

# **Sistema Inteligente de Detección de Incendio y Monitoreo de Calidad del Aire con ESP32**

**S.A.P.O**

(Sistema de Alerta de Partículas y Oxígeno)

**Versión:** 1.0

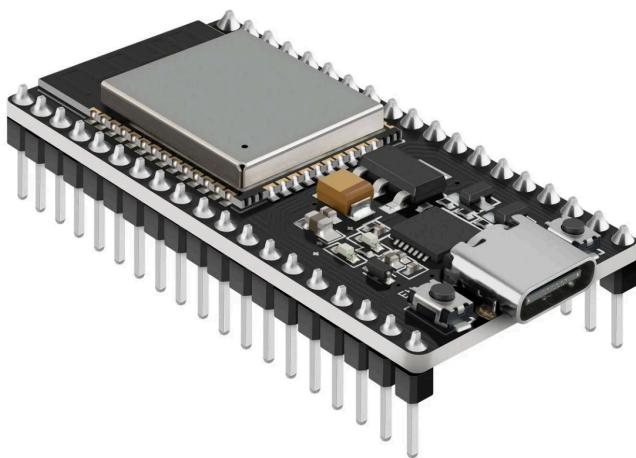
**Materia:** Sistemas Embebidos

**Docente:** Jenny Grizel Cerecero Luna

**Fecha:** Martes 17 de Febrero 2026

**Equipo:**

- Gustavo Ugalde Rios
- Gael Olguín Rangel



## ÍNDICE

<b>Definiciones, Acrónimos y Abreviaturas.</b>	<b>4</b>
<b>Propósito.</b>	<b>5</b>
<b>Resumen del documento.</b>	<b>6</b>
<b>1. Descripción del problema</b>	<b>7</b>
<b>2. Importancia del problema</b>	<b>7</b>
<b>3. Justificación del proyecto</b>	<b>8</b>
<b>4. Propuesta del proyecto</b>	<b>9</b>
<b>5. Objetivo del proyecto</b>	<b>9</b>
6. Factibilidad del Proyecto	9
Factibilidad Técnica	9
Factibilidad Económica	10
Factibilidad Operativa	10
<b>7. Justificación de la propuesta tecnológica</b>	<b>10</b>
<b>8. Funcionamiento general del sistema</b>	<b>11</b>
8.1 Perspectiva del Producto.	11
8.2 Características de los Usuarios	11
8.3 Suposiciones y Dependencias	11
<b>9. Alcance del proyecto</b>	<b>12</b>
<b>10. Metodología de desarrollo</b>	<b>12</b>
<b>11. Plan de trabajo por semanas</b>	<b>13</b>
11.1 Sprint 1	13
11.2 Sprint 2	13
<b>12. Equipo de trabajo</b>	<b>14</b>
12.1 Distribución de roles	14
<b>13. Herramientas de trabajo</b>	<b>14</b>
13.1 Trello: Enlace al tablero del equipo.	15
13.2 GitHub del equipo: Enlace al repositorio del proyecto.	15
13.3 GitHub Embedded Systems Project : Enlace al repositorio del proyecto.	15
<b>14. Requerimientos del sistema</b>	<b>16</b>
14.1 Requerimientos funcionales	16
14.2 Requerimientos funcionales	16
14.3 Requerimientos no funcionales	16
<b>14.4 Interfaces de Hardware</b>	<b>17</b>
14.4.1 Sensores	17
14.4.2 Actuadores	17
14.4.3 Buses y Comunicación Interna	17
<b>14.5 Interfaces de Software</b>	<b>18</b>
14.5.1 Interfaces de Comunicación	18
14.5.2 Interfaces Externas	18
14.6.1 Restricciones de Hardware	19
14.6.2 Restricciones Regulatorias	19

14.6.3 Restricciones Económicas	20
14.6.4 Restricciones Energéticas	20
14.6.5 Restricciones de Tiempo	20
<b>15. Componentes del sistema.</b>	<b>21</b>
<b>16. Protocolos y consideraciones técnicas</b>	<b>22</b>
16.1 Seguridad del sistema (Lógica y Software)	22
16.2 Seguridad física y eléctrica	22
16.3 Comunicación y red	23
16.4 Gestión de voltajes y compatibilidad eléctrica	23
<b>17. Flujo del prototipo</b>	<b>24</b>
<b>Referencias bibliográficas:</b>	<b>26</b>

## Definiciones, Acrónimos y Abreviaturas.

### 1. **ADC (Analog-to-Digital Converter):**

El ADC (acrónimo de convertidor analógico digital) es un dispositivo electrónico que convierte una señal analógica en una señal digital.

### 2. **ESP32:**

Es un microcontrolador SoC de bajo costo y bajo consumo, fabricado por Espressif, que integra procesador dual-core, Wi-Fi, Bluetooth y periféricos para conectar sensores y actuadores.

### 3. **MQ-2:**

Sensor semiconductor diseñado para detectar gases inflamables (como propano, metano, gas natural) y humo, mediante cambios de conductividad en un material de SnO<sub>2</sub>.

