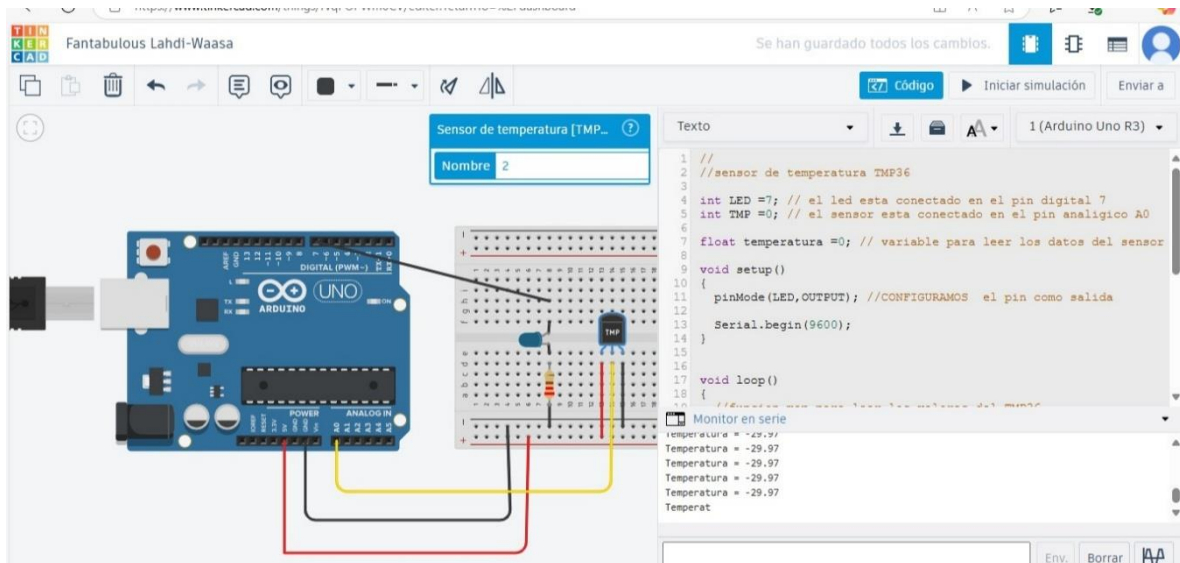


Ejercicio 13: Sensor de temperatura LM35/TMP36



SIMULACION DE QUE FUNCIONA:

El sensor TMP en Tinkercad permite simular la medición de temperatura y su respuesta en un circuito con Arduino. Puede integrarse con otros componentes, como ventiladores o alarmas, para crear sistemas automatizados de control térmico y monitoreo ambiental.

