

```

#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP_Mail_Client.h>
#include <Servo.h>

// --- DATOS WIFI ---
const char* WIFI_SSID = "XXXX";
const char* WIFI_PASSWORD = "XXXX";

// --- DATOS GMAIL ---
#define SMTP_HOST "smtp.gmail.com"
#define SMTP_PORT 465
#define AUTHOR_EMAIL "abiiledesma04@gmail.com"
#define AUTHOR_PASSWORD "qzdkbycyubxdxhvy"
#define RECIPIENT_EMAIL "cariledesma049@gmail.com"

// --- PINES ---
const int PIN_MQ2 = A0;
const int PIN_BOTON = 5; // D1
const int PIN_SERVO = 4; // D2
const int PIN_BUZZER = 14; // D5
const int PIN_LED = 12; // D6 (Va al pin IN del Relé)

const int UMBRAL_GAS = 700;

// --- CONFIGURACIÓN DE RELÉ ---

#define RELE_ENCENDIDO HIGH
#define RELE_APAGADO LOW

SMTPSession smtp;
Servo miServo;

```

```
// --- VARIABLES DE ESTADO ---
bool alarmaActiva = false;
bool correoGasEnviado = false;
unsigned long ultimoTiempoBoton = 0;
unsigned long tiempoAnteriorParpadeo = 0;
int estadoLuz = RELE_APAGADO;

// Prototipo
void smtpCallback(SMTP_Status status);

void setup() {
    Serial.begin(115200);
    Serial.println("\n\n--- INICIO SISTEMA ---");

    pinMode(PIN_BOTON, INPUT_PULLUP);
    pinMode(PIN_BUZZER, OUTPUT);
    pinMode(PIN_LED, OUTPUT);

    // Asegurar que el relé empiece APAGADO (HIGH)
    digitalWrite(PIN_LED, RELE_APAGADO);

    // Inicializar Servo
    miServo.attach(PIN_SERVO);
    miServo.write(0);
    delay(200);
    miServo.detach();

    // --- CONEXIÓN WIFI ---
    WiFi.mode(WIFI_STA);
    WiFi.setSleepMode(WIFI_NONE_SLEEP);
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

    unsigned long inicio = millis();
```

```

while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
    // Timeout de 15 segundos
    if (millis() - inicio > 15000) {
        WiFi.disconnect();
        WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
        inicio = millis();
    }
}
Serial.println("\nConectado!");

// --- HORA NTP ---
configTime(0, 0, "pool.ntp.org", "time.nist.gov");
while (time(nullptr) < 1000000000) {
    delay(100);
}

smtp.debug(0);
smtp.callback(smtpCallback);
}

void enviarCorreo(String asunto, String mensaje) {
if (WiFi.status() != WL_CONNECTED) {
    WiFi.disconnect();
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    for(int i=0; i<50; i++) { if(WiFi.status() == WL_CONNECTED) break; delay(100); }
}
if (WiFi.status() != WL_CONNECTED) return;

ESP_Mail_Session session;
session.server.host_name = SMTP_HOST;

```

```
session.server.port = SMTP_PORT;
session.login.email = AUTHOR_EMAIL;
session.login.password = AUTHOR_PASSWORD;
session.login.user_domain = "";

SMTP_Message message;
message.sender.name = "Alarma";
message.sender.email = AUTHOR_EMAIL;
message.subject = asunto;
message.addRecipient("Usuario", RECIPIENT_EMAIL);
message.text.content = mensaje;

Serial.println("Enviando mail...");
if (smtp.connect(&session)){
    MailClient.sendMail(&smtp, &message);
}
smtp.sendingResult.clear();
}

void activarAlarma() {
    alarmaActiva = true;

    // 1. Correo
    if (!correoGasEnviado) {
        enviarCorre("ALERTA: Fuga de gas detectada", "Ambiente peligroso. Por favor envie ayuda.");
        correoGasEnviado = true;
    }

    // 2. Ruido
    tone(PIN_BUZZER, 1000);

    // 3. Servo (Abrir)
```

```

    miServo.attach(PIN_SERVO);
    miServo.write(180);
    delay(1000);
    miServo.detach();
}

void desactivarAlarma() {
    Serial.println("Usuario desactivó alarma. REINICIANDO...");

    noTone(PIN_BUZZER);

    digitalWrite(PIN_LED, RELE_APAGADO);

    delay(30000);
    ESP.restart(); // Reinicio del sistema
}

void loop() {
    // --- PARPADEO DEL RELÉ (LÁMPARA 220V) ---
    if (alarmaActiva) {
        unsigned long tiempoActual = millis();

        // Parpadeo cada 1 segundo (1000ms) para proteger el relé de cambios muy
        rápidos
        if (tiempoActual - tiempoAnteriorParpadeo >= 1000) {
            tiempoAnteriorParpadeo = tiempoActual;

            // Invertir estado usando las definiciones de relé
            if (estadoLuz == RELE_APAGADO) {
                estadoLuz = RELE_ENCENDIDO;
            } else {
                estadoLuz = RELE_APAGADO;
            }
            digitalWrite(PIN_LED, estadoLuz);
        }
    }
}

```

```

        }

    } else {
        // Si no hay alarma, mantener apagado (HIGH)
        digitalWrite(PIN_LED, RELE_APAGADO);
    }

    // --- SENSOR ---
    int lecturaGas = analogRead(PIN_MQ2);
    if (lecturaGas > UMBRAL_GAS && !alarmaActiva) {
        delay(200);
        if (analogRead(PIN_MQ2) > UMBRAL_GAS) activarAlarma();
    }

    // --- BOTÓN ---
    if (digitalRead(PIN_BOTON) == LOW) {
        if (millis() - ultimoTiempoBoton > 1000) {
            if (alarmaActiva) {
                desactivarAlarma();
            } else {
                Serial.println("Boton Panico");
                enviarCorreo("ALERTA BOTON", "Boton panico presionado. Se necesita ayuda");
            }
            ultimoTiempoBoton = millis();
        }
    }
}

void smtpCallback(SMTP_Status status) {}

```