



## **Actividad |2| Alarma para Incendios.**

**Internet de las Cosas.**

---

**Ingeniería en Desarrollo de Software.**



TUTOR: Marco Alonso Rodríguez Tapia.

ALUMNO: Ramón Ernesto Valdez Felix.

FECHA: 18/02/2026.

<b>Introducción.....</b>	<b>3</b>
<b>Descripción.....</b>	<b>3</b>
<b>Justificación.....</b>	<b>4</b>
<b>Desarrollo.....</b>	<b>4</b>
<b>Armado del Circuito.....</b>	<b>5</b>
<b>Codificación.....</b>	<b>7</b>
<b>Emulación del Circuito.....</b>	<b>10</b>
<b>Conclusion.....</b>	<b>12</b>
<b>Referencias.....</b>	<b>12</b>

## Introducción.

En esta segunda actividad de la materia de Internet de las Cosas (IoT), consiste en el diseño y simulación de un sistema de seguridad inteligente en Tinkercad. El objetivo principal es evolucionar el concepto de una alarma de movimiento (visto en materiales previos) hacia un sistema de detección de incendios mediante un sensor de gas. El proyecto integra hardware y software de manera estructurada: se utiliza una placa Arduino como cerebro del sistema, conectada a una placa de pruebas donde se articulan un sensor de gas, dos luces LED y un componente piezoeléctrico (buzzer).

La lógica de programación se basa en la lectura de datos analógicos. En el código, se definen variables enteras para gestionar los periféricos; el sensor se configura como entrada (INPUT) para captar la concentración de partículas, mientras que el piezoeléctrico actúa como salida (OUTPUT). La funcionalidad crítica reside en una estructura condicional dentro del void loop: si el sensor detecta una presencia de gas mayor o igual a 600, se dispara la alerta sonora, garantizando una respuesta inmediata ante emergencias.

## Descripción.

En esta segunda actividad de Internet de las Cosas (IoT), se centra en la evolución de un sistema de seguridad básico, pasando de una alarma de movimiento a un dispositivo de prevención de incendios. Utilizando el entorno virtual de Tinkercad, los estudiantes deben integrar componentes fundamentales como una placa Arduino, un sensor de gas, un elemento piezoeléctrico (buzzer), dos LEDs y una resistencia de 220 ohmios sobre una placa de pruebas.

El núcleo del proyecto reside en la programación lógica: se definen variables de tipo entero para gestionar los datos. El sensor de gas se configura como INPUT para monitorear el ambiente, mientras que el piezoeléctrico se establece como OUTPUT para emitir la alerta. La funcionalidad crítica se dicta en el bloque void loop, donde se establece un umbral de seguridad: si la lectura del sensor es mayor o igual a 600, se dispara la alarma sonora y visual; de lo contrario, el sistema permanece en silencio,

garantizando una respuesta automatizada ante riesgos.

## Justificación.

La justificación de este proyecto radica en la aplicación práctica de los fundamentos del Internet de las Cosas (IoT) para resolver problemas de seguridad en entornos reales. Al realizar esta transición de una alarma de movimiento a una de incendios, el estudiante no sólo replica un circuito, sino que comprende la versatilidad de los sistemas embebidos y la importancia de la automatización en la protección de vidas y bienes.

Desde una perspectiva técnica, el uso de Tinkercad permite experimentar con sensores analógicos y actuadores digitales sin riesgo de dañar componentes físicos, facilitando el aprendizaje del ciclo percepción-procesamiento-acción. Programar un umbral crítico de 600 enseña la relevancia de la calibración de datos: una lectura mayor o igual dispara una respuesta inmediata (sonido y luz), mientras que un valor menor garantiza estabilidad, evitando falsas alarmas. En última instancia, esta actividad desarrolla el pensamiento computacional necesario para diseñar ciudades inteligentes y hogares conectados más seguros.