CODIGO

```
//sensor de temperatura TMP36
```

```
Int LED =7; // el led esta conectado en el pin digital 7
```

```
Int TMP =0; // el sensor esta conectado en el pin analigico A0
```

```
Float temperatura =0; // variable para leer los datos del sensor
```

```
Void setup()
{
  pinMode(LED,OUTPUT); //CONFIGURAMOS el pin como salida

  Serial.begin(9600);
}
```

```
Void loop()
{
  //funcion map para leer los valores del TMP36
  Temperatura = map(analogRead(TMP),0,1023,-5000,45000);

  Temperatura = temperatura/100.00;

  Serial.print("Temperatura = ");
  Serial.println(temperatura);

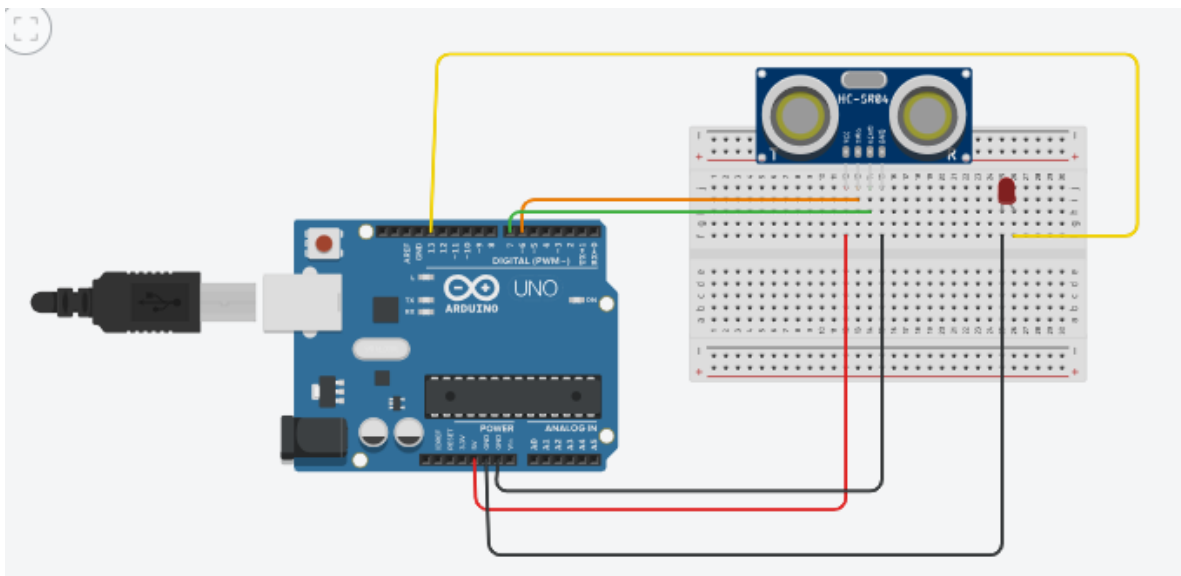
  If (temperatura>=25){ //si la temperatura es mayor que 25
    digitalWrite(LED, HIGH);
  }

  Else {
    digitalWrite(LED, LOW);
  }

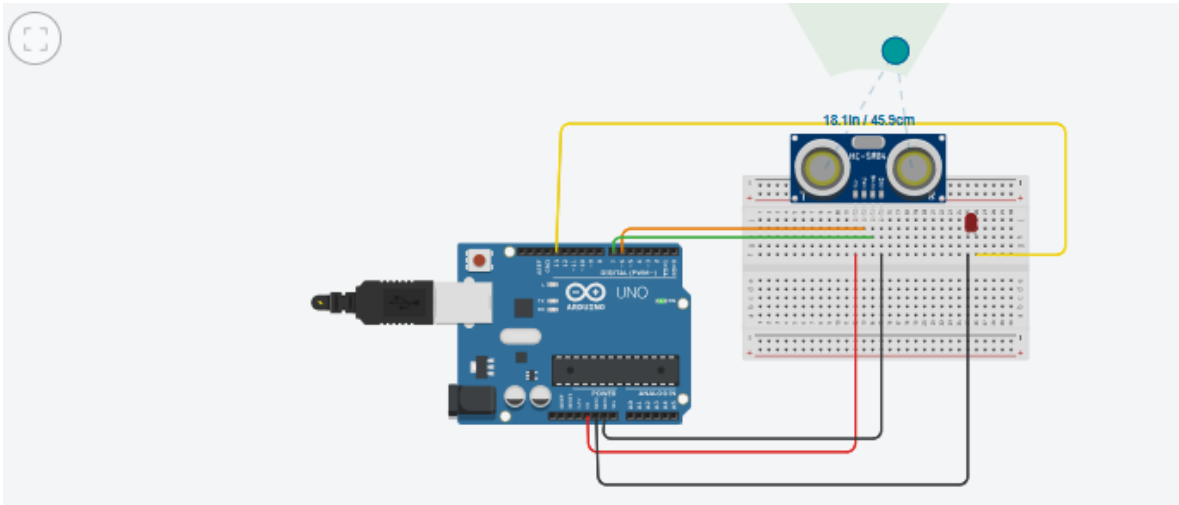
  Delay (10); // retardo de 100 milisegundos

}
```

