

*Sistema de alarma, detector y
medidor de gases.*

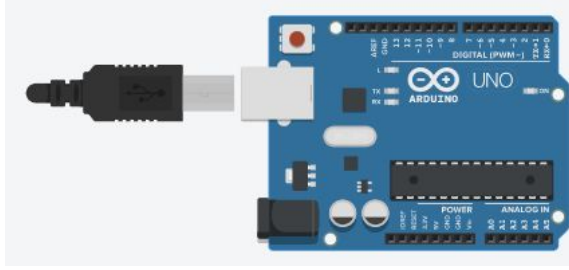
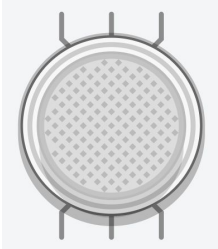

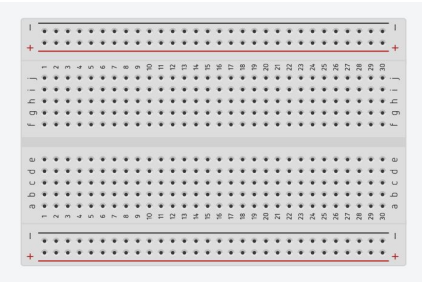
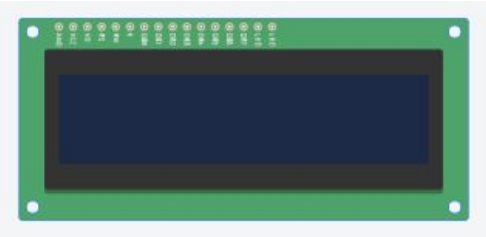
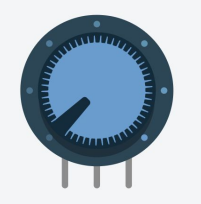
Erica Gerez





Contenidos

- I. Materiales
- II. Código fuente y librerías utilizadas
- III. Esquema
- IV. Problemáticas surgidas
- V. Proceso del armado

I. Materiales

<i>Nombre</i>	<i>Ilustración gráfica</i>
Arduino Uno R3 (2 unidades)	 A photograph of an Arduino Uno R3 microcontroller board. It is a blue PCB with a USB Type-B port, a DC power jack, a reset button, and various digital and analog pins. The text 'ARDUINO UNO' and 'DIGITAL (VCC)' are visible on the board.
Sensor de gas	 A top-down view of a circular gas sensor. It has a metallic, perforated surface and four pins extending from the bottom edge.
Buzzer	 A top-down view of a small, circular black buzzer. It has a yellow circular center and two pins extending from the bottom edge.
Breadboard	 A photograph of a standard white breadboard. It features a grid of holes for components, with two rows of power rails (VCC and GND) marked with '+' and '-' signs.
LCD 16 x 2	 A photograph of a 16x2 character LCD display. It has a green PCB with a dark blue screen and two rows of pins along the top edge.
Potenciómetro	 A top-down view of a potentiometer. It has a circular blue body with a central wiper and three pins extending from the bottom edge.

LED (6 unidades)	
Resistencia (5 unidades)	

II. Código

En el código de este trabajo se hace uso de la librería *LiquidCrystal* ya que la misma provee el manejo y manipulación del componente LCD. También se utilizó la librería *Wire* para la lograr la comunicación I2C entre las placas Arduino.

Código de la placa Arduino 1.

```
#include <LiquidCrystal.h>
#include <Wire.h>

LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

int humo;
int rojo=11;
int amarillo=12;
int verde=13;
int alarma=10;

void setup()
{
  Wire.begin(1);
  Wire.onRequest(peticion);

  Serial.begin(9600);

  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.print("Gas: ");
  lcd.setCursor(0,1);
  lcd.print("Estado: ");

  pinMode(rojo,OUTPUT);
  pinMode(amarillo,OUTPUT);
  pinMode(verde,OUTPUT);
  pinMode(alarma,OUTPUT);
}
```

```

void peticion(){
    Serial.println("Se pidio comunicacion con la central");

    if(humo>960){
        Wire.write("Llamando al nro 911 emergencias");
    }
    else{
        Wire.write("El nivel de gas es inofensivo..");
    }
}