### 4. **MQ-135:**

Sensor de gas semiconductor que mide la calidad del aire y detecta gases nocivos como amoníaco (NH<sub>3</sub>), dióxido de carbono (CO<sub>2</sub>), metano (CH<sub>4</sub>) y compuestos orgánicos volátiles (VOCs).

### 5. **LittleFS:**

Sistema de archivos ligero y seguro ante fallos de alimentación, diseñado para sistemas embebidos con memoria flash limitada, que incluye balanceo de desgaste y uso acotado de RAM/ROM.

### 6. **HTTP (Hypertext Transfer Protocol):**

Protocolo de comunicación de aplicaciones cliente-servidor que define cómo se intercambian recursos (como páginas web) en la red mediante mensajes de texto en un modelo de solicitud-respuesta.

### 7. **JSON (JavaScript Object Notation):**

Formato de intercambio de datos abierto, de texto legible, que representa objetos mediante pares nombre-valor y arrays, usado ampliamente en aplicaciones web y servicios REST.

### 8. **Standalone:**

Sistema o dispositivo que puede operar de forma independiente, sin depender de otros dispositivos o redes, integrando todos los componentes necesarios en una sola unidad.

### 9. **ISO (International Organization for Standardization)**

Organización internacional no gubernamental que reúne a organismos de normalización de más de 160 países y desarrolla y promueve estándares internacionales para tecnología y procesos.

## Propósito.

El presente documento tiene como objetivo establecer de manera formal los requisitos funcionales, no funcionales y técnicos para el desarrollo del sistema S.A.P.O. (Sistema de Alerta de Partículas y Oxígeno). Este documento sirve como guía para la implementación técnica y como base para la validación del sistema, asegurando que el monitoreo de gas y humo mediante el ESP32 cumpla con los estándares de seguridad previstos.

## Resumen del documento.

El documento se estructura en tres bloques principales: la sección de introducción donde se definió el contexto, la descripción general detalla el funcionamiento de nuestra alarma y los perfiles de usuario, finalmente tenemos la sección de requisitos específicos donde se describieron las interfaces de software y las restricciones de tiempo real para la activación de alarmas.

En el primer apartado, la **introducción** establece el inicio del proyecto S.A.P.O., identificando la problemática de seguridad que justifica su creación. Aquí se delimita el propósito de este documento como el estándar técnico de referencia para el desarrollo, además de unificar los términos clave mediante un glosario especializado. Se incluyen referencias a los manuales técnicos de los sensores MQ-2 y MQ-135, lo que garantiza que la información no sea solo descriptiva, sino que esté respaldada por marcos de calidad y seguridad de la información. Este bloque asegura que cualquier lector comprenda la importancia del sistema.

Posteriormente, la **descripción general** profundiza en la arquitectura del sistema, destacando su naturaleza como dispositivo autónomo (*standalone*). Se explica cómo el ESP32 gestiona el monitoreo de manera local, garantizando la privacidad y la rapidez de respuesta. En esta parte, se definen los perfiles de usuario, enfocándose en familias y dueños de pequeños negocios que requieren una solución de seguridad intuitiva. También se detallan las dependencias del entorno, como la necesidad de una alimentación eléctrica constante de 5V y la conectividad inalámbrica local, estableciendo las condiciones operativas necesarias para que la alarma funcione correctamente en el día a día.

Finalmente, la sección de **requisitos específicos** detalla la implementación técnica que hace posible el funcionamiento del sistema. Se describen las interfaces de software, explicando cómo el ESP32 actúa como servidor para enviar datos en tiempo real mediante el protocolo HTTP y objetos JSON, permitiendo una interfaz web fluida y ligera. Un punto crítico en este cierre son las restricciones de tiempo real, donde se establece que la activación del buzzer y los LEDs de alerta no debe superar el segundo de latencia tras la detección de una anomalía. Este último bloque asegura que el documento no sea solo una guía teórica, sino una especificación técnica diseñada para salvar vidas mediante la eficiencia y la inmediatez tecnológica.

## 1. Descripción del problema

En espacios cerrados como hogares, oficinas o salones de clase, la detección oportuna de humo o de una mala calidad del aire no siempre es inmediata. Muchas veces los incendios comienzan de forma silenciosa y cuando se perciben visualmente ya han avanzado considerablemente. Del mismo modo, la acumulación de gases o contaminantes en el ambiente puede afectar la salud sin que las personas lo noten.

Los detectores tradicionales de humo únicamente activan una alarma sonora, pero no permiten monitorear valores en tiempo real, ni consultar datos históricos, ni gestionar la alarma desde un dispositivo móvil o computadora. Esto limita la supervisión y el análisis del comportamiento ambiental en el tiempo.



## 2. Importancia del problema

La detección temprana de humo puede prevenir daños materiales y reducir riesgos para la vida humana. Actuar en los primeros segundos de un posible incendio puede marcar la diferencia entre un incidente controlable y una emergencia mayor.