Ejercicio 15: Sensor de ultrasonido HC-SR04 y LED (15 puntos)



SIMULACION



El sensor de ultrasonido HC-SR04 funciona emitiendo un pulso de sonido de alta frecuencia mediante su transmisor. Cuando este pulso choca con un objeto, se refleja y es captado por el receptor del sensor. En Tinkercad, se puede simular el uso del HC-SR04 con un microcontrolador como Arduino, donde se programa para leer la distancia medida y activar un LED si se cumple una condición específica (por ejemplo, si un objeto está a menos de cierta distancia). Esto es útil en aplicaciones como sistemas de alerta, robots autónomos y control de proximidad.

CODIGO

```
#define echoPin 7 // attach pin D7 Arduino to pin Echo of HC-SR04

#define trigPin 6 //attach pin D6 Arduino to pin Trig of HC-SR04


// defines variables

long duration; // variable for the duration of sound wave travel

int distance; // variable for the distance measurement (cm)

int inches; // variable for inches measurment


void setup() {

  pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT

  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT

  Serial.begin(9600); // // Serial Communication is starting with 9600 of baudrate speed

  Serial.println("Ultrasonic Sensor HC-SR04 Test"); // print some text in Serial Monitor
```

```

Serial.println("with Arduino UNO R3");

pinMode(13,OUTPUT);
}

void loop() {
    // Clears the trigPin condition
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    // Reads the echoPin, returns the sound wave travel time in microseconds