```

```

void loop(){
    lcd.setCursor(5, 0);
    lcd.print(humo);
    delay(500);

    humo = analogRead(A5);
    delay(500);

    if(humo<=950){
        digitalWrite(verde,HIGH);
        digitalWrite(amarillo,LOW);
        digitalWrite(rojo,LOW);
        noTone(alarma);
        lcd.setCursor(8, 1);
        lcd.print("SEGURO");
    }

    if((humo>950) and (humo<960)){
        digitalWrite(verde,LOW);
        digitalWrite(amarillo,HIGH);
        digitalWrite(rojo,LOW);
        noTone(alarma);
        lcd.setCursor(8, 1);
        lcd.print("CUIDADO");
    }

    if(humo>960){
        digitalWrite(verde,LOW);
        digitalWrite(amarillo,LOW);
        digitalWrite(rojo,HIGH);
        activarAlarma();
        lcd.setCursor(8, 1);
        lcd.print("PELIGRO");
    }
}

```

```

void activarAlarma(){
    tone(alarma,300);
    delay(500);
}

```

```
    noTone(alarma);
    delay(500);
    tone(alarma,300);
}
```

Código de la placa Arduino 2.

```
#include <Wire.h>
```

```
char mensaje1;
```

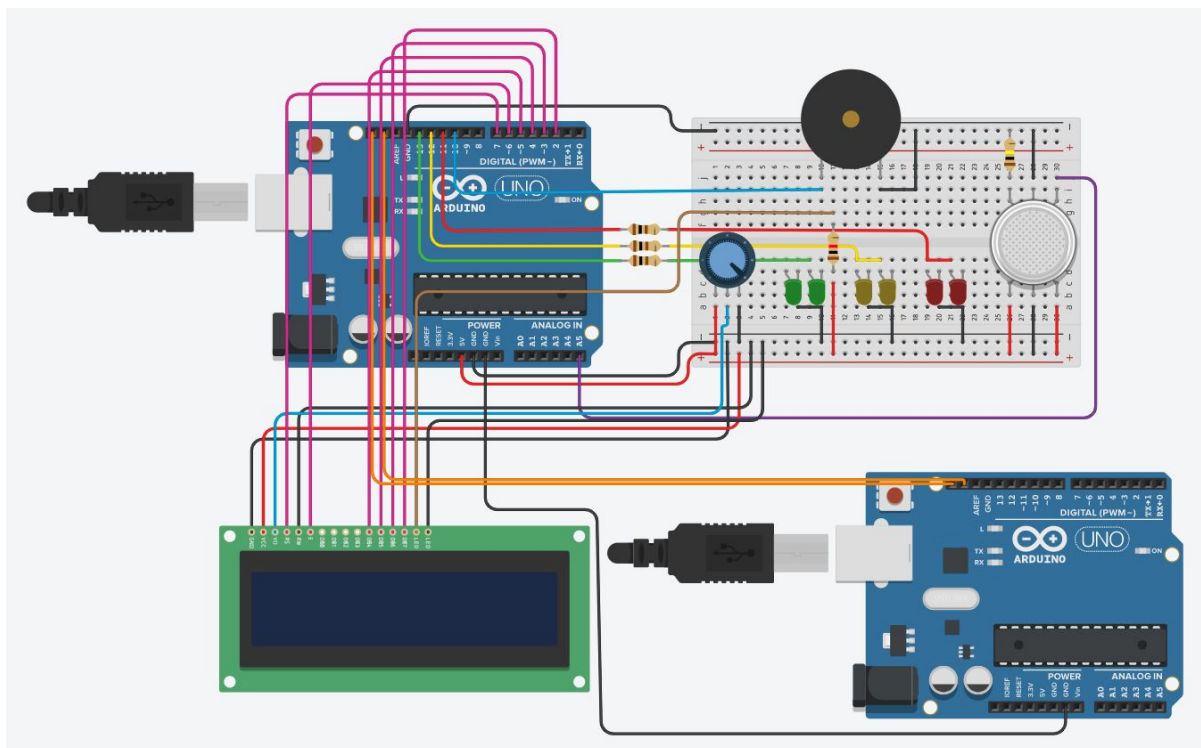
```
void setup()
{
  Wire.begin();
  Serial.begin(9600);
}
```

```
void loop()
{
  if(Serial.available()){
    char valor = Serial.read();

    if(valor == '1'){
      Wire.requestFrom(1, 31);
      Serial.println("Espere por favor..");
    }else{
      Serial.println("Ingrese un codigo adecuado");
    }
  }
}
```

```
while(Wire.available()){
  mensaje1 = Wire.read();
  Serial.print(mensaje1);
  delay(100);
}
delay(2500);
}
```

III. Esquema



IV. Problemáticas surgidas

Si bien el objetivo principal y básico de este trabajo siempre fue hacer un sistema que detecte e informe en tiempo real el nivel de gas, después de lograr esto intenté agregar más funcionalidades a modo de extensión. Fue en este punto donde surgieron algunos inconvenientes.

