danger Index (FFDI)*

Este sistema se desarrolló para medir el riesgo de incendios forestales en Australia. Este índice está diseñado para proporcionar una medida rápida y efectiva de las condiciones que favorecen la ignición y propagación de incendios. Su cálculo se basa en una combinación de variables meteorológicas que influyen directamente en el comportamiento del fuego. El *FFDI* se calcula utilizando cuatro parámetros principales: temperatura ambiente, humedad relativa, velocidad del viento y cantidad de precipitaciones [12]. Este índice utiliza una fórmula que combina estos factores y cuyo resultado es un número que se interpreta por rangos como se muestra en la Figura 3.1.

3.1.4 *F Index*

El *F Index* (índice F), propuesto por Sharples et al. [13], es un índice simplificado diseñado para evaluar el riesgo de incendios forestales mediante una combinación de tres variables meteorológicas: temperatura ambiente, humedad relativa y velocidad del viento. Este índice es especialmente valioso en contextos donde la medición de

Figura 3.1. Rangos de peligro del *Forest Fire Danger Index (FFDI)* [12].



Fuente: Elaboración propia.

variables ambientales es limitada y se requieren evaluaciones rápidas y efectivas del riesgo de incendio. A continuación, se muestra una breve explicación de la importancia y relevancia de estas variables ambientales:

- **Temperatura:** A medida que la temperatura aumenta, también lo hace la tasa de evaporación y la deshidratación de los combustibles, lo que aumenta la probabilidad de que estos den inicio a un fuego. Las temperaturas más altas generalmente se asocian con un mayor riesgo de incendios.
- **Humedad Relativa:** Este factor mide la cantidad de vapor de agua presente en el aire. Una baja humedad relativa significa que los combustibles vegetales se secan más rápidamente, aumentando su susceptibilidad a incendios. Valores de humedad por debajo del 30 % suelen considerarse críticos para el riesgo de incendio.
- **Velocidad del Viento:** La velocidad y dirección del viento influyen en la propagación del fuego. Vientos fuertes no solo ayudan a propagar el fuego más rápidamente, sino que también pueden llevar brasas a áreas no quemadas, facilitando la creación de nuevos focos de incendio.

Sharples et al. introdujeron un índice adimensional de humedad del combustible (*Fuel Moisture Index, FMI*) como se muestra en la ecuación (4.2), que es comparado con varios modelos existentes para determinar el contenido de humedad de combustibles finos y muertos. Los resultados demostraron que el *FMI* propuesto por Sharples et al.

proporciona una medida equivalente a la producida por otros modelos reconocidos. El *FMI* viene dado por la siguiente expresión, que representa la idea intuitiva de que, bajo condiciones cálidas y secas, el contenido de humedad del combustible será más bajo. En la ecuación (4.2), T representa la temperatura ambiente expresada en grados Celsius ($^{\circ}\text{C}$) y H la humedad relativa expresada como un porcentaje.

$$FMI = 10 - 0.25(T - H) \quad (3.1)$$

Sharples et al. plantean que un índice de riesgo de incendio es una combinación de información sobre la velocidad del viento y el contenido de humedad del combustible, donde este último se obtiene a partir de la temperatura ambiente y la humedad relativa (*FMI*). Intuitivamente, el peligro de incendio disminuye a medida que aumenta el contenido de humedad del combustible, pero aumenta a medida que crece la velocidad del viento. Con base en esta idea, Sharples et al. sugieren un índice de riesgo de incendio simple calculado mediante la siguiente expresión:

$$F = \frac{\max(U_0, U)}{FMI} \quad (3.2)$$

En la ecuación (3.2), U indica la velocidad del viento expresada en km/h y U_0 es un umbral introducido para garantizar que el índice de peligro de incendio sea superior a cero incluso para una velocidad del viento nula. Sharples et al. indican que en un análisis auxiliar que no se reportó en el artículo, se encontró que con $U_0 = 1$ km/h se obtienen resultados óptimos. De esta manera, se constituye la fórmula para hallar el índice F . Este índice obtiene como resultado un número adimensional, que crece a medida que aumentan las condiciones que propician los incendios y disminuye en el caso contrario. Además, el artículo también establece una escala de peligro dividida en cinco niveles, como se muestra en el Cuadro 3.1.

Cuadro 3.1. Escala de peligro del índice F.

Índice F	Riesgo de incendio
[0.0 – 0.7]	Bajo
[0.7 – 1.5]	Moderado
[1.5 – 2.7]	Alto
[2.7 – 6.1]	Muy alto
>6.1	Extremo

Fuente: Elaboración propia con base en [13].

Finalmente, Sharples et al. comparan en su artículo el comportamiento del índice F con dos variaciones del *Forest Fire Danger Index*, una conocida como *McArthur Mark 4 Grassland Fire Danger Index (GFDI)* y otra conocida como *McArthur Mark 5 Forest Fire Danger Index (FFDI)*. Además, comparan el resultado con otro índice denominado *Fosberg Fire Weather Index (FFWI)* y demuestran que no hay variaciones significativas entre el comportamiento de estos indicadores ante las mismas entradas de variables ambientales, comprobando así la confiabilidad del índice [13].

Se adopta el índice F en este proyecto por su formulación sencilla y su dependencia directa de tres variables meteorológicas fácilmente medibles: temperatura, humedad relativa y velocidad del viento. Esta simplicidad reduce el costo computacional en microcontroladores, permite evaluaciones rápidas en tiempo real y conserva una correlación coherente con otros índices de referencia, como el *Forest Fire Danger Index (FFDI)* y el *Fosberg Fire Weather Index (FFWI)*. De esta manera, el índice F constituye una alternativa práctica y precisa para sistemas de alerta temprana implementados con recursos limitados.

3.2 Microcontroladores y sensores ambientales

Los microcontroladores y los sensores ambientales son el componente *hardware* del proyecto y son fundamentales en la implementación de sistemas de monitoreo ambiental. Los microcontroladores son dispositivos electrónicos que permiten la automatización y el control de diferentes procesos mediante programación. Además, son esenciales en aplicaciones *IoT*, ya que facilitan la recolección de datos de sensores y la comunicación con otras partes del sistema.

3.2.1 Microcontroladores

Entre los microcontroladores más utilizados se encuentran *Arduino*, *Raspberry Pi* y *ESP32*. *Arduino* es una plataforma de *hardware* de código abierto que se basa en una serie de placas de circuito impreso (*PCBs*) con microcontroladores, permitiendo la programación sencilla a través de su entorno de desarrollo. Es popular por su facilidad de uso y bajo costo, lo que lo hace accesible para proyectos educativos y de investigación [14, 21].

Por otro lado, *Raspberry Pi* es una computadora de placa única que ofrece capacidades más avanzadas, incluyendo procesamiento de datos más rápido y opciones de conectividad más amplias, lo que la convierte en una excelente opción para proyectos que requieren una mayor capacidad de procesamiento [15].

Finalmente, *ESP32* es otro microcontrolador destacado en el ámbito del *IoT*. Se trata de una solución de bajo costo y alto rendimiento que combina un microcontrolador de 32 bits con conectividad *Wi-Fi* y *Bluetooth* integradas. Esta característica lo convierte en una opción interesante para aplicaciones que requieren comunicación inalámbrica, como la presentada en este proyecto [22].

Se eligió *ESP32* en la capa de dispositivos por su excelente relación costo-prestaciones, conectividad *Wi-Fi* y *Bluetooth* integradas, amplia comunidad y compatibilidad con entornos de desarrollo ligeros (*MicroPython/ESPHome*). La *Raspberry Pi* se implementa como *gateway* y servidor del sistema, encargada de recibir los datos publicados por los nodos, procesarlos y almacenarlos en una base de

datos de series temporales *InfluxDB*. Esta arquitectura permite concentrar la información en un punto central, mantener un registro histórico de las mediciones y facilitar su consulta por los servicios de análisis y visualización. En conjunto, esta configuración equilibra precisión, bajo consumo y una integración eficiente entre la capa de dispositivos y la de aplicación.

3.2.2 Sensores ambientales

Los sensores ambientales son dispositivos que miden variables físicas y químicas del entorno. En el contexto de este proyecto, se utilizan aquellos que miden la temperatura ambiente, la humedad relativa y la velocidad del viento.

Los sensores de temperatura suelen emplear termistores o sensores de temperatura digital, los cuales ofrecen lecturas precisas y rápidas [23, 24]. Los sensores de humedad miden la cantidad de vapor de agua presente en el aire, y muchos de ellos utilizan principios capacitivos o resistivos para determinar la humedad relativa [25].

Por último, la velocidad del viento se mide comúnmente con anemómetros, que pueden ser de tipo mecánico o electrónico. Su funcionamiento se basa en el principio de que el viento ejerce una fuerza sobre los elementos del anemómetro y se relaciona este movimiento con una variable que pueda medirse directamente [26]. Existen varios tipos de anemómetros, entre los cuales los más comunes son:

- **Anemómetro de copa:** Este tipo de sensor cuenta con tres o cuatro copas unidas a un eje. Cuando el viento sopla, las copas y el eje giran, y la velocidad de rotación se correlaciona con la velocidad del viento.
- **Anemómetro de palas:** Similar al de copa, este sensor utiliza palas en lugar de copas. El viento empuja las palas, provocando su rotación; la velocidad de giro se mide para determinar la velocidad del viento.
- **Anemómetro de hilo caliente:** Este sensor emplea un hilo delgado calentado eléctricamente. Cuando el viento sopla, enfría el hilo, y se mide la disminución de temperatura para calcular la velocidad del viento.

- **Anemómetro de presión:** Este tipo de sensor mide la presión del viento en una abertura; la presión cambia con la velocidad del viento, y este cambio se utiliza para calcular dicha velocidad.

La integración de microcontroladores con sensores ambientales permite la creación de estaciones meteorológicas modulares que pueden ser distribuidas en diferentes ubicaciones para cubrir un área efectiva más amplia. Estas estaciones son capaces de recolectar datos ambientales y transmitirlos mediante sistemas de comunicación inalámbrica hacia una unidad central, donde la información es procesada y analizada.

Para la medición de temperatura y humedad se emplea el sensor *DHT22* por su resolución y facilidad de integración, mientras que para la velocidad del viento se utiliza un anemómetro RS485 por su robustez en campo, buena inmunidad al ruido y posibilidad de cableados extensos hacia la *Raspberry Pi*. Este conjunto ofrece la precisión necesaria para el cálculo del índice F, manteniendo bajo consumo energético y alta mantenibilidad del sistema.

3.3 Tecnologías de comunicación inalámbrica

Los sistemas de monitoreo ambiental, especialmente aquellos distribuidos en grandes áreas geográficas, requieren de sistemas de comunicación inalámbrica para transmitir los datos capturados por los sensores hacia una estación central. Existen diversas tecnologías que se pueden utilizar dependiendo de la distancia, del consumo de energía y del tipo de datos que se necesita transmitir. Entre las opciones más relevantes se encuentran:

3.3.1 *Wi-Fi*

Es una tecnología estándar que permite la transmisión de datos en redes de área local (*LAN*). Aunque tiene un alcance limitado (alrededor de 100 metros en exteriores), es una opción viable cuando los sensores están ubicados cerca de una estación base o en zonas urbanas con infraestructura de red disponible. La facilidad de uso y la amplia disponibilidad de enrutadores *Wi-Fi* hacen que esta tecnología sea accesible

para muchos proyectos de monitoreo [27].

3.3.2 *LoRa*

LoRa es una tecnología de comunicación inalámbrica de baja potencia que permite la transmisión de datos a largas distancias (hasta 15 km en áreas rurales) con un consumo de energía extremadamente bajo. Esta característica la convierte en una opción ideal para proyectos de monitoreo ambiental en grandes áreas, donde los sensores se encuentran alejados entre sí y no es viable utilizar tecnologías de corto alcance. *LoRa* es particularmente útil en aplicaciones donde se requiere un bajo costo de operación a largo plazo [28].

3.3.3 *Zigbee*

Zigbee es otra tecnología de baja potencia diseñada para aplicaciones de *IoT*. Aunque su alcance es significativamente más corto que el de *LoRa* (alrededor de 10 a 100 metros), *Zigbee* permite la creación de redes de malla, lo que aumenta la confiabilidad del sistema al permitir que los datos sean transmitidos a través de múltiples nodos. Esta característica es especialmente valiosa en entornos donde la cobertura de señal es variable. *Zigbee* es popular en aplicaciones de domótica y monitoreo de sensores en áreas limitadas [28].

Aunque tecnologías como *LoRa* o *Zigbee* ofrecen mayor alcance o menor consumo, se seleccionó *Wi-Fi* por la naturaleza local de la red y la operación de la *Raspberry Pi* como punto de acceso dedicado. Esta elección simplifica la infraestructura, reduce costos de implementación y facilita la transferencia de datos y la administración remota, asegurando una comunicación estable y de alta velocidad dentro del área de cobertura del sistema.

3.4 Normas de telecomunicaciones y estándares ITU-T

Las normas *ITU-T Y.2060* y *ITU-T Y.4113* son esenciales para el desarrollo y la estandarización del *Internet de las Cosas (IoT)*, proporcionando un marco coherente y directo para la implementación de tecnologías *IoT*. Estas normas ayudan a definir las características, capacidades y funcionalidades que deben tener los sistemas *IoT*, asegurando una mayor interoperabilidad entre dispositivos y plataformas [29, 30].

3.4.1 *ITU-T Y.2060*

Esta norma establece una descripción general del *IoT*, definiendo conceptos clave y un modelo de referencia, como se muestra en la Figura 3.2, basado en capas: capa de aplicación, capa de apoyo a servicios y aplicaciones, capa de red y capa de dispositivo. La norma *ITU-T Y.2060* [29] establece las siguientes definiciones:

- **Capa de aplicación:** Contiene las aplicaciones *IoT*.
- **Capa de apoyo a servicios y aplicaciones:** Esta capa consiste en dos grupos de capacidades:
 - *Capacidades de soporte genéricas:* Son capacidades comunes que pueden utilizar diferentes aplicaciones *IoT*, tales como procesamiento o almacenamiento de datos. Estas capacidades también pueden ser empleadas por otras capacidades específicas.
 - *Capacidades de soporte específicas:* Son capacidades destinadas a atender las necesidades particulares de diversas aplicaciones, ofreciendo funciones de apoyo adaptadas a distintos contextos.
- **Capa de red:** Está compuesta por dos tipos de capacidades:
 - *Capacidades de red:* Ofrecen funciones de control de la conectividad en red, tales como control de acceso, gestión de recursos de transporte, movilidad y funciones de autenticación, autorización y contabilidad (AAA).