De acuerdo con la National Fire Protection Association (NFPA), los departamentos de bomberos en Estados Unidos atienden en promedio más de 350,000 incendios estructurales al año, muchos de ellos en viviendas. La misma organización señala que el riesgo de morir en un incendio doméstico se reduce aproximadamente en un 55% cuando existen detectores de humo funcionales, lo que demuestra la relevancia de contar con sistemas de detección temprana.

Por otro lado, la Organización Mundial de la Salud (OMS) señala que la contaminación del aire es uno de los principales riesgos ambientales para la salud. Según sus reportes, alrededor del 99% de la población mundial respira aire que supera los límites recomendados de calidad, y se asocia a aproximadamente 7 millones de muertes prematuras cada año a nivel global. Esto evidencia la importancia de monitorear la calidad del aire incluso en espacios interiores.

Tener acceso a información en tiempo real permite tomar decisiones oportunas, como ventilar el área o revisar posibles fuentes de riesgo. Además, contar con un sistema que registre eventos y permita visualizar datos históricos contribuye a un análisis más completo del entorno.

### 3. Justificación del proyecto

Este proyecto se desarrolla como una solución tecnológica accesible y de bajo costo que integra sensores ambientales con conectividad WiFi. Utilizando una ESP32, se propone construir un sistema capaz de detectar humo, monitorear la calidad del aire y mostrar la información a través de una página web.

La principal justificación radica en que el sistema no solo alerta ante una posible emergencia, sino que también proporciona herramientas de monitoreo, análisis y supervisión remota. Permite conocer cuántas veces se ha activado la alarma, observar el comportamiento del ambiente durante una semana y tomar decisiones basadas en datos históricos.

Un valor agregado importante del proyecto es la facilidad de acceso a la información mediante una interfaz web. Al funcionar como servidor web local, el sistema puede consultarse desde cualquier dispositivo conectado a la misma red WiFi, como un teléfono, tableta o computadora, sin necesidad de instalar aplicaciones adicionales. Esto mejora la accesibilidad, la usabilidad y la interacción con el sistema.

De esta manera, el proyecto combina prevención, monitoreo, análisis y accesibilidad tecnológica en una sola plataforma.



## 4. Propuesta del proyecto

La propuesta consiste en diseñar e implementar un sistema embebido basado en la ESP32 que funcione como servidor web local. El sistema tendrá la capacidad de:

- Detectar presencia de humo mediante el sensor MQ-2.
- Monitorear la calidad del aire mediante el sensor MQ-135.
- Activar una alarma sonora (buzzer) y una alerta visual (LED) cuando se detecte una condición de riesgo.
- Permitir que el usuario apague la alarma desde la página web.
- Mostrar en tiempo real los valores actuales de los sensores.
- Llevar un conteo de cuántas veces se ha activado la alarma.
- Guardar un historial de datos durante siete días.
- Mostrar gráficas del comportamiento ambiental en la página web.

El acceso al sistema se realizará desde cualquier dispositivo conectado a la misma red WiFi, ingresando a la dirección IP del ESP32.

## 5. Objetivo del proyecto

El objetivo del proyecto S.A.P.O (Sistema de Alerta de Partículas y Oxígeno) es diseñar e implementar un sistema embebido capaz de detectar la presencia de humo y monitorear la calidad del aire en tiempo real, integrando sensores ambientales con una ESP32 y una interfaz web local.

El sistema busca:

- Detectar condiciones de posible incendio.
- Emitir alertas visuales y sonoras.
- Permitir desactivación física y remota.
- Registrar histórico de datos.
- Mostrar gráficas para análisis.

## 6. Factibilidad del Proyecto

### Factibilidad Técnica

El proyecto es técnicamente viable ya que utiliza componentes comerciales disponibles en el mercado (ESP32, MQ-2, MQ-135). La ESP32 cuenta con capacidades suficientes de procesamiento, memoria y conectividad WiFi para soportar el sistema propuesto.

## Factibilidad Económica

El costo total estimado del sistema es bajo, ya que los componentes principales son económicos y de fácil acceso. Esto lo convierte en una solución accesible para aplicaciones académicas y domésticas.

## Factibilidad Operativa

El sistema funciona de manera autónoma en red local, sin requerir infraestructura adicional ni conexión a internet, lo que facilita su implementación y uso.

## 7. Justificación de la propuesta tecnológica

La elección de la ESP32 se debe a que integra conectividad WiFi, múltiples entradas analógicas (ADC) y capacidad de procesamiento para manejar sensores y un servidor web al mismo tiempo.

Los sensores seleccionados cumplen funciones específicas:

Componente	Función principal
MQ-2	Detectar humo y gases inflamables (posible incendio)
MQ-135	Medir calidad del aire de forma relativa
Buzzer	Emitir alarma sonora
LED	Indicar visualmente el estado de alerta

Esta combinación permite construir un sistema funcional sin necesidad de hardware complejo adicional.

El almacenamiento del histórico se realizará utilizando la memoria interna del ESP32 (LittleFS), lo que evita la necesidad de una tarjeta SD externa y simplifica el diseño.