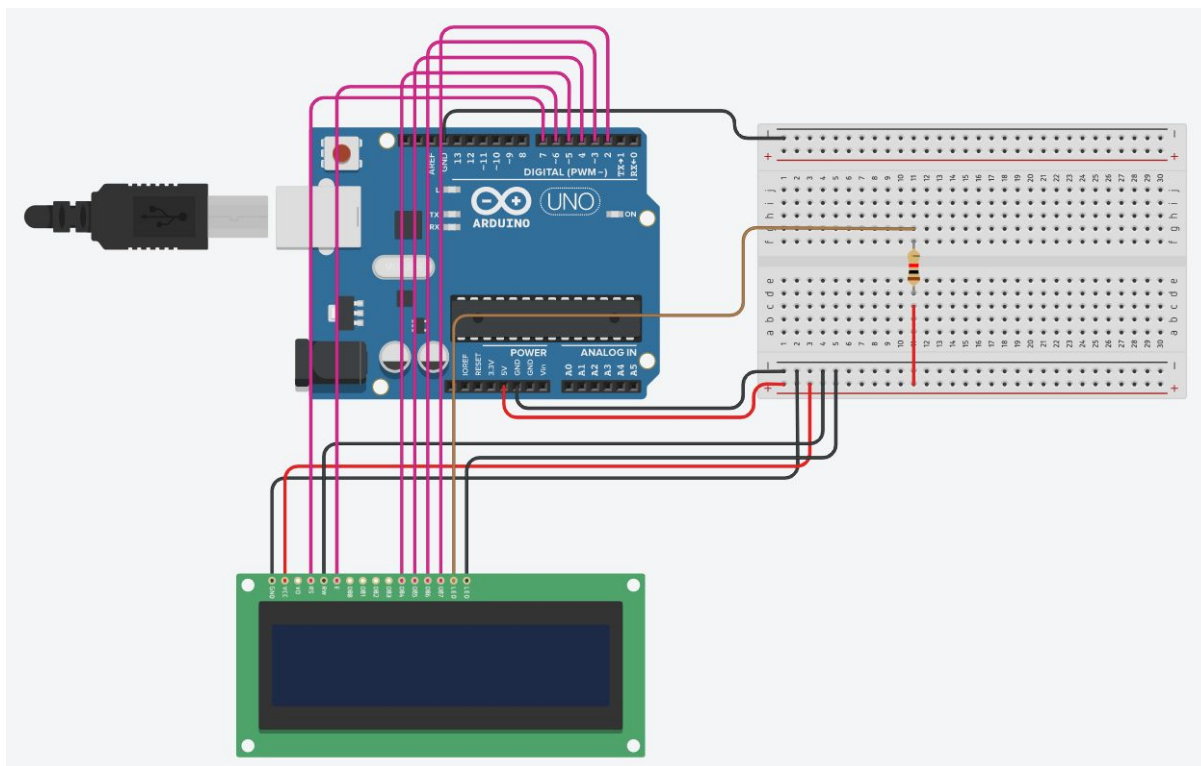
Quise incorporar un teclado el cual al detectar cierta contraseña ingresada, informe por el LCD, por ejemplo, que se estaba llamando a algún número de emergencia (simplemente se basaría en una simulación de esto). Pero el primer inconveniente que surgió es que no contaba con la suficiente cantidad de pins disponibles ya que, como se ve en el esquema, en su mayoría ya estaban siendo usados por el resto de los componentes.

Por esta situación, decidí agregar una segunda placa Arduino que se conectara al teclado. Hasta esta instancia, logré que el teclado sepa reconocer una contraseña ingresada como correcta o incorrecta. Faltaba un detalle más y era el de hacer que se conecten ambas placas Arduino entre sí ya que necesitaba que compartan información. Investigué cómo hacer esto y logré hacer la conexión entre ambas placas mediante una comunicación I2C.