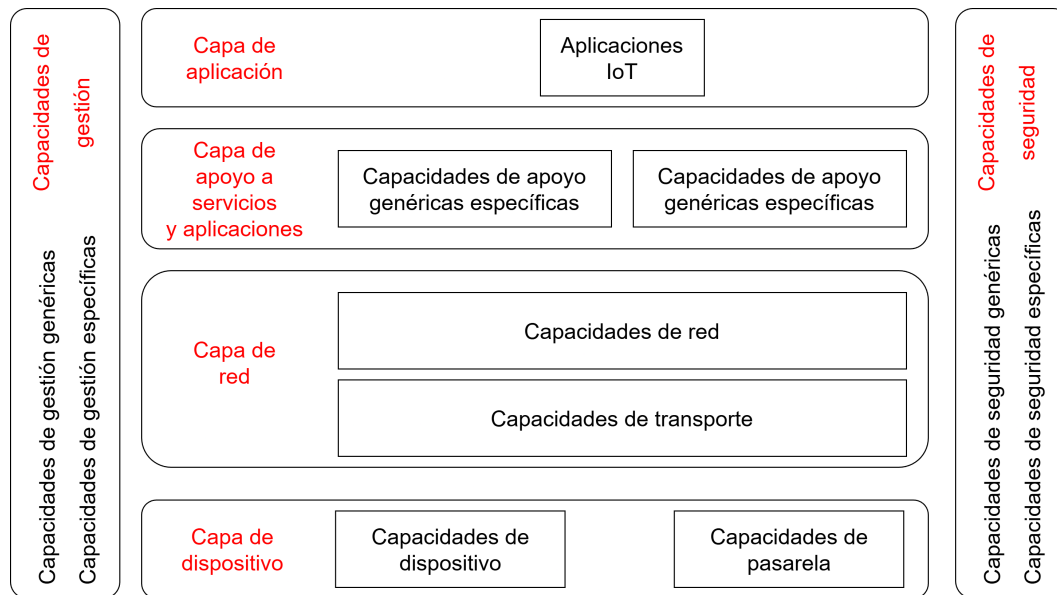
- *Capacidades de transporte*: Se centran en suministrar conectividad para el transporte de información y datos específicos de servicios y aplicaciones *IoT*, así como de la información de control y gestión relacionada.
- **Capa de dispositivo**: Presenta dos tipos de capacidades:
- *Capacidades de dispositivo*: Incluyen, entre otras, la interacción directa o indirecta con la red de comunicación. Los dispositivos pueden recopilar y cargar información directamente (sin recurrir a pasarelas) o hacerlo mediante capacidades de pasarela. También pueden recibir información (por ejemplo, instrucciones) tanto directa como indirectamente. Asimismo, pueden construir redes *ad-hoc* para aumentar la capacidad evolutiva del sistema y deben disponer de mecanismos de modo *reposo/activo* para optimizar el consumo energético.
 - *Capacidades de pasarela*: Permiten la interoperabilidad entre dispositivos y redes heterogéneas. Soportan interfaces múltiples para tecnologías alámbricas e inalámbricas como *CAN*, *Zigbee*, *Bluetooth* o *Wi-Fi*. También admiten tecnologías de red como *PSTN*, *2G/3G*, *LTE*, *Ethernet* o *DSL*. Además, realizan conversión de protocolos cuando las capas de dispositivo y red utilizan tecnologías distintas (por ejemplo, *Zigbee* en la capa de dispositivo y *3G* en la capa de red).

3.4.2 *ITU-T Y.4113*

Esta norma se centra en establecer los requerimientos de las aplicaciones basadas en redes *IoT* y ofrece un modelo más detallado de su arquitectura. La norma *ITU-T Y.4113* [30] propone el modelo básico que se muestra en la Figura 3.3 y define los siguientes componentes:

- **Modelo básico de *IoT***: El modelo básico de red para la *IoT* consta de tres partes principales: *core network* (red central), *access network* (red de acceso) y *IoT area network* (red de área *IoT*).

Figura 3.2. Modelo de referencia del *Internet de las Cosas* según la norma *ITU-T Y.2060* [29].

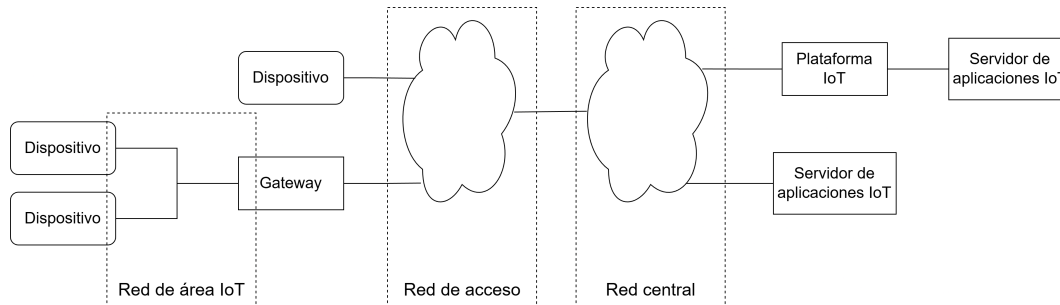


Fuente: Elaboración propia.

- **Device (Dispositivo):** Equipo con capacidades obligatorias de comunicación y capacidades opcionales de detección, actuación, captura de datos, almacenamiento y procesamiento de información.
- **Gateway (Puerta de acceso):** Unidad del *IoT* que interconecta los dispositivos con la red central, realizando la traducción necesaria entre los protocolos utilizados en la red central y los empleados por los dispositivos.
- **Core network (Red central):** Parte del sistema de suministro compuesta por redes, equipos e infraestructuras, que conecta el dominio del proveedor de servicios con la red de acceso.
- **Access network (Red de acceso):** Conecta los dispositivos *IoT* y las puertas de acceso con la red central. Puede implementarse mediante distintas tecnologías, como fibra óptica o tecnologías de acceso por radio.
- **IoT area network (Red de área IoT):** Red compuesta por dispositivos *IoT* y puertas de acceso interconectados. La conectividad se realiza a través de enlaces locales, normalmente utilizando tecnologías de comunicación de corto alcance.

- ***IoT platform (Plataforma IoT)***: Infraestructura técnica que proporciona la integración de capacidades genéricas y específicas. Estas capacidades, junto con las de la red central, pueden exponerse a uno o más servidores de aplicaciones *IoT*. La red central ofrece funcionalidades de comunicación para soportar la transferencia de datos entre dispositivos y pasarelas a través de la red de acceso.
- ***IoT application server (Servidor de aplicaciones IoT)***: Ejecuta aplicaciones y se comunica con los dispositivos, las puertas de acceso y la plataforma *IoT* a través de la red central (o directamente, en caso de comunicación con la plataforma) para prestar servicios de aplicación.

Figura 3.3. Modelo básico de red para la *Internet de las Cosas* propuesto en la norma *ITU-T Y.4113* [30].



Fuente: Elaboración propia.

Se adoptan las normas *ITU-T Y.2060* y *ITU-T Y.4113* como referencia arquitectónica para garantizar modularidad, interoperabilidad y escalabilidad en el diseño del sistema. Dichas normas estructuran la solución en capas (dispositivo, red, soporte de servicios y aplicación) y aseguran compatibilidad con futuras tecnologías de red o protocolos de transporte, fortaleciendo la alineación del proyecto con los estándares internacionales del *Internet de las Cosas*.

3.5 Protocolos de comunicación en *IoT*

Los protocolos de comunicación son un elemento esencial dentro del ecosistema del *Internet de las Cosas (IoT)*, ya que permiten el intercambio estructurado, confiable y seguro de datos entre dispositivos, pasarelas y servidores. Estos protocolos definen las reglas mediante las cuales se transmiten los mensajes, se establecen las conexiones y se

gestionan los errores de comunicación. Su elección depende de factores como el consumo energético, el ancho de banda disponible, la topología de red y el tipo de aplicación. Entre los más utilizados en sistemas *IoT* se encuentran *MQTT*, *CoAP* y *HTTP*, siendo el primero uno de los más extendidos por su eficiencia y ligereza [31, 32].

3.5.1 *Message Queuing Telemetry Transport (MQTT)*

El protocolo *MQTT* (*Message Queuing Telemetry Transport*) es un estándar de mensajería liviano basado en el paradigma *publicador-suscriptor*, ampliamente utilizado en sistemas *IoT* para la transmisión eficiente de datos entre dispositivos con recursos limitados. Fue desarrollado por Andy Stanford-Clark y Arlen Nipper en 1999 con el propósito de comunicar sensores y controladores a través de redes de baja velocidad [33].

MQTT opera sobre el protocolo *TCP/IP* y se caracteriza por su bajo consumo de ancho de banda y alta fiabilidad en entornos con conexiones inestables. El protocolo define tres componentes principales:

- **Publicador (*Publisher*):** Dispositivo o cliente que envía mensajes a un tema o canal (*topic*) determinado.
- **Suscriptor (*Subscriber*):** Dispositivo o cliente que recibe los mensajes publicados en los temas a los que está suscrito.
- **Servidor o corredor (*Broker*):** Entidad intermediaria responsable de gestionar la recepción y distribución de los mensajes entre publicadores y suscriptores.

Una de las principales ventajas de *MQTT* es su simplicidad en la implementación y su compatibilidad con múltiples plataformas de hardware, como *ESP32*, *Raspberry Pi* y otros microcontroladores. Además, soporta tres niveles de calidad de servicio (*QoS*) que permiten adaptar la fiabilidad de las transmisiones: *QoS 0* (entrega como máximo una vez), *QoS 1* (entrega al menos una vez) y *QoS 2* (entrega exactamente una vez). Estas características hacen de *MQTT* una opción idónea para proyectos de monitoreo ambiental y telemetría distribuidos [32, 31].

3.5.2 *Constrained Application Protocol (CoAP)*

El protocolo *CoAP* (*Constrained Application Protocol*) fue desarrollado por el *Internet Engineering Task Force (IETF)* como un protocolo ligero de transferencia de datos orientado a dispositivos con recursos limitados. Se basa en el modelo cliente-servidor y utiliza el protocolo *UDP* en lugar de *TCP*, lo que reduce la sobrecarga de comunicación y mejora la eficiencia energética en redes de sensores inalámbricos [34].

CoAP sigue una estructura similar a *HTTP*, pero está optimizado para funcionar en entornos con baja capacidad de procesamiento, memoria y ancho de banda. Emplea los métodos estándar *GET*, *POST*, *PUT* y *DELETE* para realizar operaciones sobre los recursos, y soporta el modelo de comunicación *request/response*. Además, permite la observación de recursos mediante un mecanismo de suscripción, lo que facilita su integración en sistemas *IoT* en tiempo real [31].

3.5.3 *Hypertext Transfer Protocol (HTTP)*

El protocolo *HTTP* (*Hypertext Transfer Protocol*) es uno de los protocolos más ampliamente utilizados para la comunicación en aplicaciones cliente-servidor en Internet. Funciona sobre el protocolo *TCP/IP* y permite el intercambio de información estructurada mediante el modelo *request/response*. Aunque no fue diseñado específicamente para entornos *IoT*, su compatibilidad universal lo convierte en una opción viable para la transmisión de datos en proyectos que no requieren optimización extrema del consumo energético [35].

En el contexto de los sistemas *IoT*, el protocolo *HTTP* se utiliza principalmente para la transferencia de información entre clientes y servidores, permitiendo la comunicación mediante solicitudes y respuestas estructuradas. Su simplicidad, compatibilidad y amplia adopción en Internet lo convierten en una opción práctica para aplicaciones donde la eficiencia energética no es una limitación crítica. No obstante, su mayor sobrecarga en comparación con protocolos diseñados específicamente para entornos de baja potencia, como *MQTT* y *CoAP*, puede restringir su uso en redes con recursos limitados o con alta latencia [31].

Se selecciona *MQTT* como protocolo principal de comunicación por su modelo publicador-suscriptor, baja sobrecarga, soporte de calidad de servicio y capacidad de operar de forma confiable en enlaces inestables. *CoAP* se consideró por su eficiencia sobre *UDP*, pero se descartó para simplificar la interoperabilidad y el manejo de errores; *HTTP* se mantiene únicamente en la capa de aplicación para exposición de servicios *REST*. Esta combinación optimiza el uso de recursos en los nodos y asegura compatibilidad con la infraestructura del servidor.

3.6 Servicios de aplicación y visualización de datos

En los sistemas *IoT*, la capa de aplicación constituye el nivel más alto dentro de la arquitectura de comunicación y tiene como propósito permitir la interacción entre los usuarios y la información procesada por el sistema. En esta capa se integran los servicios que gestionan los datos provenientes de los sensores, los transforman en información útil y los presentan de manera accesible a través de plataformas web o móviles. Esta capa también incluye los mecanismos que permiten la interoperabilidad entre distintos sistemas, donde las *API* y las aplicaciones cliente desempeñan un papel esencial para garantizar una comunicación eficiente y estandarizada [36].

3.6.1 Interfaces de programación de aplicaciones (*API*)

Las *Interfaces de Programación de Aplicaciones (API)* son conjuntos de definiciones, reglas y protocolos que permiten la comunicación entre diferentes componentes de software. En los sistemas *IoT*, las API son esenciales para conectar la capa de procesamiento de datos con las aplicaciones cliente o servicios externos que consumen dicha información. Utilizando métodos estandarizados de comunicación, como *HTTP* o *MQTT*, las API facilitan el intercambio de datos entre los servidores y las plataformas de visualización o análisis, permitiendo la integración entre diversos sistemas y dispositivos heterogéneos [37].

Las API pueden seguir diferentes estilos arquitectónicos, entre los cuales el modelo *REST (Representational State Transfer)* se ha convertido en uno de los más adoptados por su simplicidad, escalabilidad e independencia del lenguaje de programación. En

este modelo, los recursos son accedidos mediante identificadores únicos y manipulados mediante operaciones estándar como *GET*, *POST*, *PUT* y *DELETE*. Esta arquitectura es ampliamente utilizada en plataformas *IoT* modernas debido a su compatibilidad con entornos distribuidos y su facilidad para integrarse con servicios en la nube [38].

3.6.2 Aplicaciones web y móviles

Las aplicaciones web y móviles conforman la interfaz final del ecosistema *IoT*, encargada de la visualización, monitoreo y gestión de los datos recolectados. A través de ellas, los usuarios pueden interpretar en tiempo real la información proveniente de los sensores, recibir alertas automáticas ante condiciones críticas e incluso enviar comandos de control al sistema. Estas aplicaciones cumplen una función clave en la comunicación *hombre-máquina* (*HMI*), facilitando la toma de decisiones basada en datos [39].

El desarrollo de este tipo de interfaces se apoya en tecnologías modernas como *Node.js*, *React* o *Kotlin*, que permiten crear aplicaciones interactivas, escalables y multiplataforma. *Node.js* ofrece un entorno de ejecución de alto rendimiento basado en JavaScript para la creación de servidores y *API REST*, mientras que *React* se utiliza en el diseño de interfaces web dinámicas y *Kotlin* facilita el desarrollo de aplicaciones móviles nativas para el sistema operativo Android [40]. En el contexto del presente proyecto, estas tecnologías permiten construir un entorno accesible para el análisis de las variables ambientales y la generación de alertas tempranas de riesgo de incendio.

En conclusión, la capa de aplicación en los sistemas *IoT* integra los servicios de procesamiento, almacenamiento y visualización de la información, permitiendo que los datos generados por los dispositivos se transformen en conocimiento útil para la toma de decisiones. El uso de *API REST* facilita la interoperabilidad entre plataformas, mientras que las aplicaciones web y móviles proporcionan acceso flexible y en tiempo real a los datos procesados. Esta integración entre software de servidor, bases de datos de series temporales y entornos de visualización interactiva constituye un componente esencial en el desarrollo de soluciones inteligentes orientadas al monitoreo ambiental y la gestión de riesgos.

