Ingeniería
en Desarrollo de Software

Actividad | 2 | Alarma para Incendios

Internet de las Cosas

Ingeniería en Desarrollo de Software

TUTOR: Marco Alonso Rodríguez Tapia

ALUMNO: Carlos Fco Estrada Salazar

FECHA: 27/Octubre/2025

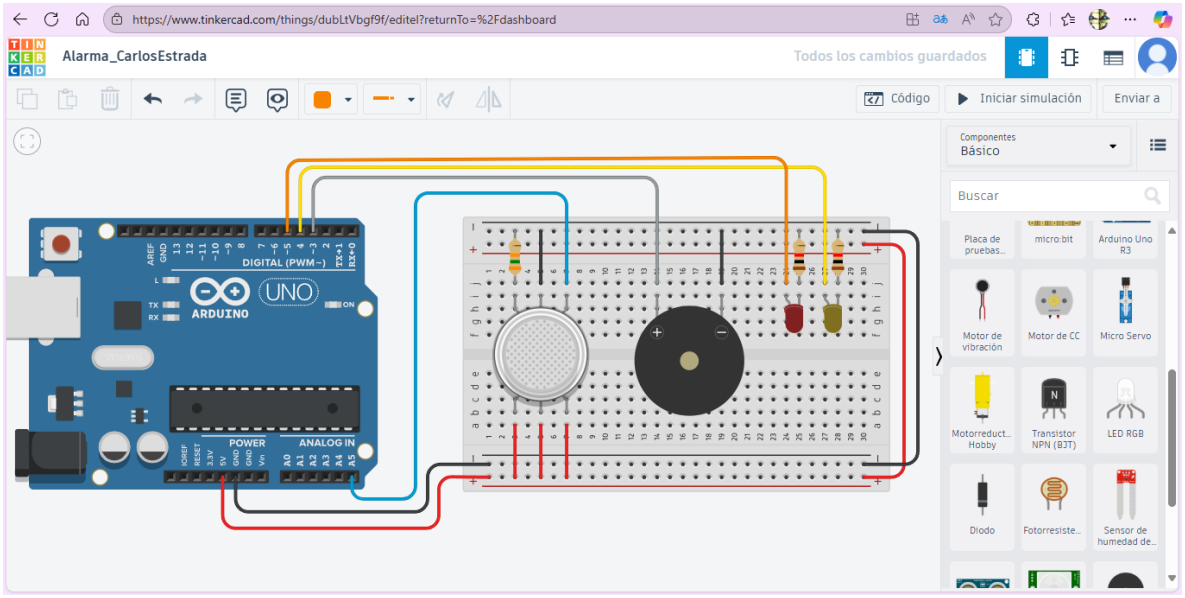
INDICE

Tinkercad Link:
https://www.tinkercad.com/things/dubLtVbgf9f/editel?returnTo=%2Fdashboard&sharecode=Rh4B1ed30Pefm0aO6m8JjSO1jgyzIQhKtzUIHjJOZpY

GitHub Link de Actividad:

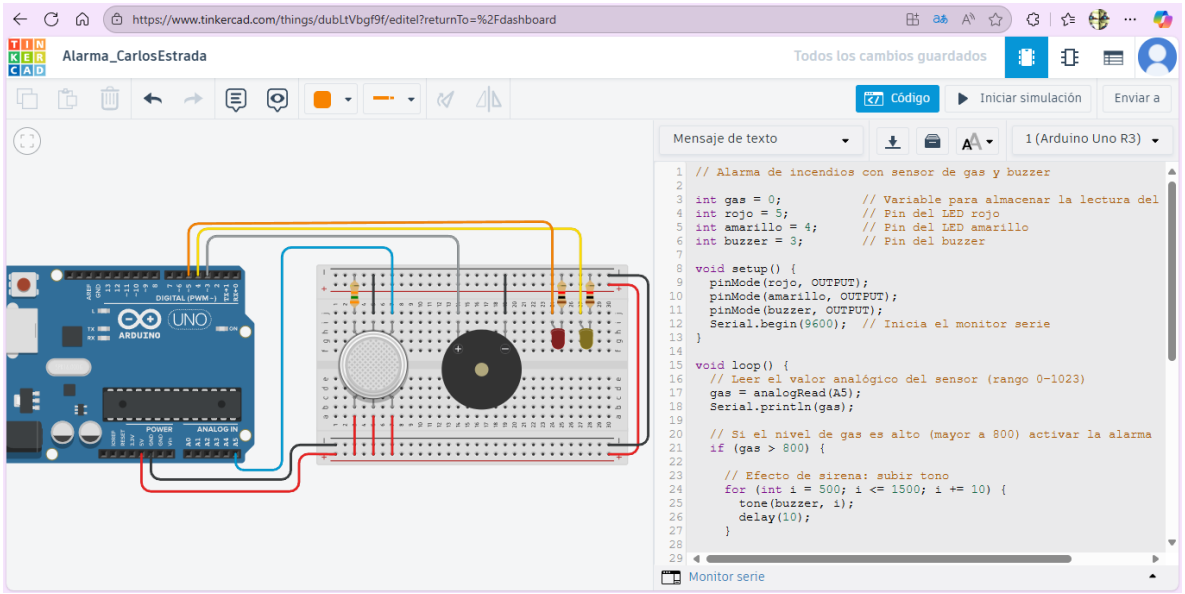
GitHub Link de Código: https://github.com/Calinny17/Internet-de-las-cosas/blob/19acc6d0bf54c873bd30f136e1ccb855b0c8d69e/Alarma%20Incendios

Armado del Circuito



Para el armado del circuito, utilizo los componentes de la actividad 2, y para el diseño tomo como ejemplo la tutoría 2 de la materia.