## 8. Funcionamiento general del sistema

El sistema operará de manera continua realizando lecturas periódicas de los sensores. Cuando el valor del MQ-2 supere un umbral definido, se considerará una posible condición de incendio. En ese momento:

- Se activará el buzzer.
- Se encenderá el LED rojo.
- Se incrementará el contador de activaciones.

La página web mostrará el estado actual del sistema (Normal o Incendio), los valores de los sensores en tiempo real, el número de activaciones registradas y las gráficas correspondientes al histórico.

El usuario podrá presionar un botón en la página para apagar la alarma sonora una vez que haya verificado la situación.

Además, el sistema almacenará periódicamente los datos de los sensores para mantener un registro de una semana completa, permitiendo visualizar tendencias y comportamientos del ambiente.

### 8.1 Perspectiva del Producto.

El sistema S.A.P.O. Está diseñado como un producto **Standalone** (independiente). Toda la lógica de procesamiento, el servidor web de monitoreo y la gestión de alertas se ejecutan localmente en el microcontrolador ESP32. No requiere de servicios en la nube ni de una infraestructura de red externa para su funcionamiento básico.

### 8.2 Características de los Usuarios

El sistema está orientado a dos perfiles principales:

1. **Dueños de negocios:** Personas que requieren monitorear la seguridad de sus establecimientos de manera local y sencilla.
2. **Familias:** Usuarios domésticos que buscan una herramienta de prevención de incendios y monitoreo de aire en el hogar sin configuraciones complejas.

### 8.3 Suposiciones y Dependencias

- **Energía:** Suministro eléctrico constante de 5V DC para mantener la temperatura de operación de los sensores.
- **Conectividad Local:** Se depende de que el usuario se mantenga dentro del rango de alcance de la señal WiFi emitida por el ESP32 (Modo Access Point) para visualizar los datos.
- **Calibración:** Se asume un uso cotidiano y doméstico, donde los sensores se encuentren en ambientes con circulación de aire normal.

## 9. Alcance del proyecto

Aunque el proyecto se desarrolla para la materia de Sistemas Embebidos, su aplicación puede extenderse a distintos entornos reales.

Puede implementarse en hogares, oficinas o pequeños comercios como sistema básico de monitoreo de humo y calidad del aire. También puede escalarse hacia soluciones IoT más completas integrando notificaciones remotas, conexión a la nube o sensores más precisos.

Además, sirve como base para desarrollos en domótica, monitoreo ambiental inteligente y sistemas de prevención de riesgos.

En resumen, S.A.P.O (Sistema de Alerta de Partículas y Oxígeno) es un prototipo funcional con potencial de evolución hacia aplicaciones prácticas y comerciales.

## 10. Metodología de desarrollo

El proyecto se desarrollará utilizando la metodología ágil Scrum, con el objetivo de organizar el trabajo en iteraciones cortas y entregables funcionales. Para la gestión y seguimiento de tareas se utilizará la herramienta Trello, donde se organizarán los pendientes en tableros divididos por sprints, listas de actividades (pendiente, en proceso y terminado) y asignación de responsables.

Se establecen los siguientes sprints:

- **Sprint 1:** 16 de febrero de 2026 – 20 de marzo de 2026  
Enfocado en el desarrollo base del sistema (hardware + servidor web inicial).
- **Sprint 2:** 23 de marzo de 2026 – 17 de abril de 2026  
Enfocado en histórico, gráficas, optimización, pruebas finales y documentación.

Para el 17 de abril de 2026 se espera tener el sistema completamente funcional y listo para entrega.



## 11. Plan de trabajo por semanas

### 11.1 Sprint 1

- **Semana 1 (16 – 20 febrero)**
  - Diseño del sistema electrónico (diagrama de conexiones).
  - Definición de arquitectura del software.
  - Organización del tablero en Trello.
  - Investigación técnica mientras se consiguen los componentes.
- **Semana 2 (23 – 27 febrero)**
  - Conexión física de los sensores MQ-2 y MQ-135.
  - Pruebas individuales de lectura por monitor serial.
  - Definición y ajuste inicial de umbrales.
- **Semana 3 (2 – 6 marzo)**
  - Implementación de la lógica de activación de alarma.
  - Integración del buzzer, LED y pulsador físico.
  - Validación del correcto funcionamiento eléctrico.
- **Semana 4 (9 – 13 marzo)**
  - Implementación del modo AP (red propia de la ESP32).
  - Desarrollo del servidor web básico.
  - Visualización en tiempo real de valores (sin gráficas aún).
- **Semana 5 (16 – 20 marzo)**
  - Pruebas integrales del sistema completo.
  - Corrección de errores.
  - Entregable funcional base del sistema.