4. Metodología

El sistema desarrollado se basa en una arquitectura modular distribuida en cuatro capas principales, conforme a las recomendaciones de las normas *ITU-T Y.2060* y *ITU-T Y.4113*. Estas capas corresponden a los niveles de dispositivo, red, procesamiento y aplicación, los cuales interactúan de manera jerárquica para garantizar la recolección, transmisión, almacenamiento y visualización de los datos ambientales y el posterior cálculo del índice F.

En la **capa de dispositivos** se encuentran los nodos sensores basados en el microcontrolador *ESP32* encargados de medir la temperatura y la humedad relativa, además de un anemómetro para medir la velocidad del viento, estas variables ambientales son necesarias para la estimación del riesgo de incendio. Los datos capturados son procesados localmente y transmitidos mediante el protocolo *MQTT* hacia la siguiente capa.

La **capa de procesamiento** está conformada por la *Raspberry Pi*, que actúa como *gateway* del sistema, en la que se ejecutan los servicios de recolección, almacenamiento y análisis de datos. Los mensajes recibidos desde los sensores son almacenados en una base de datos de series temporales *InfluxDB*, donde se registran las mediciones y se calcula el índice F para estimar el nivel de riesgo de incendio.

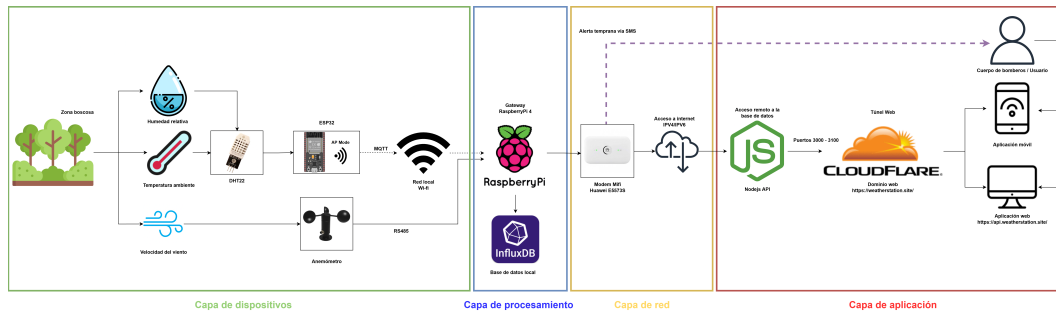
La **capa de red** proporciona la conectividad inalámbrica necesaria para el intercambio de información entre los nodos y la *Raspberry Pi*. Se implementa mediante una red *Wi-Fi* local en la que la *Raspberry Pi* actúa como punto de acceso y servidor intermedio, garantizando una comunicación eficiente dentro del área de cobertura. Adicionalmente, la conexión a Internet y el envío de alertas externas se realizan a través de un módem *MiFi (Mobile Wi-Fi) Huawei E5573 (modelo E5573)*, que establece el enlace de salida de la red local hacia servicios remotos y permite la comunicación con usuarios o entidades externas.

Finalmente, la **capa de aplicación** corresponde al nivel de interacción con el usuario. En esta capa se integran las interfaces web y móvil, desarrolladas con

tecnologías como *Node.js*, *React* y *Kotlin*, que permiten la visualización en tiempo real de las variables monitoreadas y el envío de alertas cuando se detectan condiciones críticas.

El diseño modular adoptado facilita la interoperabilidad entre las capas, la escalabilidad del sistema y la posibilidad de incorporar nuevos sensores o servicios sin modificar la estructura general. En la Figura 4.1 se presenta el diagrama general de la arquitectura del sistema.

Figura 4.1. Arquitectura general del sistema de alerta temprana de incendios forestales.



Fuente: Elaboración propia.

4.1 Capa de dispositivos

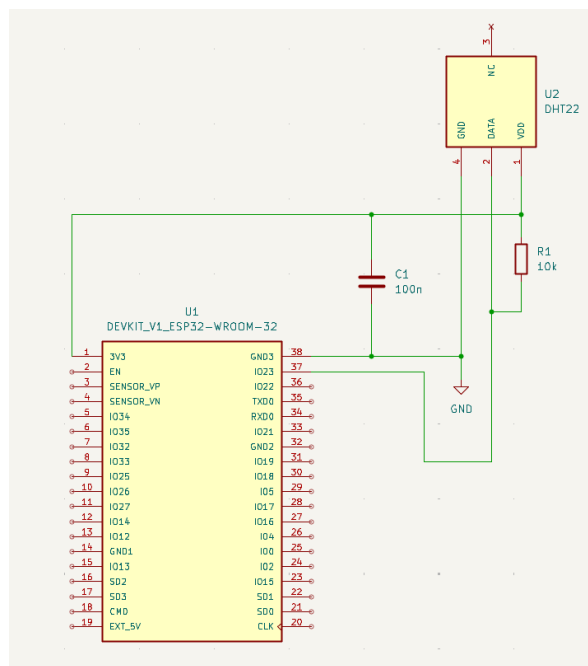
Esta capa corresponde al nivel físico de la arquitectura IoT y constituye la base para la adquisición de datos ambientales dentro del sistema. Está conformada por nodos autónomos basados en microcontroladores *ESP32*, encargados de medir temperatura y humedad relativa mediante sensores digitales *DHT22*, y por un módulo anemométrico con interfaz EIA-485 (*RS485*) dedicado a la medición de la velocidad del viento. Las lecturas de cada nodo se transmiten a la *Raspberry Pi* mediante el *MQTT broker* (*Mosquitto*), asegurando un flujo continuo y sincronizado de información. El diseño sigue los lineamientos de las recomendaciones *ITU-T Y.2060* y *ITU-T Y.4113*, priorizando dispositivos de baja complejidad y consumo energético, identificación única mediante el parámetro `sensor_id`, y un acoplamiento débil con las capas superiores a través de *MQTT topics* jerárquicos y mensajes en formato JavaScript Object Notation (JSON). Gracias a esta estructura modular, los nodos pueden operar de forma autónoma dentro

de la red local y escalar fácilmente la infraestructura mediante la incorporación de nuevos sensores o estaciones. El firmware se desarrolla bajo un esquema declarativo con *ESPHome*, utilizando archivos de configuración YAML Ain't Markup Language (YAML) reproducibles que permiten la estandarización del formato de mensajes, la publicación de estados operativos y la recuperación automática ante fallos de red o alimentación.

4.1.1 Arquitectura física de los nodos

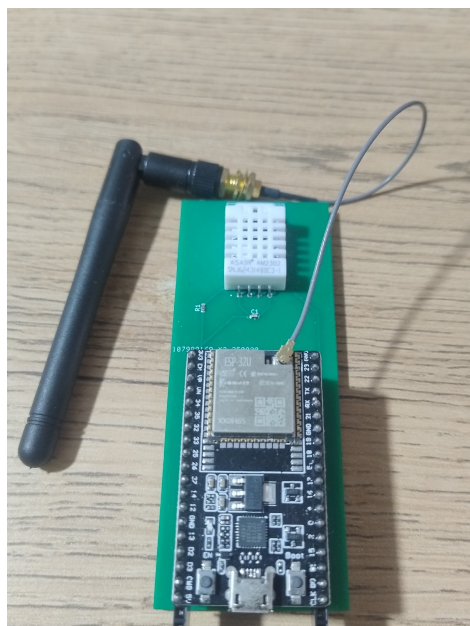
Los nodos sensores de humedad y temperatura (nodo ESP32) están conformados por un microcontrolador *ESP32 DevKit V1* y un sensor digital *DHT22*, encargados de medir la temperatura y la humedad relativa del entorno. El pin de datos del DHT22 se conecta al GPIO23 (General Purpose Input/Output) del ESP32 mediante una resistencia de $10\text{k}\Omega$ de *pull-up*. Este pin fue seleccionado para minimizar la longitud de la pista en el diseño de la PCB y reducir la interferencia en la señal digital. La alimentación del DHT22 se obtiene directamente del regulador interno de 3.3V del ESP32, con referencia común a GND (tierra). Además, se incluye un condensador de desacoplo de 100nF entre VCC (tensión de alimentación) y GND para estabilizar la línea de alimentación del sensor. La Figura 4.2 muestra el esquema de conexión empleado entre el ESP32 y el sensor DHT22, mientras que la Figura 4.3 presenta el montaje físico del nodo sobre la placa de circuito impreso (PCB).

Figura 4.2. Esquemático de conexión del nodo ESP32 con el sensor DHT22.



Fuente: Elaboración propia.

Figura 4.3. Montaje físico del nodo ESP32 sobre PCB.



Fuente: Elaboración propia.

El sistema también integra un nodo anemómetro con interfaz industrial *RS485* (*EIA-485*) (*DFRobot SEN0483*) [41], utilizado para medir la velocidad del viento. Este dispositivo posee cuatro cables: dos correspondientes a la línea de comunicación diferencial RS485 (*A* y *B*) y dos para la alimentación eléctrica (*VCC* y *GND*). El anemómetro requiere una tensión de alimentación de 12 V en corriente continua, por lo que se implementó un convertor *AC/DC* (corriente alterna a corriente continua) de 12 V soldado a los cables de alimentación del sensor. Para la comunicación con la capa de procesamiento, los cables de datos RS485 se conectan a un adaptador *RS485-USB* (*Universal Serial Bus*), que permite su integración directa con la *Raspberry Pi*, encargada de adquirir los datos de velocidad del viento. La Figura 4.4 ilustra el esquema de conexión del anemómetro con la *Raspberry Pi*.

Figura 4.4. Conexión del anemómetro RS485 (DFRobot SEN0483) a la Raspberry Pi mediante convertor RS485-USB y fuente de 12 V.



Fuente: Elaboración propia.

Cuadro 4.1. Conexiones principales del nodo sensor basado en ESP32.

Componente	Conexión	Descripción
DHT22 (datos)	GPIO23 (ESP32)	Señal digital de temperatura y humedad con resistencia de <i>pull-up</i> a 3.3 V.
Resistencia de <i>pull-up</i>	10 k Ω a 3V3	Estabiliza la línea de datos y evita falsos disparos.
Alimentación DHT22	3.3 V / GND	Desde el regulador del ESP32.
Condensador de desacoplo	100 nF (VCC–GND)	Filtra ruido en la alimentación del DHT22.

Fuente: Elaboración propia.

Cuadro 4.2. Conexiones principales del nodo anemómetro RS485.

Componente	Conexión	Descripción
Anemómetro RS485 (DFRobot SEN0483)	A / B (RS485)	Líneas diferenciales conectadas al adaptador RS485–USB de la Raspberry Pi.
Alimentación anemómetro	12 V / GND	Desde conversor AC/DC 12 V conectado al sensor.

Fuente: Elaboración propia.

4.1.2 ESPHome y archivos YAML: concepto, implementación y control de calidad de datos

ESPHome es un entorno de desarrollo que permite generar firmware de manera declarativa para microcontroladores *ESP32* y *ESP8266*. A diferencia del enfoque imperativo tradicional, en el que se programa línea por línea en lenguajes como *C++*, *ESPHome* utiliza archivos de configuración en formato YAML, donde el desarrollador describe de forma estructurada los componentes del sistema, las conexiones físicas, los

eventos que los activan y las acciones que deben ejecutarse. Con base en esta descripción, la herramienta compila automáticamente el firmware final, integrando los controladores, bibliotecas y protocolos definidos [42].

Este enfoque simplifica el desarrollo y la estandarización del firmware de los nodos, ofreciendo las siguientes ventajas:

- Reproducibilidad: los nodos pueden replicarse modificando únicamente su identificador lógico o los parámetros de red, sin alterar la lógica principal del sistema.
- Mantenibilidad: las reglas de adquisición, publicación y reconexión se declaran directamente en el archivo `.yaml`, eliminando la necesidad de escribir o mantener código fuente en `C++`.
- Observabilidad: los *MQTT topics* de estado, depuración y telemetría se generan automáticamente, permitiendo el monitoreo del funcionamiento de cada nodo desde el *MQTT broker*.

En el presente proyecto, el firmware del microcontrolador *ESP32* fue desarrollado íntegramente mediante *ESPHome*, empleando un archivo de configuración YAML que describe la estructura lógica y las reglas de operación del nodo sensor. En dicho archivo se definen los parámetros de red, el identificador único del dispositivo (`sensor_id`), los sensores físicos conectados, los *MQTT topics* de comunicación y el comportamiento ante eventos de sincronización o falla de red. El microcontrolador *ESP32 DevKit V1* se configuró con el sensor *DHT22* conectado al pin `GPIO23`, una precisión de un decimal para las variables de temperatura y humedad, y una red *Wi-Fi* con punto de acceso de respaldo (*fallback*) que asegura la reconexión automática ante interrupciones.

El sensor *DHT22* se declaró con el parámetro `update_interval: never`, deshabilitando las lecturas automáticas periódicas. De este modo, las mediciones se ejecutan únicamente al recibir un `timestamp` enviado desde la *Raspberry Pi* a través del *MQTT topic* `sync/timestamp`. El contenido de este mensaje corresponde a un `timestamp` en formato ISO 8601 (UTC), como se muestra a continuación:

2025-10-23T05:45:10Z

Esta política garantiza la coherencia temporal entre todos los nodos de la red, de modo que las lecturas ambientales se mantengan perfectamente sincronizadas con la capa de procesamiento.

Una vez recibido el `timestamp`, el nodo ejecuta la lectura del sensor y publica un mensaje JSON en el *MQTT topic* `data/sensor/{sensor_id}` usando el mismo `timestamp` y con el siguiente formato:

```
{
  "sensor_id": 1,
  "temperature": 29.7,
  "humidity": 64.6,
  "timestamp": "2025-10-23T05:45:10Z"
}
```

El campo `sensor_id` identifica de manera única a cada nodo dentro del sistema, lo que permite distinguir lecturas equivalentes (por ejemplo, de temperatura o humedad) provenientes de diferentes sensores en la base de datos de series temporales. Adicionalmente, el nodo publica mensajes de estado y depuración en los *MQTT topics* `weatherstation/status` y `weatherstation/debug`, lo que permite el monitoreo remoto de su funcionamiento y facilita la trazabilidad de eventos.

Cada lectura pasa por un conjunto de validaciones locales para garantizar la calidad de los datos:

- Temperatura válida entre -40°C y 60°C .
- Humedad relativa entre 0 y 100 %.
- Descarte automático de valores no numéricos o fuera de rango, con registro de la incidencia en `weatherstation/debug`.

Asimismo, se implementó un contador de errores consecutivos que se incrementa cada vez que una lectura resulta inválida o se produce una excepción en el sensor. Al alcanzar cinco errores consecutivos, el microcontrolador ejecuta un reinicio controlado que restaura el funcionamiento normal. Cuando se obtiene una lectura válida, el contador se restablece a cero y los datos se publican correctamente al *MQTT broker*.