## Desarrollo.

El desarrollo de esta actividad se centra en la transición del aprendizaje teórico a la implementación práctica de un sistema de control básico dentro del ecosistema del Internet de las Cosas. Al simular el circuito, se obtiene un sistema funcional que replica el comportamiento de un detector de humo real, demostrando la capacidad de automatizar procesos cotidianos mediante la programación de microcontroladores.

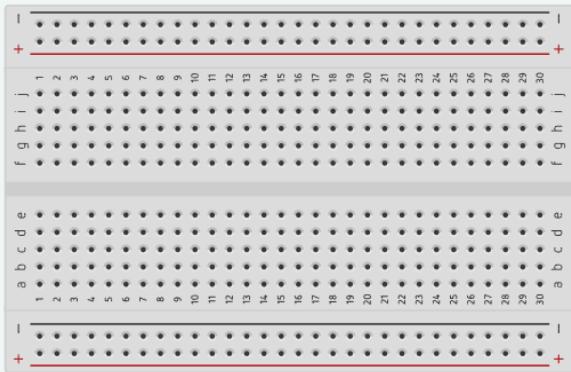
[Link: GitHub.](#)

[Link: Actividad.](#)

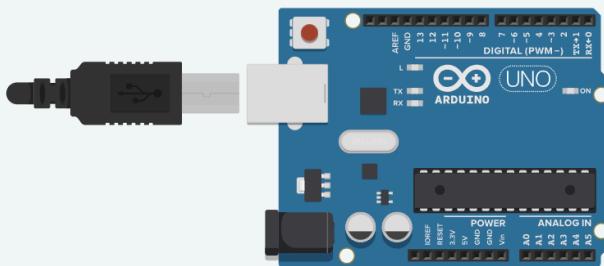
# Armado del Circuito.

Utilizamos los componentes básicos de la herramienta de trabajo tinkerCAD para realizar la actividad de la creación de un alarma de humo o detector de humo.

## Circuitos a utilizar: Alarma de humo.



**Placa de pruebas (Breadboard):** Es la base blanca con agujeros que permite conectar los componentes entre sí sin necesidad de soldar. Las líneas horizontales y verticales están conectadas internamente para compartir electricidad.



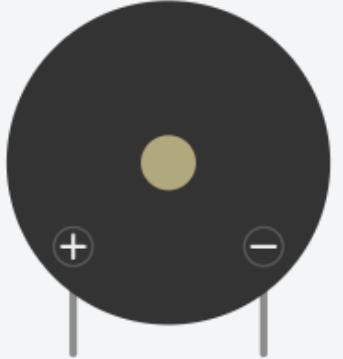
**Placa Arduino Uno:** Es el "cerebro" programable. Recibe el código que escribes y envía señales eléctricas a través de sus pines para controlar los LEDs.