### 11.2 Sprint 2

- **Semana 1 (23 – 27 marzo)**
  - Implementación de almacenamiento en LittleFS.
  - Guardado periódico de datos (cada minuto).
  - Pruebas de lectura y recuperación de datos.
- **Semana 2 (30 marzo – 3 abril)**
  - Integración de Chart.js local (sin depender de internet).
  - Desarrollo de gráficas para MQ-2 y MQ-135.
  - Pruebas de carga de histórico.
- **Semana 3 (6 – 10 abril)**
  - Optimización del sistema (memoria y rendimiento).
  - Mejora de interfaz web.
  - Validación del conteo de activaciones.
- **Semana 4 (13 – 17 abril)**
  - Pruebas finales completas.
  - Validación contra requerimientos funcionales.
  - Documentación final.

- Preparación de presentación y entrega.

Con esta planificación se garantiza un desarrollo progresivo, controlado y alineado con los objetivos establecidos.

## 12. Equipo de trabajo

El equipo está conformado por:

- Gustavo Ugalde Rios
- Gael Olguín Rangel

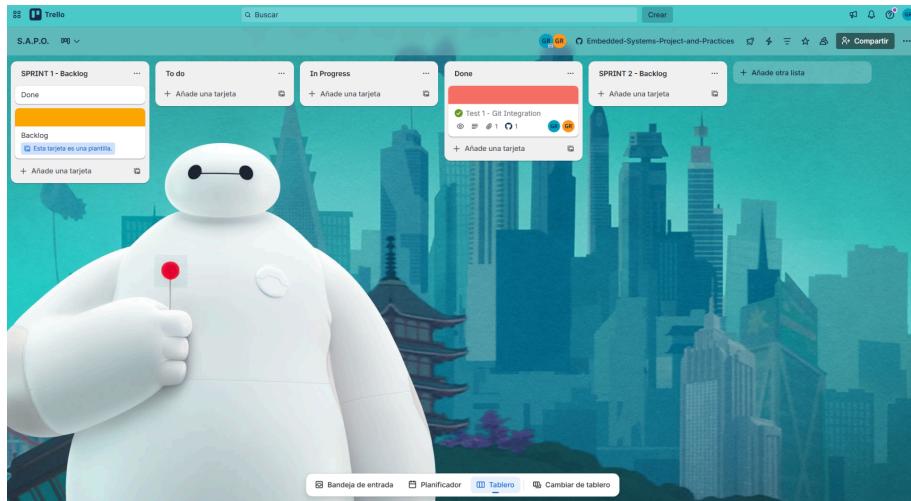
### 12.1 Distribución de roles

Integrante	Rol principal	Responsabilidades
Gustavo Ugalde Rios	Desarrollador principal (Hardware y Software)	Programación del ESP32, diseño electrónico, integración de sensores, desarrollo del servidor web, lógica del sistema, documentación técnica y elaboración del reporte final
Gael Olguín Rangel	Desarrollador y soporte técnico	Apoyo en desarrollo de software, pruebas del sistema, validación de funcionamiento, documentación parcial y apoyo en integración de componentes

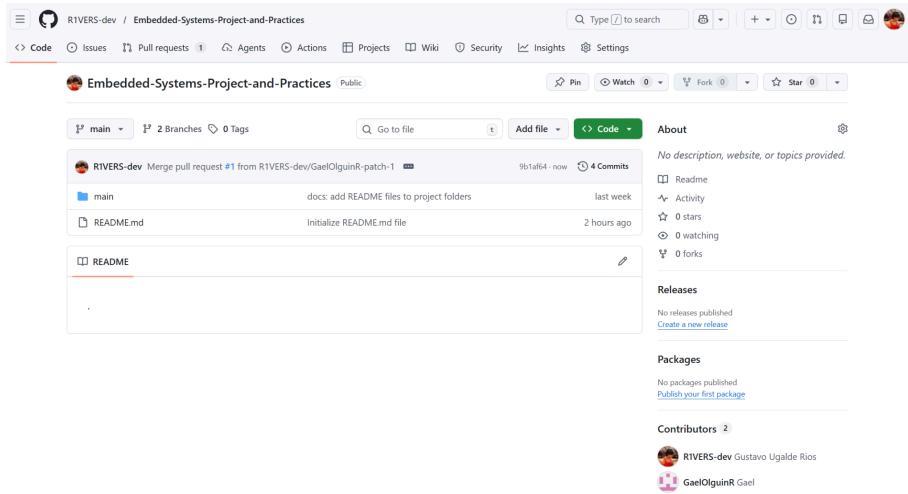
## 13. Herramientas de trabajo

Se realizará el seguimiento de actividades a través de la implementación de las herramientas.

### 13.1 Trello: [Enlace al tablero del equipo.](#)



### 13.2 GitHub del equipo: [Enlace al repositorio del proyecto.](#)



### 13.3 GitHub Embedded Systems Project : [Enlace al repositorio del proyecto.](#)

## 14. Requerimientos del sistema

Los siguientes requerimientos definen lo que el sistema debe hacer y cómo debe comportarse. Estos se dividieron principalmente en requerimientos funcionales y no funcionales.

#### **14.1 Requerimientos funcionales**

Estos **requerimientos funcionales** describen las funciones específicas que el sistema debe cumplir; es decir, las acciones y comportamientos esperados.