El conjunto de *MQTT topics* y su función dentro del sistema se resume de la siguiente manera:

- `sync/timestamp`: recepción del `timestamp` en formato ISO 8601 (UTC) desde la *Raspberry Pi*.
- `data/sensor/{sensor_id}`: publicación de lecturas válidas de temperatura y humedad en formato JSON.
- `weatherstation/status`: estado operativo del nodo (`online/offline`) con retención activa.
- `weatherstation/debug`: mensajes de diagnóstico, errores y trazas de ejecución.

La configuración completa del archivo YAML utilizado en el nodo sensor está disponible públicamente en el repositorio del proyecto [43].

4.1.3 Diseño del circuito impreso (PCB)

Para la integración física del microcontrolador *ESP32 DevKit V1* y el sensor *DHT22*, se diseñó una placa de circuito impreso (PCB) dedicada, cuyo propósito es proporcionar un montaje compacto, ordenado y confiable, reduciendo el cableado y mejorando la estabilidad eléctrica del sistema. El diseño se realizó en el software *KiCad*, considerando los requerimientos eléctricos, térmicos y de disposición mecánica del nodo sensor.

La PCB cuenta con las siguientes características principales:

- Dimensiones de 40 mm × 100 mm, con espesor de 1.6 mm.

- Configuración de dos capas conductoras (F.Cu y B.Cu).
- Tres redes funcionales principales:
 - 3V3: línea de alimentación de 3.3 V proveniente del regulador interno del ESP32.
 - GND: plano de tierra común, extendido sobre la capa inferior (B.Cu) para garantizar un retorno de corriente de baja impedancia y reducir interferencias electromagnéticas.
 - IO23: línea digital de comunicación entre el microcontrolador y el sensor DHT22, utilizada exclusivamente para la transmisión de datos de temperatura y humedad.
- Resistencia de 10 k Ω de *pull-up* entre 3V3 e IO23 para estabilizar la línea de datos del DHT22.
- Condensador de desacoplo de 100 nF entre 3V3 y GND, ubicado junto al sensor para filtrar ruido en la alimentación.
- Plano de tierra continuo en la capa inferior (B.Cu) para mejorar la integridad de señal y el retorno de corriente.
- Huella del módulo *ESP32 DevKit V1* con encabezados estándar de 2.54 mm.

El sensor *DHT22* se dispuso en el borde de la placa para permitir su exposición directa al ambiente y mejorar la precisión de las mediciones, mientras que el *ESP32* se ubicó en la zona central para optimizar la distribución de pistas y mantener una buena disipación térmica. Con el fin de minimizar el acoplamiento de ruido y las interferencias electromagnéticas, se evitó el enrutamiento y el vertido de cobre directamente debajo del sensor y del microcontrolador, preservando zonas libres de plano metálico en dichas áreas sensibles. El trazado de las pistas prioriza un retorno de tierra (GND) corto hacia el condensador de desacoplo y una línea de señal limpia en IO23, garantizando una lectura estable del DHT22 incluso ante variaciones de alimentación o interferencia ambiental.

La alimentación del *ESP32* se realiza a través de su puerto *USB-C*, lo que facilita tanto la programación como la operación del dispositivo sin requerir una fuente externa

adicional. Durante las pruebas de campo, el sistema se alimentó mediante una *powerbank* con panel solar incorporado, lo que proporcionó autonomía energética y permitió el funcionamiento continuo del nodo en entornos sin acceso a red eléctrica. No obstante, el tiempo exacto de autonomía no se especifica, ya que depende de la capacidad de la batería y del consumo del microcontrolador, el cual varía según la intensidad de la señal *Wi-Fi*, la frecuencia de transmisión y las condiciones ambientales.

La Figura 4.2 presenta el esquemático eléctrico del nodo; las Figuras A.3 y A.4 muestran las vistas superior e inferior de la PCB, respectivamente, generadas desde el entorno de diseño.

4.2 Capa de procesamiento

La capa de procesamiento está implementada en una *Raspberry Pi 4 Model B*, que actúa como *gateway*, servidor de datos y unidad central de control. Coordina la recepción, validación y almacenamiento de las mediciones de los nodos basados en *ESP32* y del anemómetro RS485, publica `timestamp` centralizados y ejecuta el cálculo del índice F. Además, acciona el sistema de alertas por *SMS* cuando se detectan condiciones críticas. De este modo, la *Raspberry Pi* integra la red local con las aplicaciones de visualización y monitoreo remoto, manteniendo el flujo de información entre capas.

4.2.1 Arquitectura de procesos en la Raspberry Pi

El funcionamiento en la *Raspberry Pi* se organiza en servicios desarrollados en *Python*, ejecutados como *daemons* gestionados por *systemd*, lo que garantiza su arranque automático tras reinicio, la supervisión continua, el reinicio en caso de fallo y el registro persistente de eventos. Todos los servicios se comunican mediante el *MQTT broker* ejecutado localmente en la *Raspberry Pi*. Cada proceso crea un cliente *MQTT* independiente con identificador propio, encargado de suscribirse y publicar en los *MQTT topics* jerárquicos definidos (por ejemplo, `sync/timestamp`, `data/sensor/{id}`, `data/sensor/anemometer`). El *MQTT topic* `sync/timestamp` se publica con retención para que los nodos recuperen el último `timestamp` al reconectar,

mientras que los *MQTT topics* de datos se publican sin retención. Toda la comunicación ocurre dentro de la red inalámbrica local *WeatherStation*, creada por la *Raspberry Pi* en modo punto de acceso; de este modo, los nodos *ESP32*, el anemómetro y la *Raspberry Pi* comparten el mismo dominio de red y operan de forma autónoma. Los `timestamp` se publican en UTC y las interfaces de usuario convierten los datos a la zona horaria del dispositivo donde se ejecutan. Los servicios registran su actividad en `/home/pi/TG/logs` para trazabilidad y diagnóstico.

Los servicios principales son: `Main.py`, encargado de la configuración *MQTT*, la sincronización temporal y la escritura en *InfluxDB*; `wind_reader.py`, responsable de la lectura del anemómetro RS485; `F_index.py`, que calcula el índice F a partir de las mediciones alineadas en el tiempo; y `sms_alert.py`, que supervisa el índice F y gestiona el envío de alertas por *SMS*. Esta estructura modular permite el aislamiento de tareas, la depuración independiente y la escalabilidad del sistema.

4.2.2 Adquisición de datos del anemómetro

El anemómetro RS485 se encuentra conectado directamente a la *Raspberry Pi* mediante un adaptador *RS485-USB*. El servicio `wind_reader.py` consulta periódicamente el instrumento utilizando el protocolo de comunicación industrial *Modbus RTU (Remote Terminal Unit)*. Cada vez que la *Raspberry Pi* publica un nuevo `timestamp` en el *MQTT topic* `sync/timestamp`, el proceso se activa para leer la velocidad del viento y generar un mensaje con su valor instantáneo en metros por segundo (m/s). Dicha lectura se publica en el *MQTT topic* `data/sensor/anemometer` y se almacena simultáneamente en la base de datos *InfluxDB* bajo la medición `wind_data`.

El mensaje publicado mantiene la misma estructura JSON utilizada por los nodos *ESP32*, con el fin de unificar el formato de intercambio de datos en todo el sistema. Ejemplo de publicación:

```
{
  "sensor_id": 2,
  "wind_speed": 3.8,
```

```
"timestamp": "2025-10-23T05:45:10Z"  
}
```

donde `sensor_id` identifica el sensor de origen del dato (por ejemplo, 2 para el anemómetro RS485), `wind_speed` representa la velocidad del viento en m/s, y `timestamp` corresponde al valor en formato ISO 8601 (UTC) generado por la *Raspberry Pi*.

El script incorpora verificación de integridad, detección de errores de comunicación y reintentos automáticos ante fallos o pérdida temporal del adaptador. La comunicación Modbus RTU opera a 9600 bps, 8N1 y dirección de esclavo 2; la velocidad se lee en el registro 0x0000 y se escala a m/s. El código fuente completo se encuentra disponible en el repositorio del proyecto [43], dentro del directorio `RaspberryPi/wind_reader.py`.

4.2.3 Sincronización temporal, ingesta de datos y almacenamiento en InfluxDB

El servicio `Main.py` cumple tres funciones esenciales: la sincronización temporal del sistema, la ingesta de mediciones desde los nodos *ESP32* y la escritura en la base de datos *InfluxDB*. Para mantener la coherencia entre todos los nodos, la *Raspberry Pi* publica periódicamente un `timestamp` en el *MQTT topic* `sync/timestamp`, en formato ISO 8601 (UTC) y con intervalo configurable. Durante las pruebas experimentales se empleó un periodo de 10 s para facilitar la validación empírica; sin embargo, en operación normal se recomienda un intervalo de 60 s, dado que las variables ambientales presentan variaciones lentas y una frecuencia menor contribuye a reducir el consumo energético de los microcontroladores.

Cada nodo *ESP32* ejecuta la lectura del sensor DHT22 al recibir este mensaje, garantizando la sincronización de las mediciones de temperatura, humedad y viento. El servicio se suscribe al *MQTT topic* `data/sensor/{sensor_id}` para recibir los datos publicados por los nodos; cada mensaje JSON contiene los campos `sensor_id`, `temperature`, `humidity` y `timestamp`. Tras validar su estructura y tipos de datos, las mediciones se escriben en *InfluxDB* mediante el *Line Protocol* con precisión de segundos.

La base de datos de series temporales utiliza una política de retención de 30 días y se organiza en tres mediciones principales:

- `sensor_data`: temperatura [°C] y humedad relativa [%], con etiqueta `sensor_id`.
- `wind_data`: velocidad del viento [m/s] proveniente de `wind_reader.py`.
- `f_index`: índice F (adimensional) calculado por `F_index.py`.

El servicio maneja reconexiones automáticas al *MQTT broker* y al motor de *InfluxDB*, registrando los eventos en los archivos de *log*. El código fuente completo se encuentra disponible en el repositorio del proyecto [43], dentro del directorio `RaspberryPi/Main.py`.

4.2.4 Cálculo del índice F

El servicio `F_index.py` obtiene, para cada `timestamp`, las lecturas válidas de temperatura (T), humedad relativa (H) y velocidad del viento (U) provenientes de los mensajes *MQTT* y registradas en la base de datos *InfluxDB*. La velocidad del viento se convierte a kilómetros por hora (km/h) mediante:

$$U_{\text{km/h}} = 3.6 \times U_{\text{m/s}} \quad (4.1)$$

A partir de estas variables se calcula el *Fuel Moisture Index* (FMI), que representa una estimación de la humedad del material combustible:

$$\text{FMI} = 10 - 0.25 (T - H) \quad (4.2)$$

y posteriormente se obtiene el índice F, empleado para estimar el nivel de riesgo de incendio:

$$F = \frac{\text{máx}(U_0, U_{\text{km/h}})}{\text{FMI}} \quad (4.3)$$

donde $U_0 = 1$ km/h corresponde a la velocidad mínima efectiva utilizada para evitar valores nulos o inestables. En todos los casos, T se expresa en °C, H en %, U en km/h, y tanto FMI como F son adimensionales.

El sistema exige que las tres variables —temperatura, humedad relativa y velocidad del viento— correspondan exactamente al mismo `timestamp`. La ventana de sincronización es estricta (0 s); si alguna variable falta o genera un $\text{FMI} \leq 0$, el cálculo se omite y el evento se registra en el archivo de *log*. El resultado se publica en `data/sensor/calcf` y se almacena en la medición `f_index` de *InfluxDB*. El script implementa deduplicación por `timestamp`, control de valores extremos e histéresis para estabilizar la señal ante ruido instrumental. El código fuente completo se encuentra disponible en el repositorio del proyecto [43], dentro del directorio `RaspberryPi/F_index.py`.

4.2.5 Sistema de alerta por SMS

El servicio `sms_alert.py` monitorea continuamente los valores del índice F registrados en *InfluxDB* o recibidos por *MQTT*, y evalúa su magnitud frente a un umbral de alerta configurado. En la implementación se utiliza el umbral $F > 1.5$, valor que marca el inicio del nivel alto de riesgo de acuerdo con la escala propuesta por Sharples et al. [13] (Cuadro 3.1, Capítulo 3).

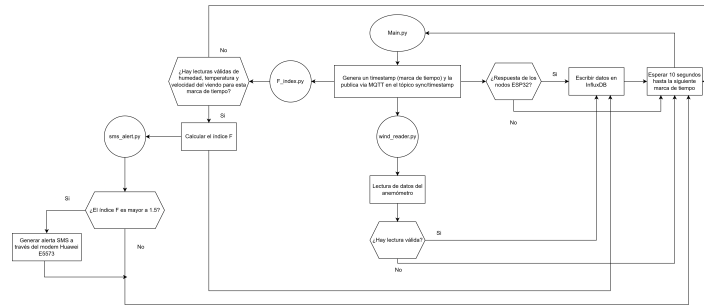
Cuando se supera este valor, el sistema envía un mensaje de texto a través del módem *MiFi Huawei E5573 (modelo E5573)* utilizando la biblioteca `huawei_lte_api`. El procedimiento incluye autenticación, envío y verificación del estado del mensaje, con hasta tres intentos (uno inicial y dos reintentos) y un incremento exponencial del tiempo de espera entre ellos. Asimismo, se aplica un *cooldown* de 60 s por destinatario para evitar tormentas de notificaciones; no se emplea histéresis adicional más allá de ese intervalo. Cada operación se registra en el archivo de *log*. El número de destino se define explícitamente en el propio script. El mensaje enviado incluye el valor del índice F, el umbral de referencia y la hora local en Colombia (zona horaria *America/Bogota*).

El código fuente completo se encuentra disponible en el repositorio del proyecto [43], dentro del directorio `RaspberryPi/sms_alert.py`.

4.2.6 Diagrama de flujo de datos

La Figura 4.5 resume el flujo de datos extremo a extremo y la interacción entre los servicios `Main.py`, `wind_reader.py`, `F_index.py` y `sms_alert.py`, así como los puntos de integración con *MQTT*, *InfluxDB* y el módem *MiFi Huawei E5573*.

Figura 4.5. Flujo de datos e interacción entre servicios: sincronización temporal, adquisición, cálculo de índice F y alerta por SMS.



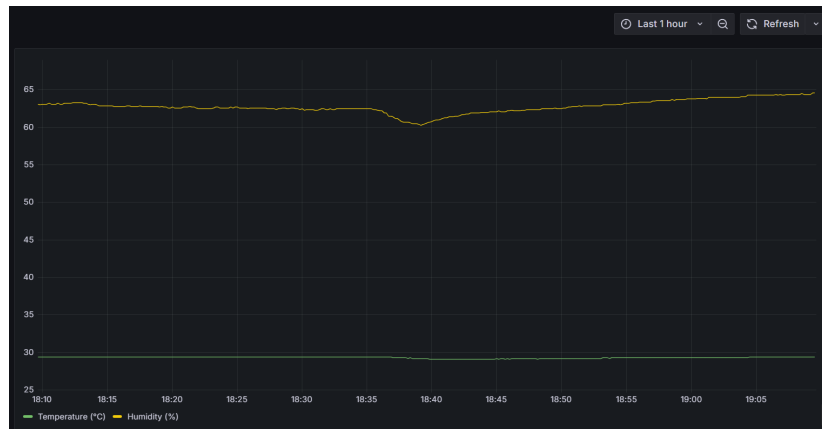
Fuente: Elaboración propia.