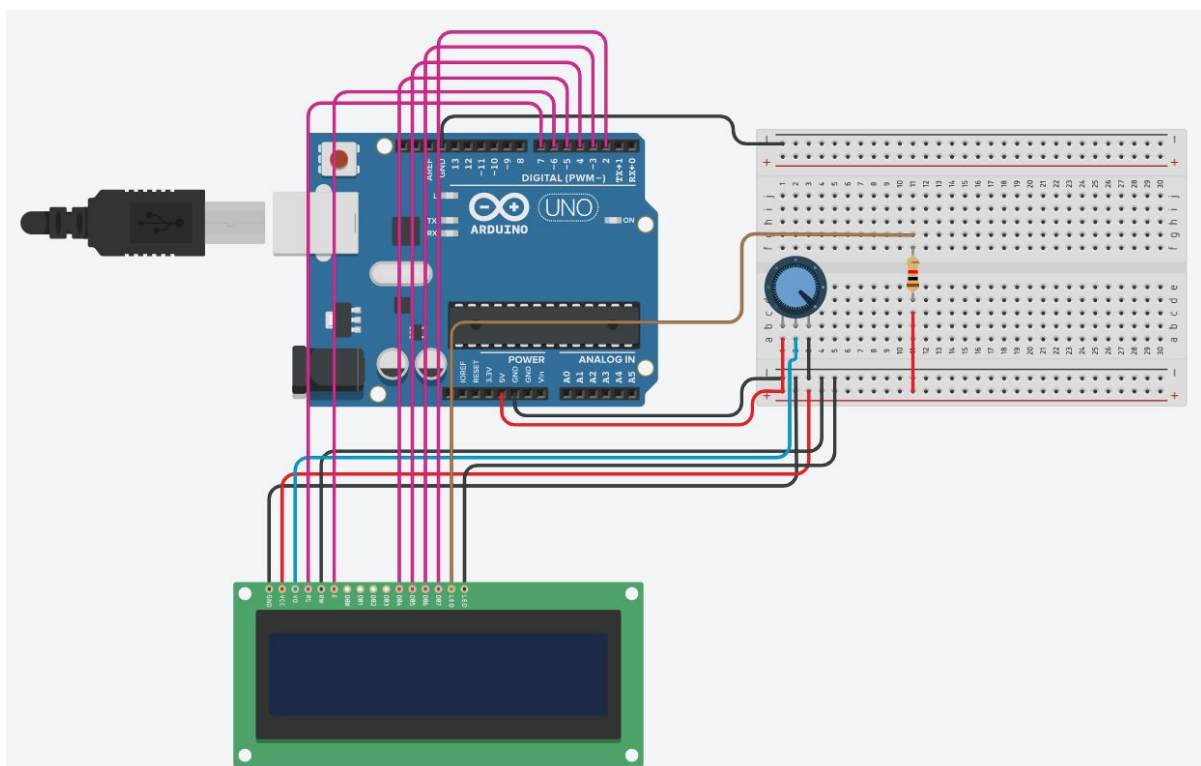
Sin embargo, el mayor inconveniente que tuve se presentó a partir de este momento ya que debía pasar de una placa a la otra la información que se recibía mediante las teclas del teclado y no lo conseguía. Por este motivo, opté por reemplazar el teclado por el ingreso de datos mediante Serial desde un Arduino y la lectura de los mismos desde el Serial del otro Arduino.. Así, finalmente logré de una forma distinta simular la situación que buscaba.