#### **14.2 Requerimientos funcionales**

ID	Requerimiento
RF1	El sistema debe detectar presencia de humo mediante el sensor MQ-2.
RF2	El sistema debe monitorear la calidad del aire mediante el sensor MQ-135.
RF3	El sistema debe activar un buzzer cuando se detecte una condición de incendio.
RF4	El sistema debe encender un LED cuando se active la alarma.
RF5	El sistema debe permitir desactivar la alarma desde la página web.
RF6	El sistema debe permitir desactivar la alarma físicamente mediante un pulsador.
RF7	El sistema debe mostrar en la web el estado actual (Normal/Incendio).
RF8	El sistema debe mostrar el conteo de activaciones de la alarma.
RF9	El sistema debe almacenar histórico de datos durante siete días.
RF10	El sistema debe mostrar gráficas del histórico en la página web.

#### **14.3 Requerimientos no funcionales**

Estos **requerimientos no funcionales** describen las condiciones bajo las cuales el sistema debe operar, incluyendo aspectos de rendimiento, seguridad, confiabilidad y usabilidad.

ID	Requerimiento
RNF1	El sistema debe funcionar con alimentación de 5V estable.
RNF2	La página web debe ser accesible desde cualquier dispositivo en la misma red WiFi.
RNF3	El sistema debe operar de manera continua sin reinicios inesperados.
RNF4	La interfaz web debe ser clara y fácil de interpretar.

**RNF5**

El sistema debe almacenar datos de forma eficiente en la memoria interna.

## 14.4 Interfaces de Hardware

Esta sección describe los componentes físicos que interactúan directamente con el sistema, incluyendo sensores, actuadores y los tipos de conexión utilizados.

### 14.4.1 Sensores

Componente	Tipo de Señal	Interfaz	Descripción de Interacción
MQ-2	Analógica	Entrada ADC	Envía señal analógica proporcional a la concentración de humo o gases inflamables. La ESP32 convierte la señal mediante el ADC interno.
MQ-135	Analógica	Entrada ADC	Proporciona señal analógica relacionada con la calidad del aire. La lectura es procesada por la ESP32 para determinar condiciones ambientales.

### 14.4.2 Actuadores

Componente	Tipo de Señal	Interfaz	Descripción de Interacción
Buzzer activo	Digital	GPIO (mediante transistor)	Se activa cuando la lógica del sistema detecta condición crítica. Genera alerta sonora inmediata.
LED rojo	Digital	GPIO	Indica visualmente el estado de alarma del sistema.
Pulsador físico	Digital	GPIO	Permite desactivar manualmente la alarma mediante entrada digital hacia la ESP32.

### 14.4.3 Buses y Comunicación Interna

Tipo de Comunicación	Uso en el Sistema	Observaciones
ADC (Convertidor Analógico-Digital)	Lectura de sensores MQ	Permite convertir señales analógicas a valores digitales procesables.
GPIO Digital	Control de actuadores y pulsador	Señales digitales de activación y lectura de estado.

WiFi (modo Access Point)	Comunicación con dispositivos cliente	Permite conexión local entre ESP32 y navegador web.
--------------------------	---------------------------------------	---

El sistema no utiliza buses I2C, SPI, UART o CAN para sensores en esta versión, ya que los dispositivos empleados funcionan mediante señales analógicas y digitales directas.

## 14.5 Interfaces de Software

Esta sección describe las interfaces lógicas y protocolos utilizados para la comunicación del sistema con dispositivos externos.

### 14.5.1 Interfaces de Comunicación

Interfaz	Protocolo	Descripción
Servidor Web Embebido	HTTP	Permite acceso a la interfaz web desde dispositivos conectados a la red local.
Intercambio de Datos	JSON	Los datos de sensores y estado del sistema se transmiten en formato estructurado JSON.
Red Local	WiFi (Access Point)	La ESP32 genera su propia red para permitir la conexión directa sin internet.

### 14.5.2 Interfaces Externas

Tipo	Descripción
Navegador Web	Interfaz gráfica accesible desde teléfono, tablet o computadora.
API Interna	Endpoints HTTP para consulta de datos y control de alarma.

## 14.6 Restricciones.

Esta sección describe las limitaciones técnicas, económicas y operativas que afectan el diseño e implementación del sistema S.A.P.O.

#### **14.6.1 Restricciones de Hardware**

Tipo	Descripción
Microcontrolador	El sistema debe utilizar exclusivamente una ESP32 como unidad de procesamiento.
Sensores	Se limita el uso a sensores MQ-2 y MQ-135 para detección ambiental.
Voltaje	La lógica del sistema debe operar a 3.3V, mientras que los sensores funcionan a 5V.
Memoria	El almacenamiento se limita a la memoria interna del ESP32 (LittleFS).

#### **14.6.2 Restricciones Regulatorias**

Tipo	Descripción
Certificación	El sistema no cuenta con certificación oficial contra incendios y no sustituye dispositivos industriales regulados.
Seguridad eléctrica	Debe cumplir con prácticas básicas de seguridad eléctrica para evitar cortocircuitos y sobrecargas.