4.2.7 Visualización local con Grafana

Para la supervisión y validación operativa de las mediciones almacenadas en la base de datos, se utiliza la plataforma de visualización *Grafana*. Esta es una herramienta de código abierto que permite crear paneles interactivos para la representación gráfica de series temporales provenientes de diversas fuentes de datos, como *InfluxDB*. En este proyecto, *Grafana* se empleó no solo para monitorear en tiempo real las variables ambientales —temperatura, humedad y velocidad del viento— y el valor calculado del índice F, sino también como entorno de referencia para la validación de la página web y de la aplicación móvil, verificando la coherencia de los datos mostrados y la correcta actualización de las lecturas. Su integración con la instancia local de *InfluxDB* permite una visualización dinámica y configurable, lo que facilita el análisis rápido del estado del sistema, la comparación entre interfaces de usuario y el diagnóstico del funcionamiento

general, como se ilustra en la Figura 4.6.

Figura 4.6. Interfaz de referencia de *Grafana* empleada para la visualización de las mediciones ambientales (temperatura y humedad) en tiempo real.



Fuente: Elaboración propia.

4.2.8 Resumen de servicios y ejecución como *daemons*

El Cuadro 4.3 resume los servicios ejecutados en la *Raspberry Pi* y su función principal. Todos se ejecutan como *daemons* gestionados por *systemd*, con arranque automático, reinicio en caso de fallo (*on-failure*) y registro persistente de actividad.

Cuadro 4.3. Servicios ejecutados en la Raspberry Pi y su función principal

Servicio	Función principal
<code>Main.py</code>	Publicación periódica de <code>timestamp</code> y registro de datos de temperatura y humedad en la base de datos.
<code>wind_reader.py</code>	Lectura del anemómetro RS485 sincronizada con los <code>timestamp</code> publicados por la Raspberry Pi.
<code>F_index.py</code>	Cálculo del índice F a partir de las variables ambientales y supervisión del umbral de riesgo de incendio.
<code>sms_alert.py</code>	Envío de alertas por <i>SMS</i> cuando el índice F supera el umbral crítico.

Fuente: Elaboración propia.

4.3 Capa de red

Esta capa garantiza la conectividad entre los dispositivos de medición y la unidad central, delimitando un dominio local estable y un enlace de salida controlado hacia Internet. La red se implementa con la *Raspberry Pi* como punto de acceso *Wi-Fi* y puerta de enlace (*gateway*) de la subred de sensores, donde el tráfico se mantiene dentro de la *LAN* y solo se enruta al exterior cuando es necesario mediante traducción de direcciones de red (*NAT*, por sus siglas en inglés *Network Address Translation*). La conexión hacia Internet se realiza a través de un módem *MiFi Huawei E5573* configurado como interfaz *WAN*, lo que permite mantener una comunicación controlada con los servicios remotos sin comprometer la seguridad interna del sistema. Con este diseño, la estación central preserva su operatividad local aun ante pérdidas temporales de Internet, manteniendo la transmisión de datos dentro de la red local y garantizando la continuidad de las mediciones, mientras la exposición externa se limita estrictamente a los servicios que requieren acceso remoto.

4.3.1 Infraestructura y topología de la red local

La *Raspberry Pi* actúa como nodo central y punto de acceso sobre la interfaz `wlan0` (modo *AP/Master*), con *SSID WeatherStation*, en la banda de 2.4GHz (canal 1, ancho 20MHz) y potencia de transmisión de 31dBm. El dominio regulatorio se fija en CO y se habilita el estándar *IEEE 802.11n* con *Short GI* de 20MHz, suficiente para las tasas requeridas por los sensores y los servicios locales. La topología adoptada es de tipo estrella, con la *Raspberry Pi* en el centro del dominio inalámbrico actuando como servidor de datos y coordinador de la red, mientras que los nodos *ESP32* operan como estaciones (*STA*) que publican su telemetría hacia el *broker* de mensajes alojado en la misma. El anemómetro *RS485*, aunque no forma parte de la red *IP*, está conectado físicamente a la *Raspberry Pi* mediante un conversor *RS485-USB*, integrándose lógicamente en el mismo dominio de adquisición de datos.

La subred *LAN* configurada para el punto de acceso corresponde a 192.168.50.0/24, con la *Raspberry Pi* asignada como *gateway* principal (192.168.50.1) y encargada de ofrecer los servicios *DHCP* y *DNS* a los clientes conectados. El tráfico saliente se enruta hacia la interfaz `eth1` mediante *NAT*, lo que

permite compartir la conexión *WAN* del módem *MiFi Huawei E5573* entre los dispositivos conectados sin exponer los servicios internos. Todos los procesos esenciales —incluyendo la mensajería, la base de datos y el monitoreo local— permanecen confinados a la *LAN*, asegurando un entorno cerrado y estable.

La infraestructura de red descrita forma parte de la arquitectura general del sistema mostrada en la Figura 4.1, donde se observa la conexión entre los nodos sensores, la *Raspberry Pi* como punto de acceso y puerta de enlace, el módem *MiFi Huawei E5573* como interfaz de salida y el túnel *Cloudflare* que permite la exposición segura de los servicios hacia Internet. Dicha arquitectura refleja la separación física y lógica entre la red interna de adquisición y la red pública, garantizando que el flujo de datos se mantenga protegido dentro del dominio local y solo se exporte de manera controlada. Para complementar la información de la Figura 4.1 y facilitar la trazabilidad de la implementación, el Cuadro 4.4 resume los parámetros efectivos de configuración de la red, los cuales permiten reproducir el entorno de conexión o diagnosticar posibles comportamientos de conectividad en campo.

Cuadro 4.4. Parámetros efectivos de la capa de red

Elemento	Valor
Interfaz <i>AP</i>	<code>wlan0</code> (modo <i>AP/Master</i>)
<i>SSID</i>	<code>WeatherStation</code>
Banda y canal	2.4 GHz, canal 1 (2412 MHz), ancho 20 MHz
Potencia de transmisión	31 dBm
País regulatorio	CO (Colombia)
Estándar inalámbrico	<i>IEEE 802.11n</i> , <i>Short GI</i> habilitado
Subred <i>LAN</i>	<code>192.168.50.0/24</code> (<i>gateway</i> <code>192.168.50.1</code>)
Compartición	<i>NAT</i> entre <i>LAN</i> (<code>wlan0</code>) y <i>WAN</i> (<code>eth1</code>)
Red <i>WAN</i>	<code>192.168.8.0/24</code> (<i>MiFi Huawei E5573</i>)
Puerta de enlace <i>WAN</i>	<code>192.168.8.1</code>
Seguridad <i>Wi-Fi</i>	<i>WPA2-PSK</i>
Aislamiento de clientes	Desactivado
Exposición externa	Túnel <i>Cloudflare</i> del puerto local 3100

Fuente: Elaboración propia a partir de la configuración del sistema.

4.3.2 Conectividad externa y política de exposición

La conexión a Internet se provee mediante el módem *Huawei E5573*, conectado a la *Raspberry Pi* por la interfaz `eth1` dentro de la red `192.168.8.0/24`, con puerta de enlace `192.168.8.1`. Debido a su batería interna, el módem *MiFi* proporciona tolerancia ante interrupciones eléctricas breves, manteniendo la conectividad *WAN* activa durante posible desconexiones transitorias. Esta arquitectura separa de manera explícita la *LAN* inalámbrica de la *WAN* móvil, permitiendo que incluso ante la pérdida total del enlace a Internet, los sensores continúen publicando datos dentro de la red local y la *Raspberry Pi* siga almacenando y procesando la información sin depender de recursos externos.

Desde la perspectiva de la capa de red, la exposición de servicios hacia Internet se gestiona mediante un túnel *Cloudflare*, que publica de forma segura el puerto local 3100 bajo un dominio público configurado. Este mecanismo actúa como intermediario cifrado entre la red local y los usuarios remotos, evitando la apertura directa de puertos en el *MiFi* y eliminando la dependencia de una dirección *IP* pública. Así, la *API HTTP* desarrollada con *Node.js* y *Express* puede ser accedida de forma segura por las interfaces cliente sin comprometer el aislamiento de la *LAN*. Este esquema simplifica la administración, mejora la seguridad y garantiza la interoperabilidad con las aplicaciones móviles y web descritas en la capa de aplicación.

4.3.3 Seguridad y operación del punto de acceso

La red *Wi-Fi* emplea el protocolo *WPA2-PSK* con clave precompartida, garantizando autenticación robusta y confidencialidad en las comunicaciones inalámbricas. El aislamiento entre clientes se mantiene desactivado para permitir la comunicación directa entre los sensores y los servicios alojados en la *Raspberry Pi*, aunque el tráfico permanece restringido a la subred interna, sin posibilidad de acceso lateral o externo. La configuración de canal fijo (1) y la potencia de transmisión de 31 dBm proporcionan una cobertura estable de aproximadamente 10 m a 15 m en línea de vista, con latencias inferiores a 10 ms, adecuadas para la frecuencia de publicación y sincronización establecida en el sistema.

El servicio del punto de acceso se ejecuta como *daemon* bajo *systemd*, con reinicio automático (*on-failure*) y arranque en el inicio del sistema operativo, asegurando su disponibilidad continua. Los registros del sistema conservan eventos de red, concesiones *DHCP* y reconexiones, lo que permite el diagnóstico de incidencias y la supervisión operativa del enlace inalámbrico. Gracias a esta configuración, la capa de red mantiene un equilibrio entre accesibilidad, seguridad y confiabilidad, garantizando la estabilidad de las comunicaciones tanto en entornos con conectividad plena como en condiciones de operación aislada.

4.4 Capa de aplicación

La capa de aplicación constituye el nivel superior de la arquitectura *IoT*. Su objetivo es proveer la interfaz de interacción con el usuario final y facilitar el acceso a la información mediante servicios web estandarizados. En esta capa se integran tres componentes: la *API*, el servicio web expuesto mediante *HTTP* con interfaz en *HTML* y *JavaScript* (gráficas con *ECharts*), y la aplicación móvil *Weather-Station-APP* desarrollada en *Kotlin* con *Jetpack Compose*.

El servicio *backend* de la *API* está implementado en *Node.js* con el marco *Express*, y se comunica directamente con la base de datos *InfluxDB* mediante la librería oficial de cliente para *Node.js*, distribuida bajo el paquete `@influxdata/influxdb-client`. Esta herramienta provee una interfaz de alto nivel para interactuar con *InfluxDB* desde aplicaciones *JavaScript*, permitiendo ejecutar consultas, escribir puntos de datos y gestionar autenticación mediante tokens definidos en las variables de entorno del sistema. La conexión entre la *API* y *InfluxDB* se establece dentro del mismo entorno local de la *Raspberry Pi*, evitando dependencias externas y reduciendo la latencia en las consultas.

Cada consulta realizada por la *API* traduce los parámetros recibidos en las solicitudes *HTTP* (por ejemplo, rango temporal, tipo de agregación o métrica solicitada) en sentencias *Flux*, el lenguaje nativo de consultas de *InfluxDB*. Las respuestas se procesan, se convierten a formato *JSON* y se devuelven al cliente con los campos normalizados (`timestamp`, `value`, `unit`, `sensor_id`). Este mecanismo garantiza la interoperabilidad entre la capa de procesamiento y las interfaces de

visualización, preservando la precisión temporal y la coherencia semántica de los datos ambientales.

La *API* se expone de forma segura a través de un túnel cifrado gestionado por *Cloudflare*, el cual publica el servicio local sin necesidad de abrir puertos en el módem *Huawei E5573*. Esta estrategia evita la exposición directa de la red interna y garantiza un canal *HTTPS* autenticado mediante certificados gestionados automáticamente. Tanto la interfaz web como la *API* están disponibles públicamente en el dominio `https://api.weatherstation.site/`. Adicionalmente, se aplica una política *CORS* restrictiva que solo permite el acceso desde los dominios autorizados (interfaz web oficial y aplicación móvil), bloqueando solicitudes externas no registradas.

En caso de pérdida temporal de conectividad a Internet, la *Raspberry Pi* mantiene la operación autónoma: continúa registrando y procesando los datos de los sensores en la base local *InfluxDB*, y reanuda la sincronización con la *API* una vez restaurado el enlace. De esta forma, el sistema conserva la integridad de las mediciones y asegura tolerancia ante fallos de red o energía.

En el portal existe una sección de documentación de la *API* accesible desde la interfaz (pestaña “Docs”), donde se presentan las rutas disponibles, parámetros de consulta y ejemplos de respuesta en formato *JSON*.

4.4.1 Convenciones transversales de la API

Formato temporal. Tiempos en *ISO 8601* expresados en *UTC*. Las interfaces web y móvil detectan automáticamente la zona horaria del dispositivo (navegador o teléfono) y convierten las horas a esa referencia local para mostrar los datos de manera coherente.

Unidades y variables. Temperatura en °C, humedad relativa en %, velocidad del viento en m/s e índice F adimensional.

Control de errores. Códigos *HTTP* estandarizados: 200 (éxito), 400 (solicitud inválida), 404 (no encontrado) y 500 (error interno), con mensajes descriptivos y códigos

estables.

Paginación y límites. Respuestas limitables vía `limit` y ordenables con `order`.

4.4.2 Endpoints expuestos y su semántica

Todos los puntos de acceso están bajo `/api` con estilo *REST*.

- **/api/health:** Verifica disponibilidad del servicio y de la base de datos; retorna estado y hora del servidor.
- **/api/latest:** Último valor confirmado de una métrica; parámetros `metric` (`temperature`, `humidity`, `wind_speed`, `F_index`) y `sensor_id`; retorna valor, unidad e `timestamp` en *UTC*.
- **/api/range:** Serie histórica entre `from` y `to` (*ISO 8601*); admite `agg`, `interval`, `order` y `fill`; retorna lista ordenada de puntos.
- **/api/latest_summary:** Resumen compacto para panel principal (web y móvil) con valor, unidad, `sensor_id` e `timestamp` en *UTC*.

Las rutas y parámetros aquí descritos están disponibles también en la sección de documentación de la página web, accesible desde la interfaz (pestaña “*Docs*”).

4.4.3 Opciones de visualización y análisis

Las vistas de ambas plataformas —web y móvil— comparten la misma estructura funcional, diferenciándose únicamente en la disponibilidad de exportación de imágenes (habilitada solo en la versión web). Cada entorno cuenta con dos vistas principales: *Summary* y *Details*, que consumen directamente los datos desde la *API*.

- **Summary:** Consume `/api/latest_summary` para mostrar, en formato de tarjetas o medidores, un resumen consolidado con las cuatro métricas principales

del sistema: temperatura, humedad relativa, velocidad del viento e índice F. Cada indicador aplica franjas de color que representan visualmente el nivel o condición de la variable (por ejemplo, normal, advertencia o riesgo alto) e incluye el sello temporal de la última medición recibida.