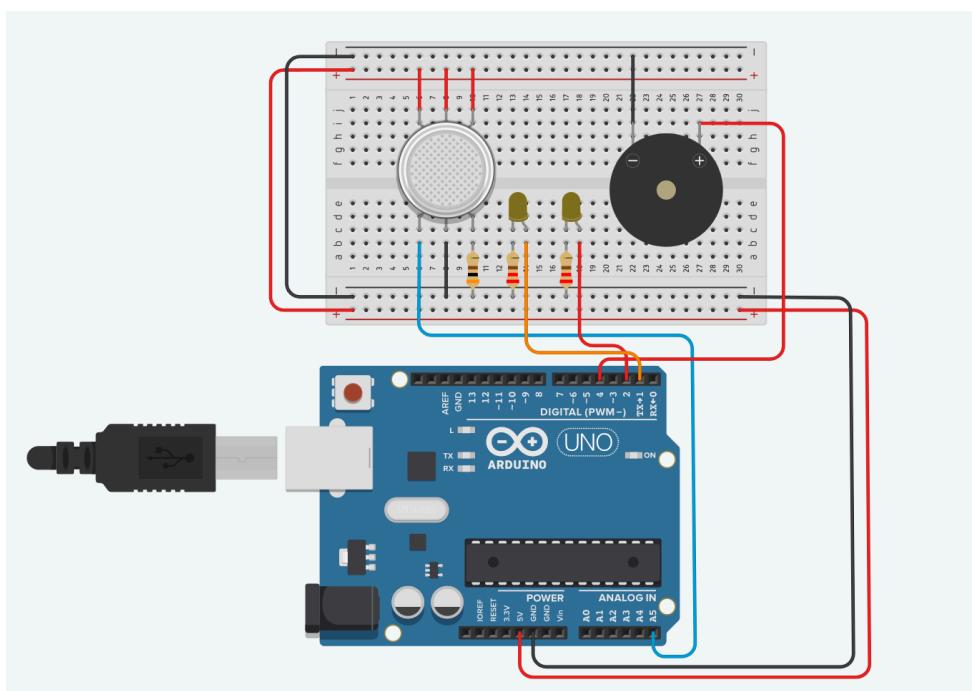


**Diodos LED (Amarillo):** Son los indicadores luminosos. Solo permiten el paso de la corriente en un sentido y se encienden cuando reciben energía desde el Arduino.



**Resistencias ( 2 de 220 y 1 de 300 ohms):** Son esos pequeños cilindros con bandas de colores. Su función es limitar la cantidad de corriente que llega a los LEDs para evitar que se quemen.

	<p><b>El Piezoeléctrico (Zumbador o Buzzer):</b> El componente negro circular en la parte superior es un transductor piezoeléctrico. Es un dispositivo que convierte señales eléctricas en vibraciones mecánicas para generar sonido (pitidos).</p>
	<p><b>Sensor de Gas (MQ-2):</b> Es un sensor que detecta la presencia de diferentes gases en el aire, como gas LP, propano, hidrógeno y humo. Dentro tiene un filamento que se calienta. Cuando los gases entran en contacto con el sensor, la conductividad interna cambia y el sensor entrega una variación de voltaje.</p>
<p><b>Cables de conexión (Jumpers):</b> Los hilos de colores que transportan la energía y los datos entre la placa Arduino y la placa de pruebas. Con todos estos circuitos básicos realizamos el armado de actividad (Alarma de Incendios).</p>	



## Codificación.

En este punto de la actividad realizaremos la documentación de la codificación de la alarma de humo o detector de humo que nos pide realizar la materia de internet de las cosas como primer actividad y daremos un explicación breve de lo que realiza el código.

### Codificación de alarma de humo.

```
const int pinSensor = A0;
const int pinLed1 = 2;
const int pinLed2 = 3;
const int pinBuzzer = 4;
int umbral = 600;
```

#### 1. Declaración de Variables y Constantes

Al principio del código, le ponemos "nombre" a los agujeros (pines) de la placa Arduino para no confundirnos.

- **pinSensor = A0:** Es una entrada analógica. A diferencia de un botón (que

```

void setup() {
    pinMode(pinSensor, INPUT);
    pinMode(pinLed1, OUTPUT);
    pinMode(pinLed2, OUTPUT);
    pinMode(pinBuzzer, OUTPUT);
    Serial.begin(9600);

    void loop() {
        int valorGas = analogRead(pinSensor);
        Serial.print("Presencia de Gas: ");
        Serial.println(valorGas);

        if (valorGas >= umbral) {
            digitalWrite(pinLed1, HIGH);
            digitalWrite(pinLed2, HIGH);
            tone(pinBuzzer, 1000);
        }
    }
}

```

es solo encendido/apagado), este sensor mide "qué tanto" gas hay, devolviendo valores entre 0 y 1023.

- **umbral = 600**: Esta es la cifra mágica. Es el punto exacto donde decidimos que hay peligro.
- **pinLed y pinBuzzer**: Son las salidas que darán el aviso visual y sonoro.

## 2. Configuración (**void setup**)

Esta parte se ejecuta una sola vez cuando enciendes el Arduino.

- **pinMode**: Aquí definimos el rol de cada pin. El sensor es un entrada porque mete información al Arduino. Los LEDs y el Buzzer son salida porque el Arduino les envía una orden.
- **Serial.begin(9600)**: Abre un canal de comunicación con tu computadora. Te permite ver en la pantalla de Tinkercad (Monitor Serie) los números exactos que está leyendo el sensor.