#### **14.6.3 Restricciones Económicas**

Tipo	Descripción
Costo	El sistema debe mantenerse dentro de un presupuesto bajo, utilizando componentes comerciales accesibles.
Escalabilidad	No se contemplan módulos adicionales que incrementen significativamente el costo en esta versión inicial.

#### 14.6.4 Restricciones Energéticas

Tipo	Descripción
Alimentación	El sistema requiere fuente estable de 5V.
Consumo	Los sensores MQ presentan consumo constante debido a su calentador interno.
Autonomía	No se contempla alimentación mediante batería en esta versión.

#### 14.6.5 Restricciones de Tiempo

Tipo	Descripción
Frecuencia de lectura	Las lecturas de sensores deben ejecutarse en intervalos menores a 1 segundo.
Actualización web	La interfaz debe actualizar datos en intervalos de 1–2 segundos.

Activación de alarma	La alarma debe activarse inmediatamente tras validación de umbral crítico.
----------------------	--

## 15. Componentes del sistema.

Componente	Cantidad	Función
ESP32 (Steren)	1	Microcontrolador principal y servidor web
Sensor MQ-2	1	Detección de humo/incendio
Sensor MQ-135	1	Monitoreo de calidad del aire
Buzzer activo	1	Alarma sonora
LED rojo	1	Indicador visual de alerta
Pulsador	1	Desactivar alarma físicamente
Transistor NPN (2N2222)	1	Control del buzzer
Resistencia 1kΩ	1	Base del transistor
Resistencia 220Ω	1	Protección del LED
Protoboard	1	Montaje del circuito
Cables jumper	Varios	Conexiones
Fuente 5V	1	Alimentación del sistema

## 16. Protocolos y consideraciones técnicas

El diseño del sistema S.A.P.O considera buenas prácticas alineadas con estándares internacionales de calidad y seguridad. A continuación, se explican brevemente los estándares tomados como referencia y cómo se aplican en el proyecto:

- **ISO 9001 (Gestión de la Calidad):** establece lineamientos para asegurar que un producto o servicio cumpla consistentemente con requisitos definidos y busque mejora continua. En este proyecto se aplica mediante planificación por sprints, definición clara de requerimientos y validación del funcionamiento antes de la entrega final.
- **ISO/IEC 27001 (Seguridad de la Información):** establece controles para proteger la confidencialidad, integridad y disponibilidad de la información. En el sistema se aplica mediante el uso de contraseña segura en la red WiFi, restricción del acceso a red local y protección básica contra accesos no autorizados.
- **ISO 31000 (Gestión de Riesgos):** proporciona lineamientos para identificar, analizar y reducir riesgos. En este proyecto se aplica al definir umbrales adecuados, evitar falsas alarmas y tomar medidas de seguridad eléctrica para prevenir daños.

Estas referencias se adaptan al alcance académico del sistema, buscando que el desarrollo tenga fundamentos técnicos sólidos.

### 16.1 Seguridad del sistema (Lógica y Software)

Siguiendo principios de ISO/IEC 27001, se implementarán medidas básicas de protección y control:

- Establecer una contraseña segura para la red WiFi utilizada por la ESP32.
- Configurar el acceso a la red únicamente para dispositivos autorizados.
- Mantener el sistema dentro de una red local (LAN) y no exponerlo directamente a internet.
- Definir umbrales calibrados para evitar falsas alarmas.
- Implementar validación por lecturas consecutivas antes de activar la alarma.
- Manejar adecuadamente errores de conexión o reinicios inesperados.

Con estas acciones se protege la integridad del sistema y se garantiza su disponibilidad.

### 16.2 Seguridad física y eléctrica

En concordancia con principios de gestión de riesgos (ISO 31000) y buenas prácticas técnicas:

- Verificar conexiones antes de energizar el circuito.
- No manipular el sistema cuando esté alimentado.
- Utilizar resistencias de protección para LED y transistor.

- Asegurar correcta polaridad en la alimentación.
- Evitar cortocircuitos mediante montaje ordenado en protoboard.
- Utilizar fuente regulada y estable de 5V.

Estas medidas reducen el riesgo de daño a componentes y fallas del sistema.

### 16.3 Comunicación y red

Desde el enfoque de confiabilidad y seguridad:

- Utilizar protocolo HTTP en red local para visualización web.
- Intercambiar datos en formato JSON estructurado.
- Implementar reconexión automática en caso de pérdida de WiFi.

En una futura versión comercial podría incorporarse HTTPS con cifrado TLS para mayor seguridad.

### 16.4 Gestión de voltajes y compatibilidad eléctrica

Para garantizar funcionamiento seguro y estable:

- La ESP32 opera con lógica de 3.3V.
- Los sensores MQ se alimentan a 5V debido a su calentador interno.
- Verificar que la salida analógica no supere 3.3V en los pines ADC.
- Utilizar divisor de voltaje si es necesario.
- Emplear transistor para controlar el buzzer y evitar sobrecargar el GPIO.