- **Details:** Consume `/api/range` para representar series temporales de las variables ambientales. En la versión web se visualizan de forma simultánea las cuatro métricas —cada una en su gráfico independiente—, mientras que en la aplicación móvil el usuario selecciona una métrica específica para analizarla en detalle. En ambos casos, el usuario puede configurar los siguientes parámetros de consulta:

- *Time range* fijo (1m, 5m, 10m, 15m, 1h, 6h, 24h, 7d): define la ventana temporal visible y el intervalo de consulta a la *API*. Todos los cálculos y *KPI* (*Key Performance Indicators*) se restringen a esta ventana activa.
- *Aggregation* (raw, mean, min, max, median, count): función de agregación aplicada sobre bloques de tiempo. **raw** muestra los puntos originales; **mean** calcula el promedio; **min/max** extraen el valor extremo; **median** selecciona la mediana; **count** contabiliza los registros por bloque.
- *Interval* (auto, 10s, 30s, 1m, 5m, 10m, 30m, 1h): determina la granularidad temporal usada en la agregación. **auto** selecciona dinámicamente un tamaño adecuado según el rango.
- *Fill* (manejo de huecos: none o previous): define cómo se representan intervalos sin datos. **none** deja huecos visibles; **previous** repite el último valor conocido para mantener continuidad visual.
- *Live*: activa la actualización periódica automática de la gráfica mientras la vista permanece abierta.
- *Apply*: aplica inmediatamente los cambios de rango, agregación, intervalo y relleno en la consulta y actualización del gráfico.

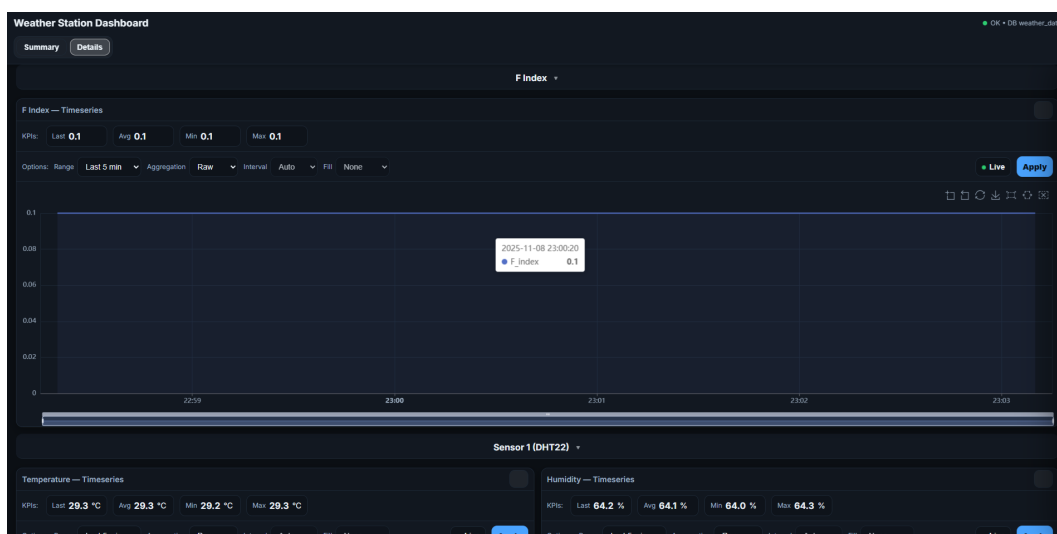
Las gráficas incluyen indicadores clave de desempeño (*KPI*) calculados sobre el rango activo: *Last* (último), *Avg* (promedio), *Min* y *Max*. En la versión web se ofrece además la opción de exportar la visualización a formato *PNG*. En la interfaz web, el resumen consolidado de las cuatro métricas en *Summary* se ilustra en la Figura 4.7, mientras que las gráficas de detalles se ilustran en la Figura 4.8.

Figura 4.7. Página web: vista de la sección *Summary*



Fuente: Elaboración propia.

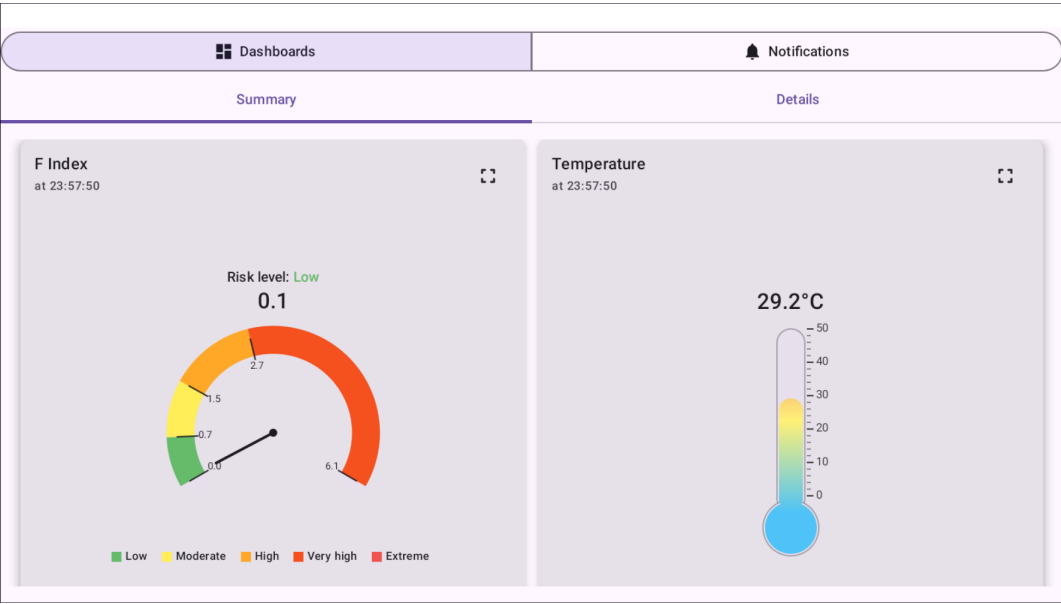
Figura 4.8. Página web: vista de la sección *Details*



Fuente: Elaboración propia.

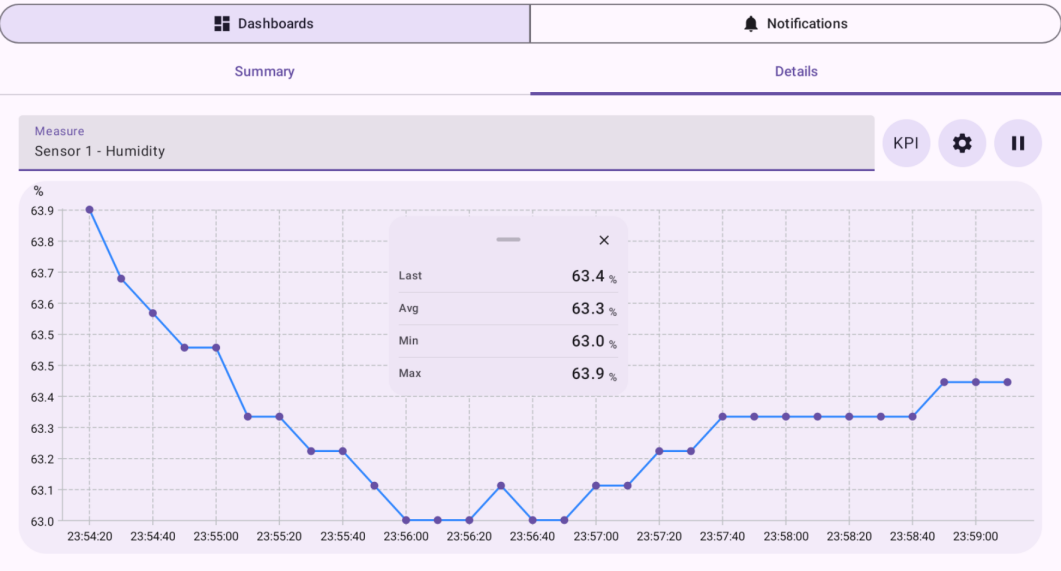
Como complemento a la versión web, la aplicación móvil implementa las mismas pestañas. La Figura 4.9 ilustra la vista *Summary* en Android y la Figura 4.10 muestra la vista *Details*.

Figura 4.9. APP móvil: vista de la sección *Summary*



Fuente: Elaboración propia.

Figura 4.10. APP móvil: vista de la sección *Details*



Fuente: Elaboración propia.

4.4.4 Notificaciones locales en *Android* por umbral

Además de las vistas de consulta, la aplicación móvil implementa un sistema de notificaciones locales basado en umbrales para monitorear continuamente las variables clave. Este sistema se ejecuta mediante un servicio en primer plano (*foreground service*) y conserva su estado tras reinicios del dispositivo.

- **Reglas activas:** Hasta 3 notificaciones simultáneas, cada una asociada a una métrica y a un umbral configurable.
- **Métricas soportadas:** *temperature*, *humidity*, *wind_speed* y *F_index*.
- **Condiciones disponibles:** *mayor que*, *menor que* o *igual que*, seleccionables por el usuario al crear cada regla.
- **Condición de disparo:** La notificación se activa cuando el valor cumple la condición seleccionada y el umbral configurado para la regla.
- **Persistencia de configuración:** Preferencias guardadas con *DataStore Preferences*, lo que garantiza la restauración del estado entre sesiones.
- **Ejecución en primer plano:** Servicio *AlarmForegroundService* con evaluación periódica mediante el componente *AlarmEvaluator*; el ciclo de evaluación se alinea con la frecuencia de sincronización vigente en la capa de procesamiento (10s en pruebas).
- **Estabilidad de alertas:** Histéresis y *cooldown* para prevenir ráfagas de notificaciones y oscilaciones alrededor del umbral.
- **Canales de notificación:** Canales independientes por métrica para gestión clara desde el sistema *Android*.
- **Permisos y continuidad:** Uso de los permisos `POST_NOTIFICATIONS` y `RECEIVE_BOOT_COMPLETED` para mostrar notificaciones y reanudar el servicio tras reinicios.
- **Contenido de la alerta:** Mensaje con la métrica afectada, el valor observado, el umbral configurado, la condición y la hora local del evento.

Figura 4.11. Aplicación móvil *Weather-Station-APP*: sección de notificaciones por umbral con hasta 3 reglas activas, condiciones configurables y estado del servicio en primer plano.

Fuente: Elaboración propia.

En conjunto, estos componentes aseguran la disponibilidad, coherencia y accesibilidad de la información ambiental, permitiendo la supervisión tanto local como remota en tiempo real.

4.4.5 Repositorio del proyecto

El código fuente completo del *backend* de la *API*, la interfaz web y la aplicación móvil se encuentra disponible en el repositorio del proyecto [43]. Este repositorio incluye los archivos de configuración, *scripts* de servicio, documentación técnica y ejemplos de despliegue correspondientes a la capa de aplicación del sistema.

5. Resultados y discusión

Este capítulo presenta los experimentos realizados para validar el funcionamiento integral del sistema bajo dos escenarios ambientales extremos: uno favorable y otro desfavorable para el riesgo de incendio forestal. El objetivo fue analizar la respuesta del sistema ante variaciones controladas de temperatura, humedad relativa y velocidad del viento, y verificar la evolución del índice F calculado a partir de dichas variables.

Los ensayos tuvieron una duración de una hora cada uno y se ejecutaron utilizando el nodo *ESP32* alimentado por *powerbank*, conectado al punto de acceso *WeatherStation* y publicando lecturas cada 10 s hacia el broker *MQTT* de la *Raspberry Pi*, con almacenamiento automático en *InfluxDB*. Las mediciones se visualizaron simultáneamente en el panel de *Grafana*, la interfaz web y la aplicación móvil, garantizando coherencia entre las tres plataformas.

Cada ensayo generó aproximadamente 360 muestras por variable, correspondientes al intervalo de muestreo de 10 s. Estos experimentos no se diseñaron como pruebas de calibración de sensores, sino como una verificación funcional del sistema completo de monitoreo y alerta temprana: desde la adquisición de datos en el nodo *ESP32* hasta la visualización y notificación en las capas de aplicación.

5.1 Escenarios experimentales

Los ensayos se definieron de acuerdo con dos conjuntos de condiciones ambientales extremas, representativos del rango operativo esperado del sistema:

Cuadro 5.1. Escenarios experimentales y condiciones ambientales

Escenario	Condiciones prácticas	Índice F esperado
Escenario 1 — Favorable	Sensor dentro de un refrigerador, con baja temperatura, alta humedad relativa y viento controlado.	Bajo.
Escenario 2 — Desfavorable	Sensor expuesto al sol, con lámpara de calor y gel de sílice para generar alta temperatura, baja humedad y viento controlado.	Alto.

Fuente: Elaboración propia.

5.2 Resultados experimentales

Escenario 1 — Condiciones favorables (bajo riesgo)

Este escenario reprodujo un entorno frío y húmedo, donde se esperaba un índice F bajo y estable. El ensayo se realizó entre las 14:00:00 y las 15:00:00 h.

Procedimiento experimental

El experimento se dividió en dos etapas de 30 minutos, aplicando un mismo esquema de variación del viento basado en cuatro niveles: nivel 0 (apagado), nivel 1 (bajo), nivel 2 (medio) y nivel 3 (alto).

Primera etapa (14:00:00–14:30:00), sensor bajo techo:

- 14:00:00–14:15:00 h: ventilador apagado.
- 14:15:00–14:20:00 h: nivel 1.

- 14:20:00–14:25:00 h: nivel 2.
- 14:25:00–14:30:00 h: nivel 3.

Segunda etapa (14:30:00–15:00:00), sensor ubicado en un recipiente plástico dentro de un refrigerador doméstico, para limitar la condensación directa sobre el dispositivo:

- 14:30:00–14:45:00 h: ventilador apagado.
- 14:45:00–14:50:00 h: nivel 1 y aumento de humedad mediante pulverización de agua.
- 14:50:00–14:55:00 h: nivel 2.
- 14:55:00–15:00:00 h: nivel 3.

Figura 5.1. Montaje del escenario favorable: sensor dentro del refrigerador.



Fuente: Elaboración propia.

Resultados observados

La Figura 5.2 muestra el comportamiento de las variables. Tras el ingreso al refrigerador, la temperatura disminuyó de forma continua mientras la humedad relativa aumentó, especialmente tras la pulverización de agua a las 14:45:00 h. Los escalones de viento generaron variaciones discretas en la lectura del anemómetro.

Como resultado, el índice F descendió y permaneció en valores bajos durante la etapa dentro del refrigerador. Durante la fase previa bajo techo (14:00:00–14:30:00), el incremento temporal de temperatura y viento sí generó algunos picos del índice F que superaron el umbral de alerta, produciendo mensajes SMS. Una vez el sensor ingresó al refrigerador, el índice F cayó y no volvió a activar el sistema de notificaciones.

Figura 5.2. Registro de temperatura, humedad relativa, velocidad del viento e índice F en Grafana durante el escenario favorable.



Fuente: Elaboración propia.

Los valores observados fueron coherentes entre las tres plataformas de visualización, presentando la misma evolución temporal y los mismos valores dentro de la resolución de muestreo de 10 s.