    duration = pulseIn(echoPin, HIGH);
    // Calculating the distance
    distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)
    inches = distance * 0.394;

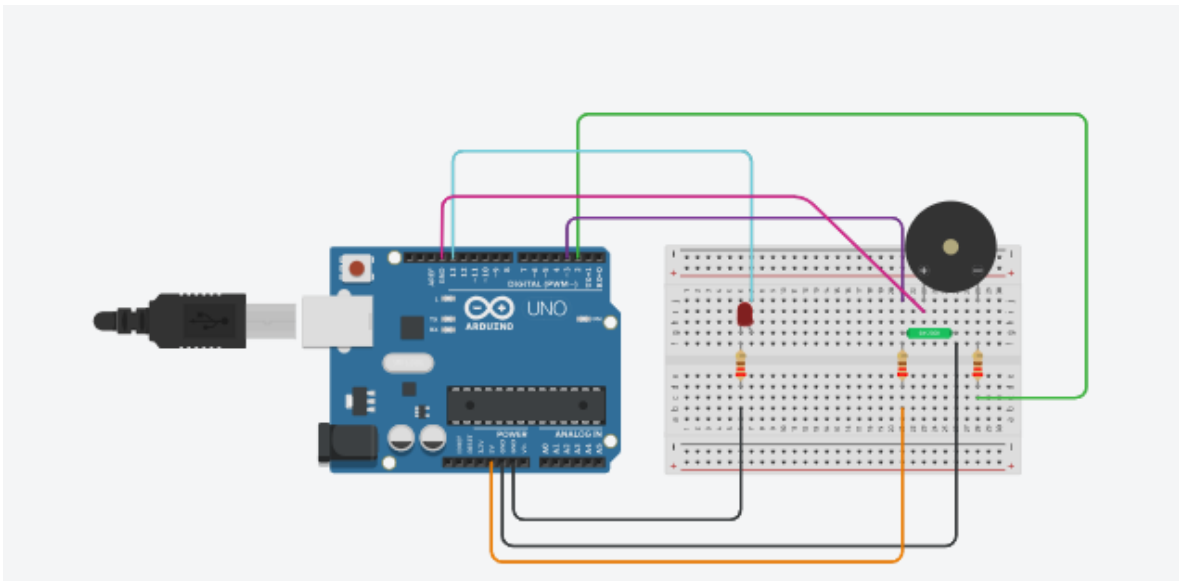
    if(inches<12){
        digitalWrite(13,HIGH);
    }
    else{
        digitalWrite(13,LOW);
    }

    // Displays the distance on the Serial Monitor
    Serial.print("Distance: ");
    Serial.print(distance);
    Serial.println(" cm");
    Serial.print("Distance: ");
    Serial.print(inches);

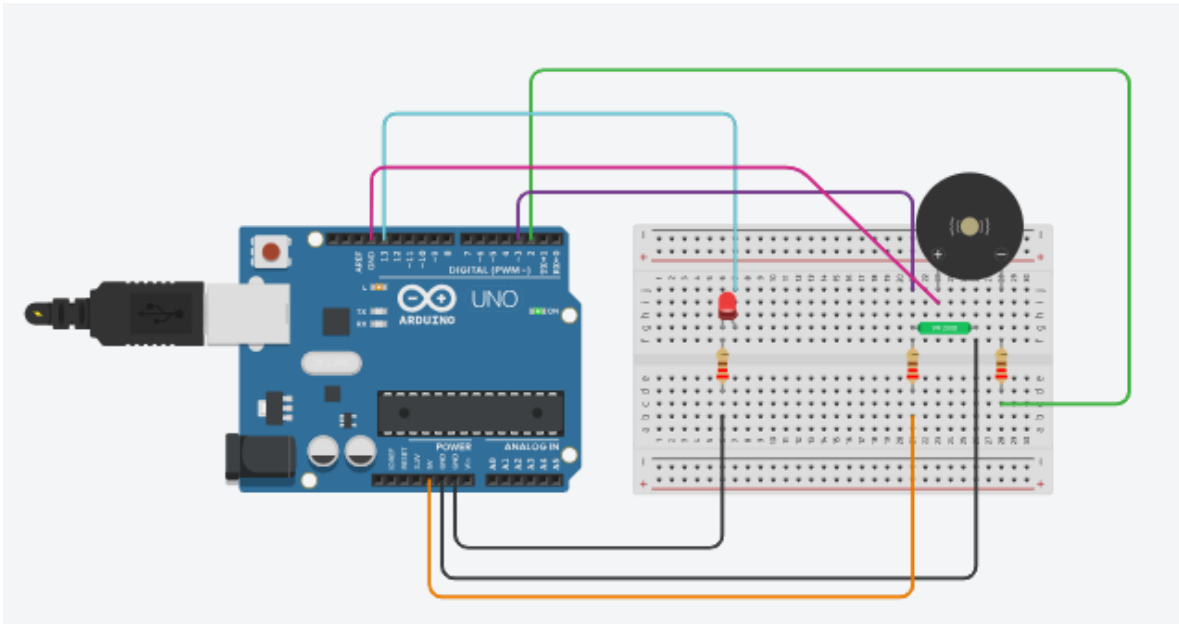
```

```
Serial.println(" inches");  
  
}
```

Ejercicio 14: Sensor de inclinación y buzzer (15 puntos)



SIMULACION DE QUE FUNCIONA



El sensor de inclinación detecta cambios en la posición mediante un mecanismo interno que cierra o abre un circuito dependiendo del ángulo. En combinación con un buzzer, se puede crear un sistema de alerta donde el buzzer emite un sonido cuando el sensor detecta una inclinación mayor a un valor predefinido, útil para alarmas de seguridad o detección de movimientos no deseados.

CODIGO

```
int Led = 13;

int Tild = 3;

int Buzz = 2;

void setup()
{
  // pinMode(LED_BUILTIN, OUTPUT);

  pinMode(Led, OUTPUT);
  pinMode(Tild, INPUT);
```

```
pinMode(Buzz, OUTPUT);
}

void loop()
{

    int Estado = digitalRead(Tild);
    if(Estado==HIGH){

        digitalWrite(Led, HIGH);

        tone(Buzz,380);
    }
    else{

        Estado=LOW;
        digitalWrite(Led, LOW);

        noTone(Buzz);
        delay(1000);

    }

}
```