En conjunto, estas consideraciones técnicas aseguran que el sistema sea estable, seguro y alineado con principios reconocidos de calidad, seguridad y control de riesgos.

## 17. Flujo del prototipo

El funcionamiento del prototipo puede entenderse como un flujo continuo de información entre sensores, microcontroladores y dispositivos de usuario.

### 1. Captura de datos (Sensores → ESP32)

Los sensores MQ-2 y MQ-135 detectan variaciones en el ambiente.

- El MQ-2 mide niveles de humo o gases inflamables.
- El MQ-135 mide la calidad del aire de forma relativa.

Ambos sensores envían señales analógicas hacia los pines ADC de la ESP32.

### 2. Procesamiento de datos (ESP32)

La ESP32 recibe las señales, las convierte a valores digitales y ejecuta la lógica del sistema.

- Compara los valores con umbrales definidos.
- Determina si existe condición de alarma.
- Actualiza el contador de activaciones.
- Guarda registros periódicos en la memoria interna (LittleFS).

### 3. Acción física (ESP32 → Actuadores)

Si se detecta una condición de incendio:

- Se activa el buzzer (alarma sonora).
- Se enciende el LED rojo (alerta visual).

En caso de que el usuario presione el pulsador físico o el botón en la página web, la ESP32 desactiva el buzzer.

### 4. Visualización y monitoreo (ESP32 → Dispositivos del usuario)

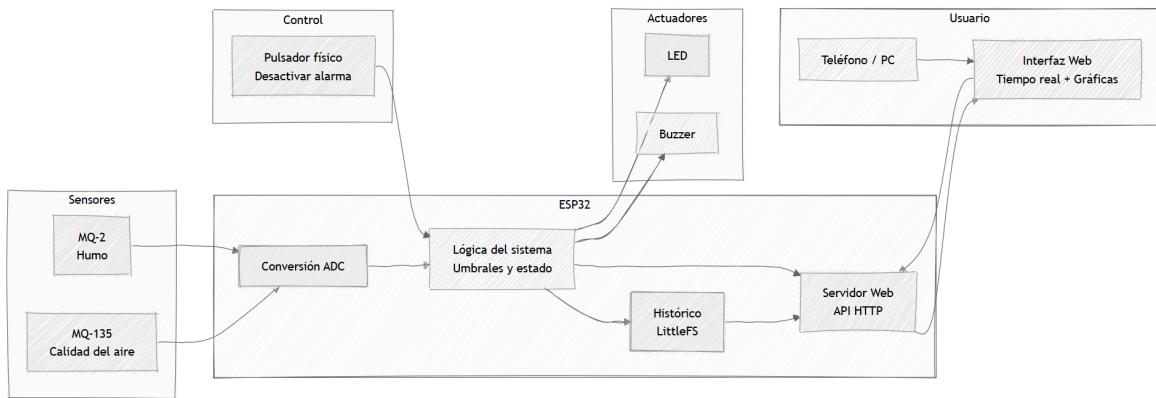
La ESP32 funciona como servidor web en modo Access Point.

Los dispositivos (teléfono, tableta o computadora) se conectan a su red local e ingresan a la dirección IP del sistema.

Desde la interfaz web se puede:

- Visualizar el estado actual (Normal/Incendio).
- Consultar valores en tiempo real.
- Observar gráficas del histórico.
- Desactivar la alarma.

Diagrama de interacción entre componentes:



## Referencias bibliográficas:

- I. Bangalore, I. (2025, 23 agosto). *MQ-135 Gas Sensor—Detect Air Quality Like a Pro Quickly.* IIES - Indian Institute Of Embedded Systems. <https://iies.in/blog/what-is-the-mq-135-sensor-and-how-does-it-work/>
- II. GeeksforGeeks. (2026, 17 enero). *Hypertext Transfer Protocol HTTP.* GeeksforGeeks. <https://www.geeksforgeeks.org/html/what-is-http/>
- III. *littlefs - Open Source Embedded Library.* (s. f.). <https://www.osrtos.com/library/littlefs/>
- IV. Loshin, P. (2021, 29 octubre). *ISO (International Organization for Standardization).* Search Data Center. <https://www.techtarget.com/searchdatacenter/definition/ISO>
- V. *MQ-2 Semiconductor Sensor for Smoke & Flammable Gas.* (s. f.). <https://www.winsen-sensor.com/product/mq-2.html>
- VI. Nick. (2025, 24 junio). *The ESP32 Chip explained: Advantages and Applications.* DeepSea. <https://www.deepseadev.com/en/blog/esp32-chip-explained-and-advantages/>
- VII. *Qué es un ADC: Convertidor de analógico a digital.* (2023, 30 septiembre). Más IP. <https://www.masip.es/diccionario/adc/>
- VIII. *What is a Standalone System or Device and How Does it Work? | Lenovo Philippines.* (s. f.). <https://www.lenovo.com/ph/en/glossary/standalone/>
- IX. Wikipedia contributors. (2005, 6 marzo). *JSON.* Wikipedia. <https://en.wikipedia.org/wiki/JSON>