Figura 5.3. Visualización del índice F en la interfaz web durante el escenario favorable.



Fuente: Elaboración propia.

Figura 5.4. Visualización del índice F en la aplicación móvil durante el escenario favorable.



Fuente: Elaboración propia.

Escenario 2 — Condiciones desfavorables (alto riesgo)

Este escenario buscó reproducir condiciones críticas para el índice F: alta temperatura, baja humedad y variaciones del viento. El ensayo se realizó entre las 10:00:00 y las 11:00:00 h.

Procedimiento experimental

Se mantuvo el mismo esquema escalonado de viento utilizado en el escenario favorable.

Primera etapa (10:00:00–10:30:00), sensor bajo techo:

- 10:00:00–10:15:00 h: ventilador apagado.
- 10:15:00–10:20:00 h: nivel 1.
- 10:20:00–10:25:00 h: nivel 2.
- 10:25:00–10:30:00 h: nivel 3.

Segunda etapa (10:30:00–11:00:00), sensor expuesto al sol y a una lámpara de calor, dentro de un recipiente con gel de sílice:

- 10:30:00–10:45:00 h: ventilador apagado.
- 10:45:00–10:50:00 h: nivel 1.
- 10:50:00–10:55:00 h: nivel 2.
- 10:55:00–11:00:00 h: nivel 3.

Figura 5.5. Montaje del escenario desfavorable: sensor expuesto al sol, fuente de calor y gel de sílice.



Fuente: Elaboración propia.

Resultados observados

En la Figura 5.6 se aprecia un aumento pronunciado de temperatura, alcanzando alrededor de 52°C . La humedad descendió hasta valores próximos al 20 %, mientras que el viento registró picos de hasta 11 m/s.

Como consecuencia, el índice F aumentó progresivamente, superó el umbral de alerta configurado y se mantuvo elevado mientras persistieron las condiciones extremas.

Figura 5.6. Registro de temperatura, humedad relativa, velocidad del viento e índice F en Grafana durante el escenario desfavorable.



Fuente: Elaboración propia.

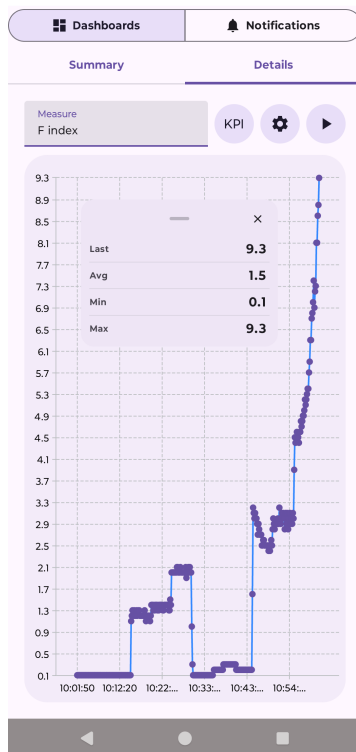
Las plataformas de visualización web y móvil mostraron la misma evolución temporal del índice F que el panel de *Grafana*, reproduciendo los picos, valles y cruces del umbral de alerta dentro de la resolución de muestreo de 10 s.

Figura 5.7. Visualización del índice F en la interfaz web durante el escenario desfavorable.



Fuente: Elaboración propia.

Figura 5.8. Visualización del índice F en la aplicación móvil durante el escenario desfavorable.



Fuente: Elaboración propia.

Activación de alertas SMS

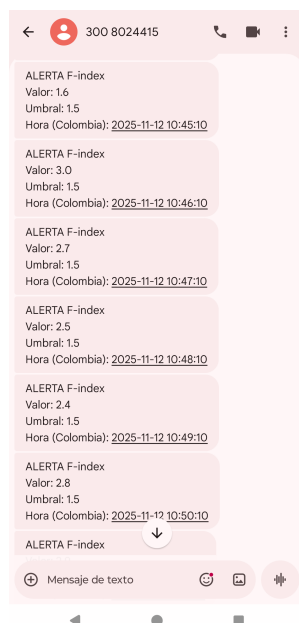
El sistema de alertas reaccionó correctamente: las notificaciones se generaron cuando el índice F superó el umbral de $F = 1,5$, primero entre 10:24:50 y 10:29:50 h y posteriormente entre 10:45:10 y 11:00:00 h.

Figura 5.9. Mensajes SMS recibidos entre 10:24:50–10:29:50 h.



Fuente: Elaboración propia.

Figura 5.10. Mensajes SMS recibidos entre 10:45:10–11:00:10 h.



Fuente: Elaboración propia.

5.3 Resumen del comportamiento del índice F

El Cuadro 5.2 resume, de forma comparativa, cómo respondió el índice F ante los cambios de temperatura, humedad relativa y velocidad del viento en los dos escenarios evaluados.

Cuadro 5.2. Relación entre las condiciones ambientales y el índice F

Condición ambiental	Efecto teórico sobre el índice F	Comportamiento observado en los experimentos
Temperatura del entorno	Temperaturas más altas elevan el índice F.	En condiciones frías del refrigerador (escenario favorable), el índice F disminuyó y se mantuvo en niveles bajos. En condiciones de calor intenso (escenario desfavorable), el índice F aumentó de manera continua.
Humedad relativa del aire	Humedades relativas más altas reducen el índice F.	En el escenario favorable, el aumento de humedad —especialmente tras el rociado de agua— produjo una caída marcada del índice F. En el escenario desfavorable, la humedad descendió hasta valores bajos y el índice F se incrementó de forma sostenida.
Velocidad del viento	Velocidades de viento más altas incrementan el índice F.	Los escalones de velocidad del ventilador generaron aumentos puntuales del índice F en ambos escenarios. Estos incrementos fueron más notorios cuando coincidían con altas temperaturas y baja humedad en el escenario desfavorable.

Fuente: Elaboración propia.

6. Conclusiones

El sistema IoT desarrollado demostró ser una solución funcional, modular y de bajo costo para el monitoreo local de variables ambientales asociadas al riesgo de incendio forestal. La arquitectura implementada, basada en las recomendaciones *ITU-T Y.2060* y *ITU-T Y.4113*, permitió integrar de manera coherente los niveles de dispositivo, red, procesamiento y aplicación, garantizando un flujo de datos continuo y confiable desde los sensores hasta las plataformas de visualización del usuario final.

Los ensayos experimentales confirmaron la estabilidad operativa del nodo *ESP32* bajo diferentes condiciones ambientales, así como la robustez del enlace inalámbrico establecido con el punto de acceso *WeatherStation*. La sincronización mediante *MQTT* y el manejo centralizado de marcas de tiempo aseguraron coherencia temporal entre las mediciones, lo cual es esencial para la estimación correcta del índice F y para el análisis comparativo entre escenarios.

La integración con la *Raspberry Pi* como pasarela y unidad de procesamiento permitió almacenar las mediciones en *InfluxDB*, procesarlas en tiempo real y presentarlas simultáneamente en tres plataformas: el panel de *Grafana*, la interfaz web y la aplicación móvil. La consistencia entre estas visualizaciones demuestra la correcta implementación del *backend* y la estabilidad del túnel de acceso remoto, lo que habilita la supervisión del sistema desde ubicaciones externas sin comprometer la seguridad de la red local.

La evaluación del índice F bajo escenarios ambientales contrastantes confirmó su utilidad como métrica operativa de riesgo inmediato. En el escenario favorable, caracterizado por temperaturas bajas y humedades elevadas, el índice permaneció estable y en niveles reducidos. Por el contrario, en el escenario desfavorable —con temperaturas cercanas a 52°C, humedades relativas alrededor del 20 % y velocidades de viento que alcanzaron 11 m/s— el índice experimentó incrementos sostenidos que activaron las alertas configuradas. Este comportamiento, alineado con la teoría, valida el modelo computacional empleado para la estimación del índice F.

El sistema de notificaciones por *SMS* mostró un funcionamiento preciso y oportuno. Las alertas se generaron únicamente cuando el índice F superó el umbral de $F = 1,5$, evitando falsas alarmas y demostrando un desempeño adecuado en situaciones donde la conectividad a Internet puede ser limitada. Esta característica hace que el sistema sea especialmente útil en zonas rurales o de difícil acceso, donde las comunicaciones móviles son el principal medio disponible de conectividad.

A nivel de diseño, la modularidad del sistema permite incorporar sensores adicionales, modificar los intervalos de muestreo o integrar nuevos modelos de riesgo sin alterar la estructura general. Esto garantiza escalabilidad y facilita futuras ampliaciones o adaptaciones a diferentes contextos ambientales. Asimismo, el uso de tecnologías ampliamente soportadas —como *MQTT*, *Node.js*, *InfluxDB* y *Grafana*— favorece la interoperabilidad y prolonga la vida útil del sistema.

Si bien el sistema incorpora mecanismos orientados a la recuperación ante fallos de conectividad o alimentación —como la reconexión de servicios, la operación autónoma de la red local y la persistencia de datos—, la validación experimental de estos comportamientos bajo escenarios controlados de corte de energía y procesos de reconexión no formó parte del alcance del presente trabajo. No obstante, dichos aspectos fueron considerados a nivel de diseño e implementación con el objetivo de mejorar la robustez del sistema en condiciones reales de operación.

En conjunto, los resultados obtenidos permiten concluir que el sistema cumple con los objetivos planteados: adquiere, transmite y procesa datos ambientales de forma coherente; mantiene sincronización temporal centralizada; presenta la información de manera clara y accesible en múltiples plataformas; y genera alertas confiables en condiciones de riesgo. Todo lo anterior posiciona esta solución como una herramienta viable para el fortalecimiento de sistemas de vigilancia y alerta temprana de incendios forestales a escala local, especialmente en comunidades que requieren alternativas tecnológicas de bajo costo, mantenimiento sencillo e instalación flexible.

7. Recomendación para trabajos futuros

El sistema desarrollado constituye una base funcional para la medición ambiental, el cálculo del índice F y la emisión de alertas tempranas. No obstante, existen oportunidades de mejora orientadas a aumentar su alcance, robustez y aplicabilidad en escenarios reales. A continuación se presentan las principales recomendaciones técnicas para futuras iteraciones.

- Integrar sensores adicionales para variables como radiación solar, humedad del material vegetal o temperatura del suelo, con el fin de fortalecer la estimación del riesgo.
- Incorporar mecanismos de calibración automatizada o comparaciones cruzadas entre nodos para reducir la deriva de los sensores en ambientes extremos.
- Evaluar topologías de red redundantes, como enlaces de malla o tecnologías de baja potencia tipo LoRaWAN, para aumentar la resiliencia frente a fallos de conectividad.
- Evaluar experimentalmente el comportamiento del sistema ante fallos de alimentación y conectividad, analizando los tiempos de recuperación, la continuidad del registro de datos y los procesos de reconexión de los nodos sensores y del *gateway*.
- Ampliar la plataforma analítica mediante algoritmos de aprendizaje automático que permitan identificar tendencias y anticipar condiciones críticas del índice F.
- Extender la aplicación móvil con funciones de almacenamiento temporal local, notificaciones avanzadas y acceso a series históricas, garantizando su utilidad aun sin conexión estable.
- Desarrollar módulos de alimentación autónoma basados en energía solar y baterías de larga duración, orientados a despliegues remotos.

- Integrar el sistema en plataformas colaborativas de gestión de emergencias para facilitar la recepción de alertas y el acceso al estado de los sensores por parte de organismos ambientales.

Estas recomendaciones permiten proyectar el sistema hacia aplicaciones de monitoreo continuo en campo, mejorando su escalabilidad y utilidad operativa.

Referencias

- [1] Food and Agriculture Organization of the United Nations, *The State of the World's Forests 2024 – Forest-sector innovations towards a more sustainable future*. Rome: FAO, 2024.
- [2] M. Silveira, C. Silva-Junior, L. Anderson, and L. A. ao, “Amazon fires in the 21st century: The year of 2020 in evidence,” 2022.
- [3] Unidad Nacional para la Gestión del Riesgo de Desastres (UNGRD), “Ungrd informa sobre la situación de los incendios forestales en el país,” 2024.
- [4] N. Celis, A. Casallas, E. A. López-Barrera, M. Felician, M. D. Marchi, and S. Pappalardo, “Climate change, forest fires, and territorial dynamics in the amazon rainforest: An integrated analysis for mitigation strategies,” 2023.
- [5] R. Barriá, “Wildfires as a public health problem: a setting for nursing in disasters,” 2019.
- [6] J. Herrera, “El impacto de los incendios forestales y el fenómeno de el niño en la economía del país,” *Portafolio*, Enero 2014.
- [7] D. Pascual and M. Quiroga, “Colombia, un país en llamas: la ineficiencia en la lucha contra los incendios forestales es la constante,” *Periódico UNAL*, Enero 2024.
- [8] A. Hasson, G. A. Mills, B. Timball, and K. Walsh, “Assessing the impact of climate change on extreme fire weather in southeast australia. cawcr technical report no. 007,” tech. rep., Diciembre 2008. Disponible en: http://www.cawcr.gov.au/technical-reports/CTR_007.pdf, Último acceso: 17 Septiembre 2024.
- [9] Y. Liu, S. L. Goodrick, and J. A. Stanturf, “Future u.s. wildfire potential trends projected using a dynamically downscaled climate change scenario,” 2013.
- [10] B. J. Stocks, B. D. Lawson, M. E. Alexander, C. E. Vanwagner, R. S. McAlpine, T. J. Lynham, and D. E. Dube, “The canadian forest fire danger rating system: An overview,” 1989.

- [11] L. S. Bradshaw, J. E. Deeming, R. E. Burgan, and J. D. Cohen, “The 1978 national fire-danger rating system: technical documentation,” tech. rep., USDA Forest Service, 1984.
- [12] A. G. McArthur, *Fire behaviour in eucalypt forests*. Canberra: Australia. Forestry and Timber Bureau, 1967.
- [13] J. Sharples, R. McRae, R. Weber, and A. Gill, “A simple index for assessing fire danger rating,” 2009.
- [14] M. Banzi and M. Shiloh, *Getting Started with Arduino*. 2015.
- [15] H. Dipak, L. Solanki, and G. Sahu, “A review paper on raspberry pi and its applications,” 2020.
- [16] L. Li, A. Sali, N. K. Noordin, A. Ismail, and F. Hashim, “Prediction of peatlands forest fires in malaysia using machine learning,” 2023.
- [17] O. Satir, S. Berberoglu, and A. Cilek, “Modelling long term forest fire risk using fire weather index under climate change in turkey,” 2016.
- [18] A. Tsipis, A. Papamichail, I. Angelis, G. Koufoudakis, G. Tsoumanis, and K. Oikonomou, “An alertness-adjustable cloud/fog iot solution for timely environmental monitoring based on wildfire risk forecasting,” 2020.
- [19] C. Chandler, P. Cheney, P. Thomas, L. Trabaud, and D. Williams, *Fire in forestry. In Forest Fire Management and Organization*. 1983.
- [20] A. Sahour, F. Boumehrez, F. Maamri, H. Djellab, and B. Abdelaali, “Forest fire risk monitoring using fuzzy logic and iot technology,” 2024.
- [21] M. Yusro, N. Arif, and R., “Utilization of microcontroller technology using arduino board for internet of things (a systematic review),” 2021.
- [22] Á. B. Herranz, *Desarrollo de aplicaciones para IoT con el módulo ESP32*. 2019.
- [23] A. Ukil, H. Braendle, and P. Krippner, “Distributed temperature sensing: Review of technology and applications,” 2011.
- [24] M. Mansoor, I. Haneef, S. Akhtar, A. D. Luca, and F. Udrea, “Silicon diode temperature sensors – a review of applications,” 2015.