## 3. El Bucle Infinito (**void loop**)

```

    }

else {

    digitalWrite(pinLed1, LOW);

    digitalWrite(pinLed2, LOW);

    noTone(pinBuzzer);

}

delay(100);

}

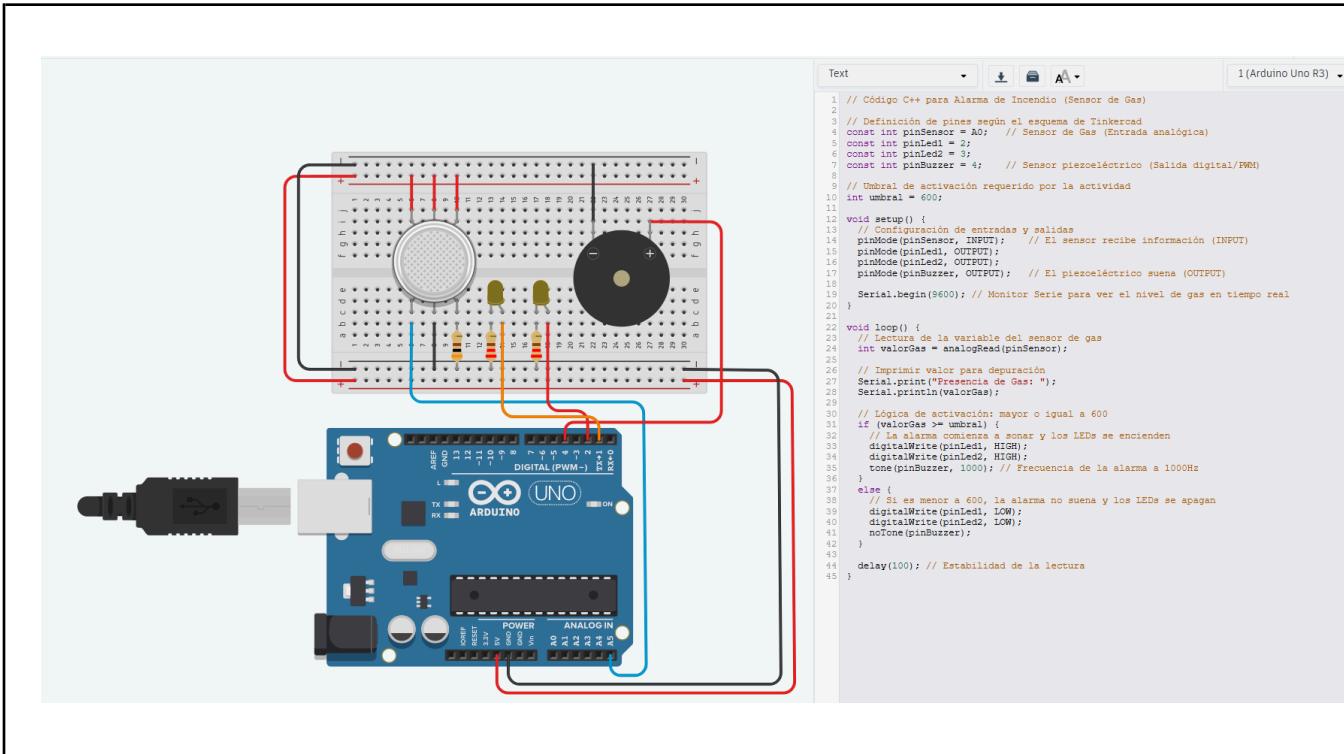
```

Aquí es donde ocurre la magia constantemente (miles de veces por segundo).

- **analogRead(pinSensor)**: El Arduino "mira" el sensor y guarda el número en la variable **valorGas**.
- **La Condición (if / else)**:
  - **Si (if)** el valor es 600 o más: El Arduino envía energía a los LEDs y activa el sonido con **tone**.
  - **De lo contrario (else)**: Si el valor baja de 600, el Arduino corta la energía y apaga el sonido con **noTone**. Esta es la parte que garantiza que la alarma se detenga cuando el gas desaparece.