Codificación



Código que utilice para hacer funcionar la alama.

Funcionamiento

```
// Alarma de incendios con sensor de gas y buzzer

int gas = 0;           // Variable para almacenar la lectura del sensor de gas
int rojo = 5;          // Pin del LED rojo
int amarillo = 4;      // Pin del LED amarillo
int buzzer = 3;        // Pin del buzzer
```

- Declaro las variables que voy a utilizar.

```
void setup() {
  pinMode(rojo, OUTPUT);
  pinMode(amarillo, OUTPUT);
  pinMode(buzzer, OUTPUT);
  Serial.begin(9600);  // Inicia el monitor serie
}
```

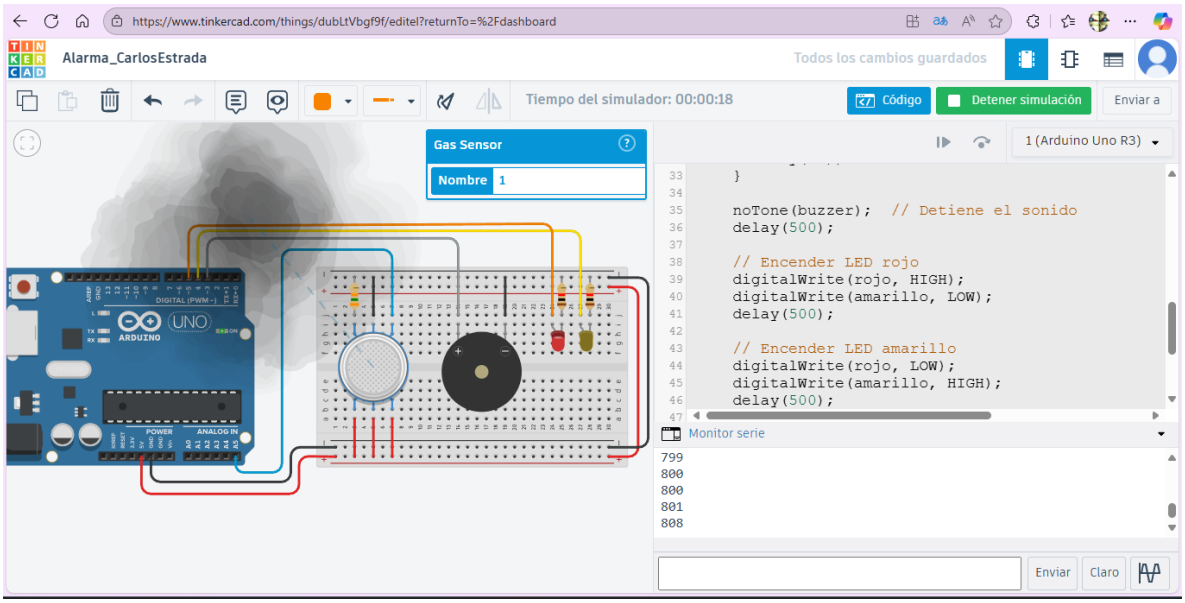
- Configuro los pines, como entrada o salida, que voy a utilizar y se activa salida por USB para ver las lecturas en el monitor serie (para calibrar).

```
void loop() {
  // Leer el valor analógico del sensor (rango 0-1023)
  gas = analogRead(A5);
  Serial.println(gas);

  // Si el nivel de gas es alto (mayor a 800) activar la alarma
  if (gas > 800) {

    // Efecto de sirena: subir tono
    for (int i = 500; i <= 1500; i += 10) {
      tone(buzzer, i);
      delay(10);
    }

    // Efecto de sirena: bajar tono
    for (int i = 1500; i >= 500; i -= 10) {
      tone(buzzer, i);
      delay(10);
    }

    noTone(buzzer);  // Detiene el sonido
    delay(500);
```

- Se lee el valor entre 0 y 1023 proporcional a la concentración detectada, Si la lectura supera el umbral:
- Se abre un bucle para iniciar la variable “i” con un valor de **500**.
- Con la condición “**i <= 1500**” indico que el bucle seguirá mientras “i” sea menor o igual a **1500**.
- La condición “**i += 10**” me ayuda a aumentar “i” en 10 en cada repetición, esto ara que “i” tome los valores de 500, 510, 520… 1500.
- Llamo la función “**tone ( )**” para generar una señal en el buzzer, en el segundo parámetro utilizo el valor almacenado en la variable “i”, esto seria la frecuencia en Hertz (Hz), asi al principio serán 500Hz, luego 510 Hz, etc. Con cada llamada cambiara la frecuencia, haciendo que suba el volumen gradualmente.
- La condicion “**delay(10);**” pausa el programa por 10 milisegundos, en este tiempo el buzzer estará sonando a la frecuencia establecida por “tone ()” hasta la siguiente iteración. Esa pequeña pausa determina la “suavidad” y velocidad del ascenso de la sirena.

- Cierra el primer “for”. Al terminar, la frecuencia habrá subido desde 500 hasta 1500 Hz.
- Inicia un segundo bucle que hace lo contrario: comienza en **1500 Hz** y va **disminuyendo** i de 10 en 10 hasta 500.

Emulación del circuito



Se genera la emulación y cuando el umbral excede su limite se enciende tanto la alarma como los leds.