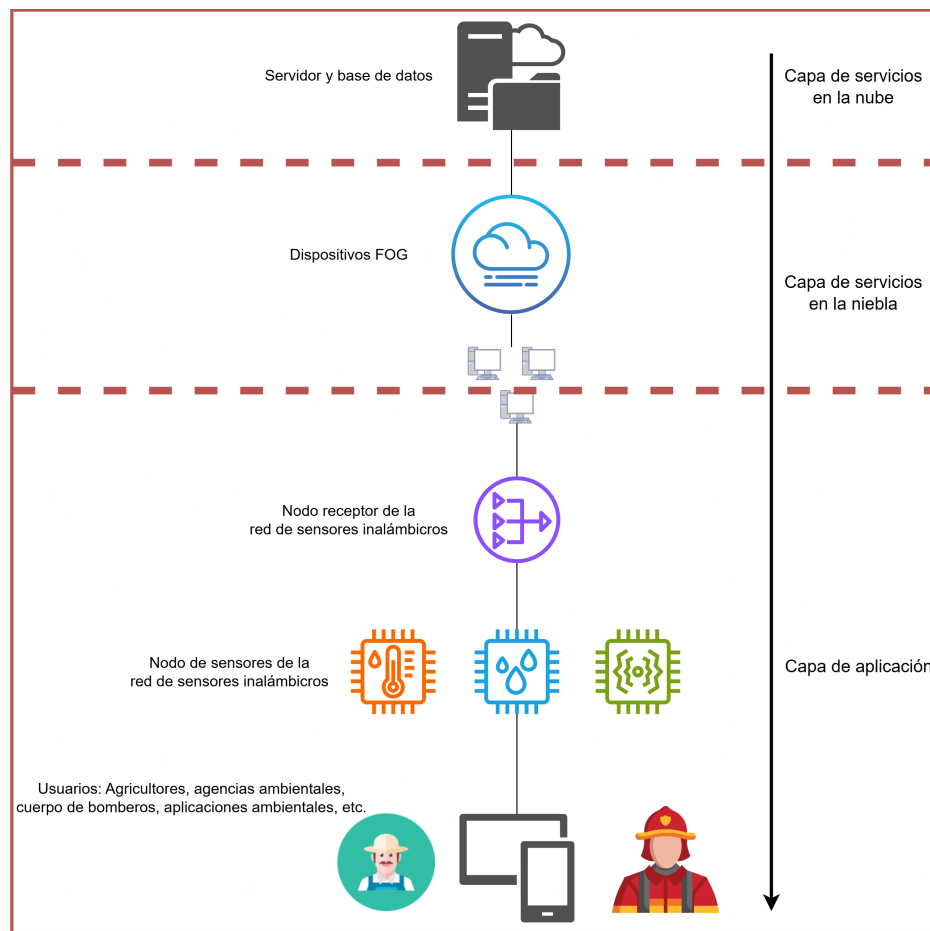
- [25] B. A. Kuzubasoglu, “Recent studies on the humidity sensor: A mini review,” 2022.
- [26] S. Pindado, J. Cubas, and F. Sorribes, “The cup anemometer, a fundamental meteorological instrument for the wind energy industry. research at the idr/upm institute,” 2014.
- [27] U. G. Acer, A. Boran, C. Forlivesi, W. Liekens, F. Pérez-Cruz, and F. Kawsar, “Sensing wifi network for personal iot analytics,” 2015.
- [28] V.-D. Gavra and O. A. Pop, “Usage of zigbee and lora wireless technologies in iot systems,” 2020.
- [29] Telecommunication Standardization Sector of ITU, “Y.2060 overview of the internet of things,” tech. rep., 2012.
- [30] Telecommunication Standardization Sector of ITU, “Y.4113 requirements of the network for the internet of things,” tech. rep., 2016.
- [31] N. Naik, “Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http,” *2017 IEEE International Systems Engineering Symposium (ISSE)*, pp. 1–7, 2017.
- [32] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, “Mqtt-s—a publish/subscribe protocol for wireless sensor networks,” in *Proceedings of the 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE 2008)*, pp. 791–798, IEEE, 2008.
- [33] OASIS Open, “Mqtt version 5.0,” tech. rep., OASIS Standard, Mar. 2019. OASIS Standard, March 2019.
- [34] Z. Shelby, K. Hartke, and C. Bormann, “The constrained application protocol (coap),” *RFC 7252*, 2014. IETF Request for Comments 7252.
- [35] R. T. Fielding, “Architectural styles and the design of network-based software architectures,” 2000. Doctoral Dissertation, defining the REST architectural style.
- [36] D. Bandyopadhyay and J. Sen, “Internet of things: Applications and challenges in technology and standardization,” *Wireless Personal Communications*, vol. 58, no. 1, pp. 49–69, 2011.

- [37] F. Palma, T. Olsson, A. Wingkvist, and J. González-Huerta, “Assessing the linguistic quality of rest apis for iot applications,” *arXiv preprint arXiv:2205.06533*, 2022.
- [38] E. Mabothe, “Exploring dynamic restful api implementation in iot,” *Scientific Reports*, 2025.
- [39] T. A. Chen *et al.*, “Internet of things: Development intelligent programmable system for real-time environmental monitoring,” *Sensors*, vol. 22, no. 14, p. 5138, 2022.
- [40] G. Jošt and V. Taneski, “State-of-the-art cross-platform mobile application development frameworks: A comparative study of market and developer trends,” *Informatics*, vol. 12, no. 2, p. 45, 2025.
- [41] DFRobot, “Rs485 wind speed transmitter (sen0483).” https://wiki.dfrobot.com/RS485_Wind_Speed_Transmitter_SKU_SEN0483, 2023.
- [42] ESPHome Contributors, “Esphome: Smart home automation framework for esp8266/esp32.” <https://esphome.io/>, 2024. Nabu Casa, Inc.
- [43] A. R. Maldonado, “Sistema iot de alerta temprana de riesgo de incendios forestales basado en la norma itu-t y.4113,” 2025. Repositorio GitHub del trabajo de grado. Todos los códigos utilizados están disponibles de manera pública en este repositorio.

Apéndice A

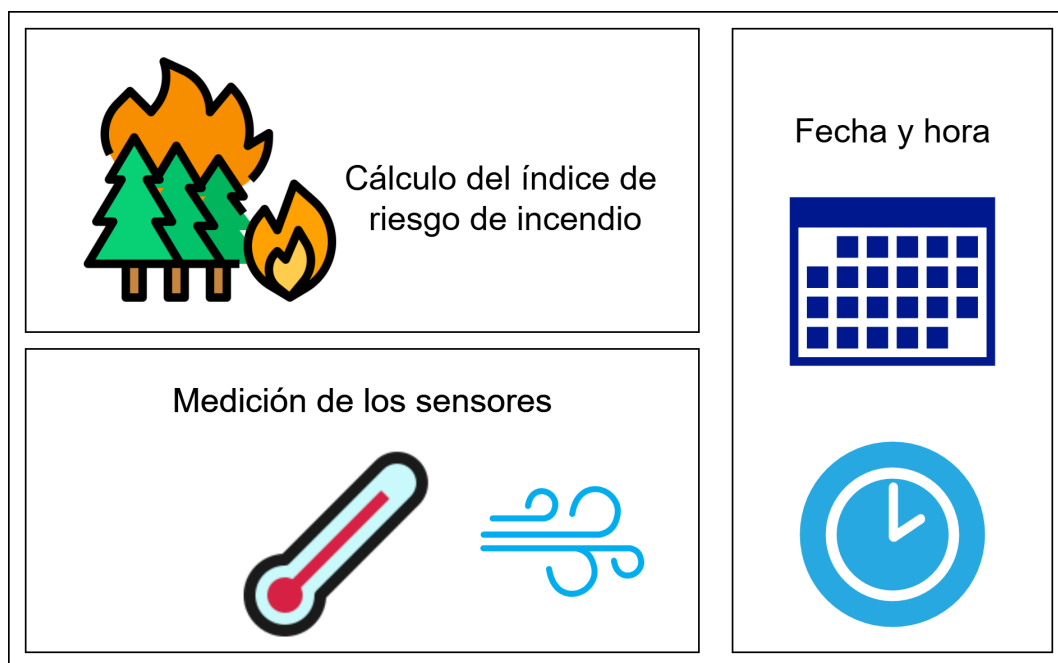
Figuras

Figura A.1. Arquitectura *IoT* propuesta en An Alertness-Adjustable Cloud/Fog IoT Solution for Timely Environmental Monitoring Based on Wildfire Risk Forecasting.



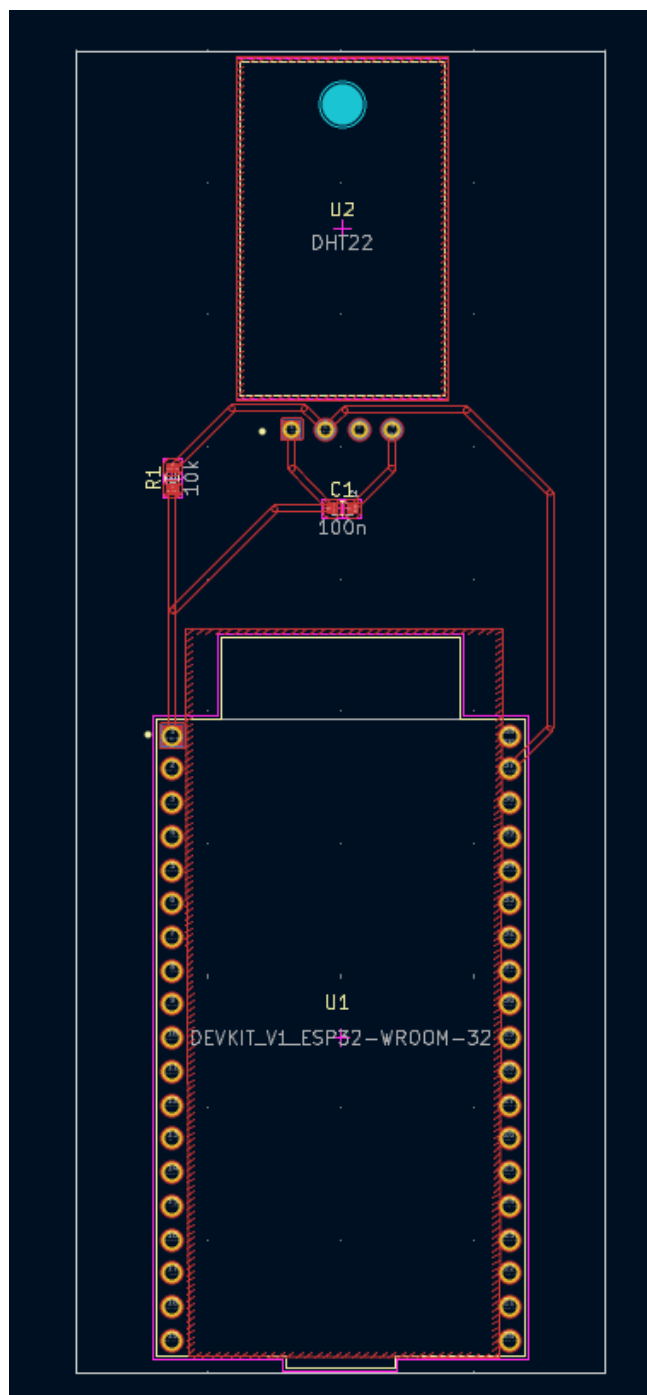
Fuente: Elaboración propia.

Figura A.2. Esquema general de interfaces de usuario para proyectos de estas características.



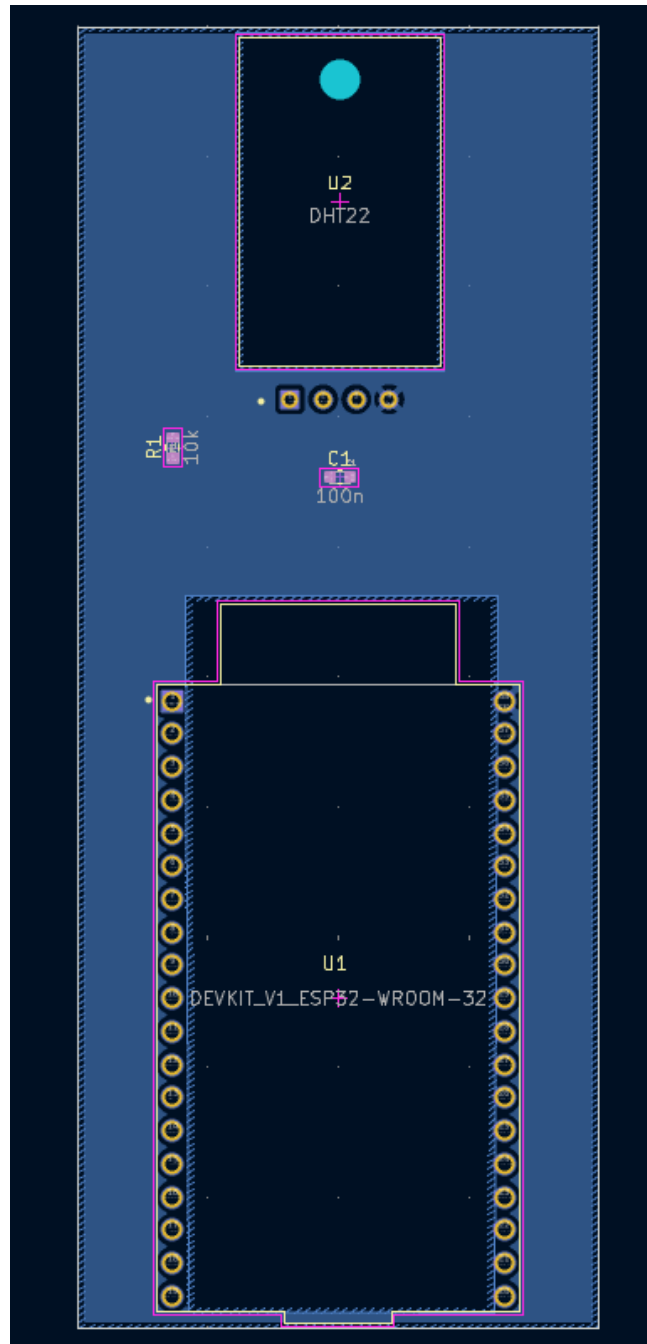
Fuente: Elaboración propia.

Figura A.3. Diseño de PCB, vista superior (capa F.Cu).



Fuente: Elaboración propia.

Figura A.4. Diseño de PCB, vista inferior (capa B.Cu).



Fuente: Elaboración propia.