#### 4. Estabilidad (**delay**)

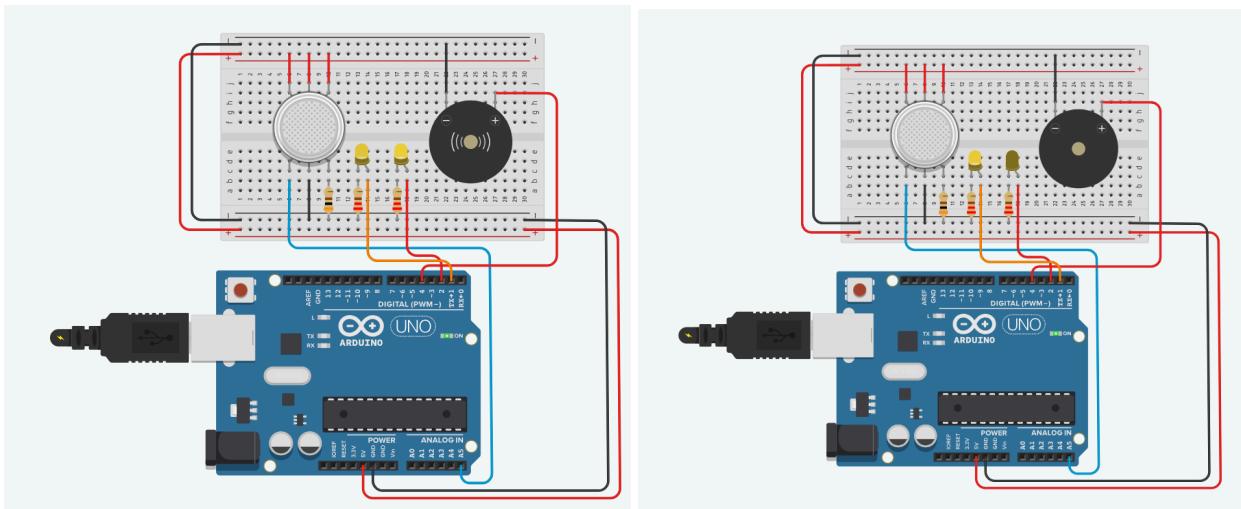
- **delay(100)**: Hace que el Arduino espere una décima de segundo antes de volver a medir. Sin esto, el código iría tan rápido que los valores en pantalla serían difíciles de leer y el sistema podría volverse inestable.

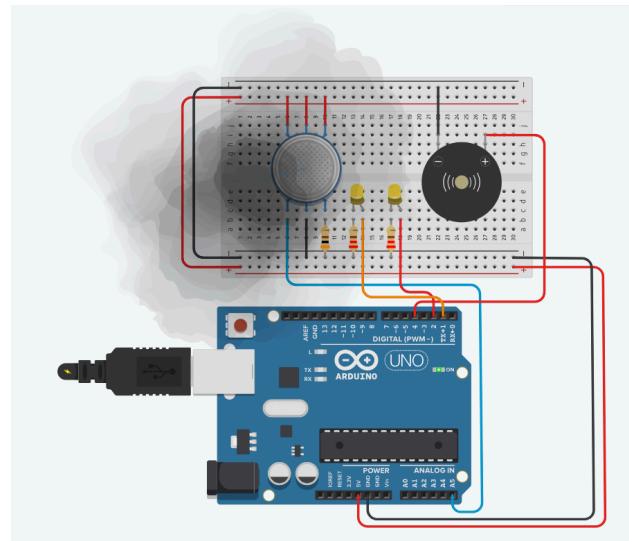


## Emulación del Circuito.

Al emular este código en una plataforma como Tinkercad, estás ejecutando un programa que convierte instrucciones lógicas en señales eléctricas reales. Aquí se tiene la emulación de los circuitos en su ejecución sonando al detectar el humo de un posible incendio, encendiéndo y apagando uno de los LED aplicado en la lógica de la codificación detector de humo.

### Prueba de emulación de actividad de la alarma de incendios.





< Gallery    Private



## Frantic Waasa

React 0



Circuit by  
Ramón Valdez

Tinker this

Share to Classroom

Copy link

Design is visible only to you.

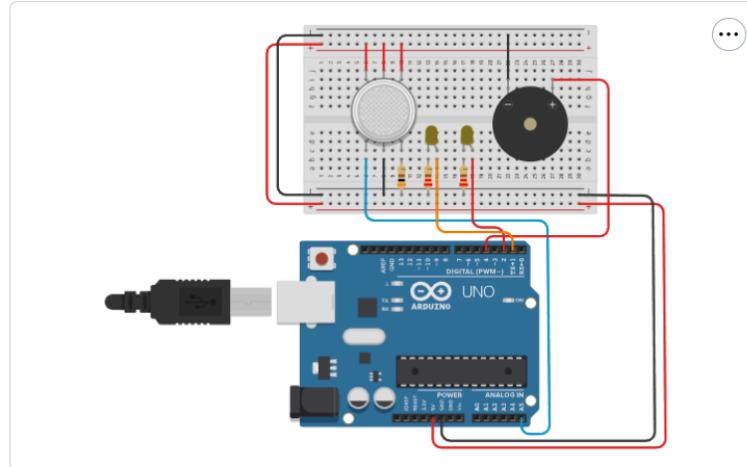
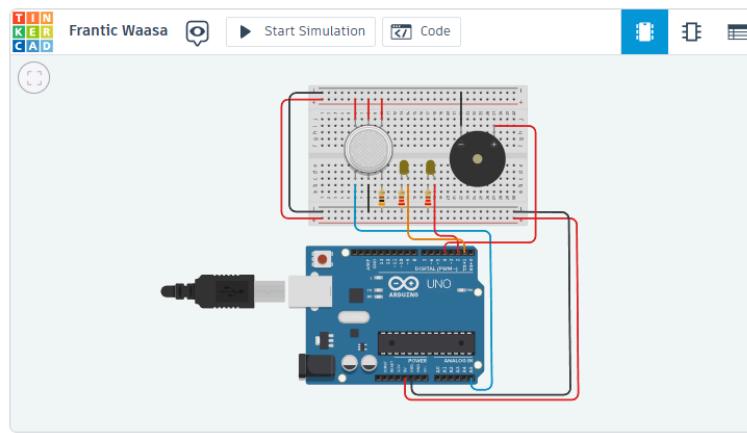
[Change visibility to share](#)

Edited February 19, 2026

Created February 18, 2026



Report content



<https://www.tinkercad.com/things/iEI7xul8wIx-frantic-waasa>

## Conclusion.

En conclusión: La creación de una alarma de incendio mediante el uso de sensores de gas y actuadores piezoelectricos representa un pilar fundamental en el Internet de las Cosas (IoT), con un impacto directo tanto en la seguridad industrial como en el entorno doméstico. En el campo laboral, este tipo de proyectos demuestra cómo la automatización y el monitoreo en tiempo real pueden prevenir catástrofes, optimizar protocolos de respuesta y proteger activos críticos mediante la detección temprana de anomalías químicas o humo.

En la vida cotidiana, la integración de estos dispositivos convierte un hogar convencional en uno inteligente, brindando tranquilidad a través de sistemas de alerta que no solo suenan localmente, sino que pueden enviar notificaciones remotas. La importancia de esta actividad radica en la capacidad de transformar datos analógicos del entorno en decisiones digitales críticas. Dominar estas tecnologías permite al profesional diseñar soluciones escalables que salvan vidas, reducen costos por daños y establecen una base sólida para la creación de ciudades e infraestructuras más resilientes y conectadas.

## Referencias.

*Google. (n.d.). Gemini. Retrieved February 19, 2026, from*

*<https://gemini.google.com/>*

*Tinkercad. (n.d.). Tinkercad. Retrieved February 19, 2026, from*

*<https://www.tinkercad.com/dashboard/designs/all>*