V. Proceso del armado

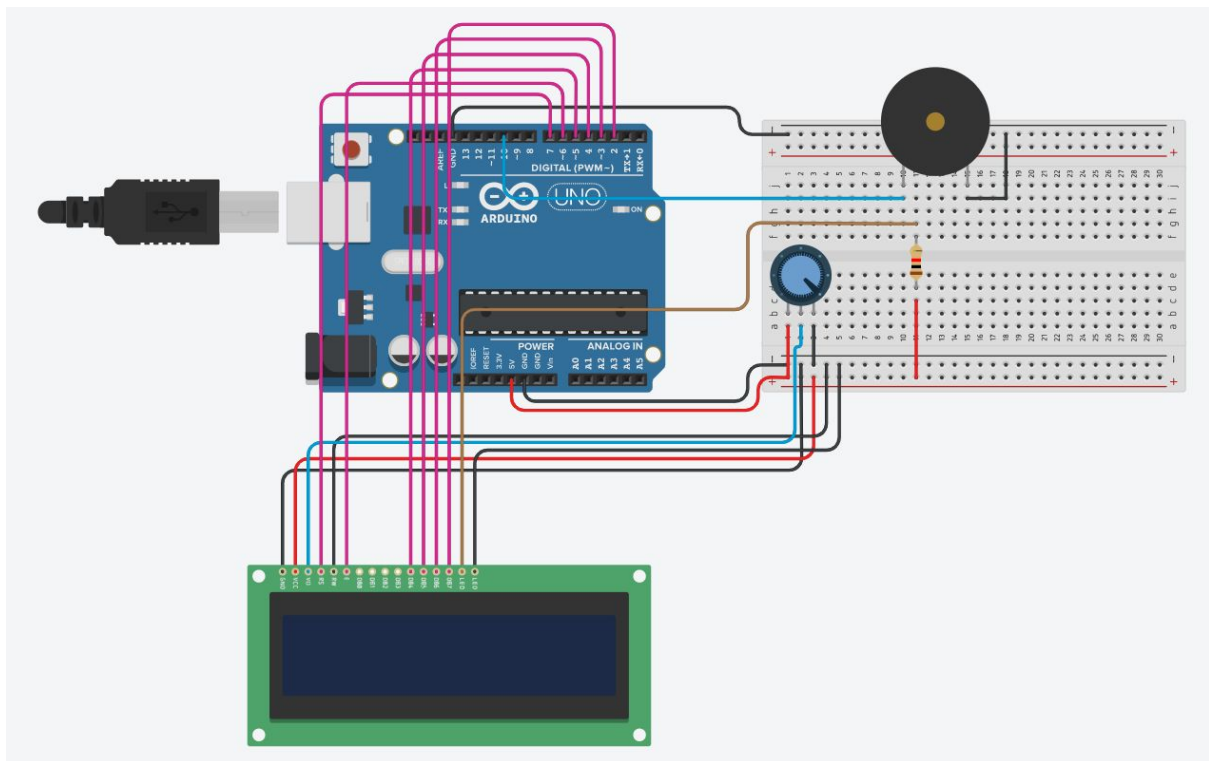
En primer lugar conecté el LCD a una de las placas Arduino y a la breadboard. Los digital pins que utilicé para esto fueron el 7, 6, 5, 4, 3, y 2. También conecté este componente y la placa Arduino a la carga negativa y positiva como correspondía. Asimismo, usé una resistencia para la conexión del LED ánodo del LCD:



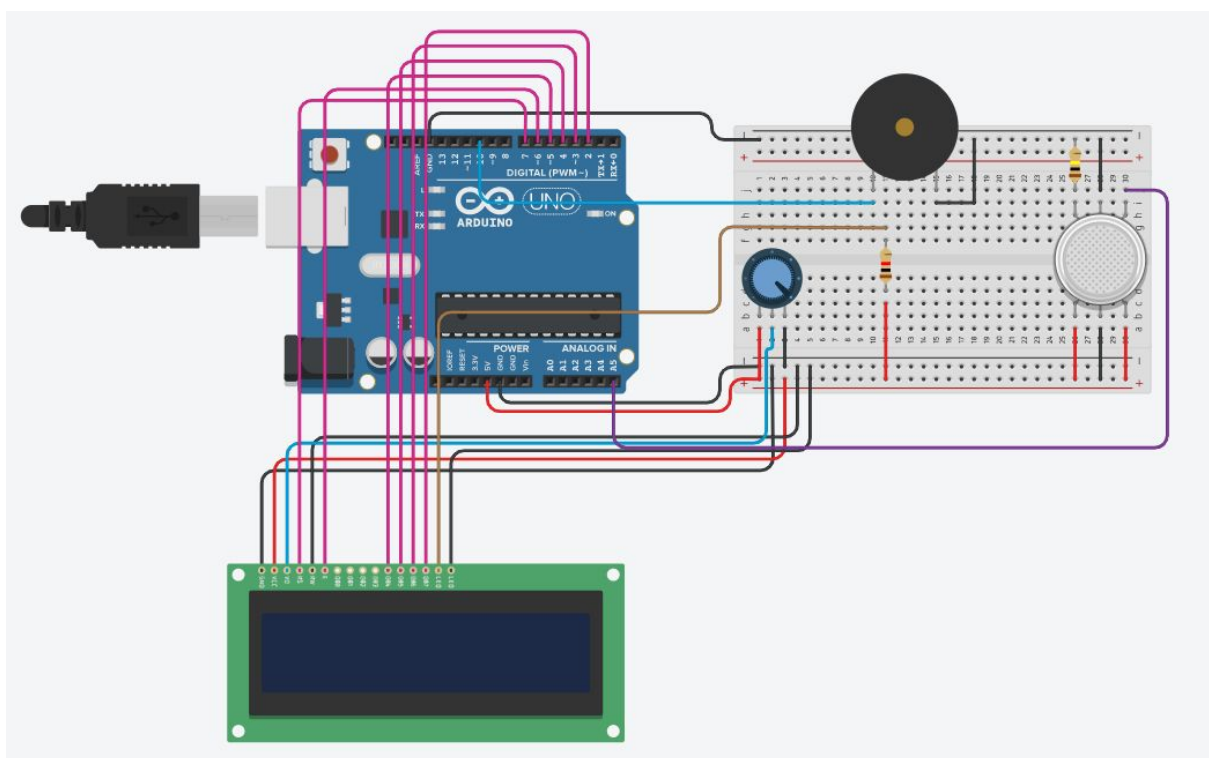
Lo siguiente que hice fue incorporar el potenciómetro para regular el paso de la energía hacia el LCD:



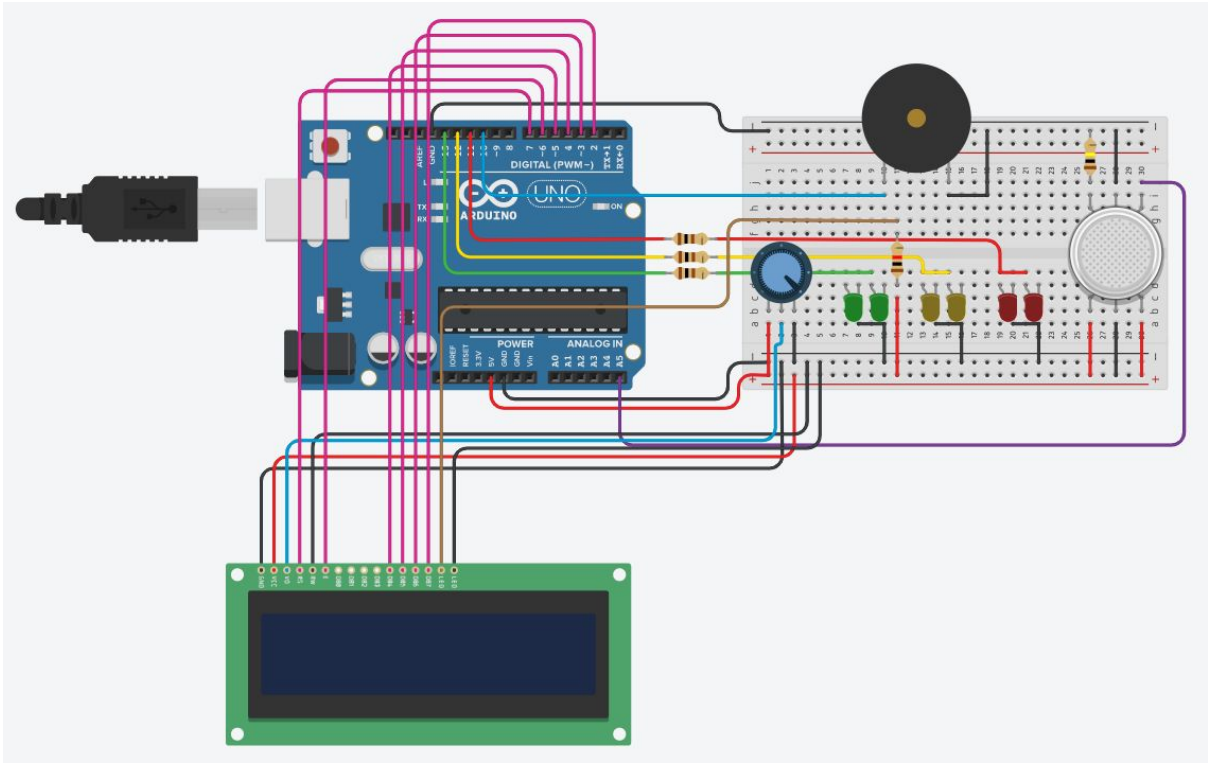
En tercer lugar incorporé el buzzer, el cual serviría para alertar que se superó cierto nivel de gas. Lo conecté el pin 10:



Finalmente agregué el componente principal de este sistema, el sensor de gas al pin A5 de la placa Arduino:



A continuación, incorporé 3 pares de led, los cuales marcaría los distintos niveles de gas. Cada uno de ellos acompañado de una resistencia de 100 Ω



Por último, agregué una segunda placa Arduino y la conecté con la ya existente. La conexión se hizo entre los puertos SCL y SDA respectivamente de ambos componentes. Además, como muestra la figura a continuación, las placas están también conectadas por GND:

