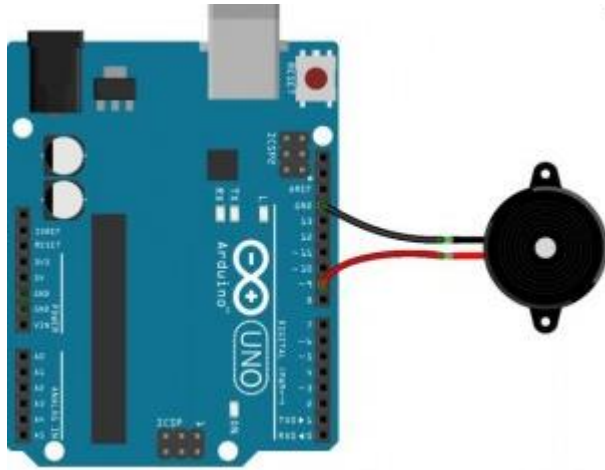


PRÁCTICA 12: ALARMA DE TEMPERATURA ACTIVADA POR UN ZUMBADOR

Para conectar un buzzer o zumbador a la placa de Arduino sólo tenemos que conectar una pata (la roja) a un pin de salida analógico y la otra (la negra) a GND. Lo único que tenemos que tener en cuenta es que los **buzzers** tienen **polaridad** (como los diodos LED), y hay que asegurarse de conectarlo correctamente.



Si conectamos las patas al revés simplemente no sonará. Con darle la vuelta ya está arreglado.

Lo primero vamos a probar que esté bien conectado. Vamos a utilizar un programilla muy simple para que suene de forma intermitente.



Y ahora podemos combinar un zumbador y cualquiera de los sensores que hemos utilizado durante estas sesiones (sensor de luz, sensor de infrarrojos, sensor de temperatura o sensor de humedad) y programar una alarma que nos avise cuando el nivel de agua, de luz o de temperatura supere un determinado valor.

Yo voy a utilizar el sensor de temperatura, de forma que si la temperatura sobrepasa un determinado límite, el buzzer empiece a sonar de la forma que hemos dicho.

Si no recordáis cómo se conecta o cómo funciona el sensor de temperatura podéis ir a la práctica anterior, que está dedicada a este sensor.

Vamos a hacer que la alarma suene de forma intermitente, y más rápido cada vez a medida que se supere más el límite que hayamos puesto.

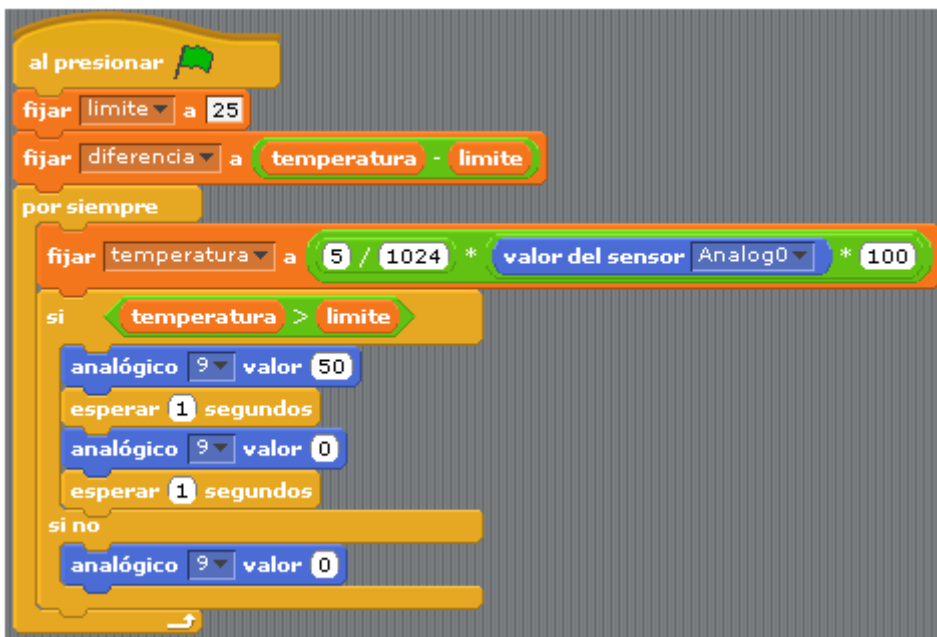
Es una buena costumbre crear variables y asignarles un valor en vez de meter directamente a mano ese valor en el programa, sobre todo si lo vamos a usar varias veces. De esa forma si queremos modificar ese valor en todo el programa

solo tendremos que cambiar la instrucción en la que asignemos el valor a la variable, en lugar de tener que cambiarlo cada vez que se usa en el programa.

Por eso vamos a crear dos variables que se llamen "límite" y "diferencia".

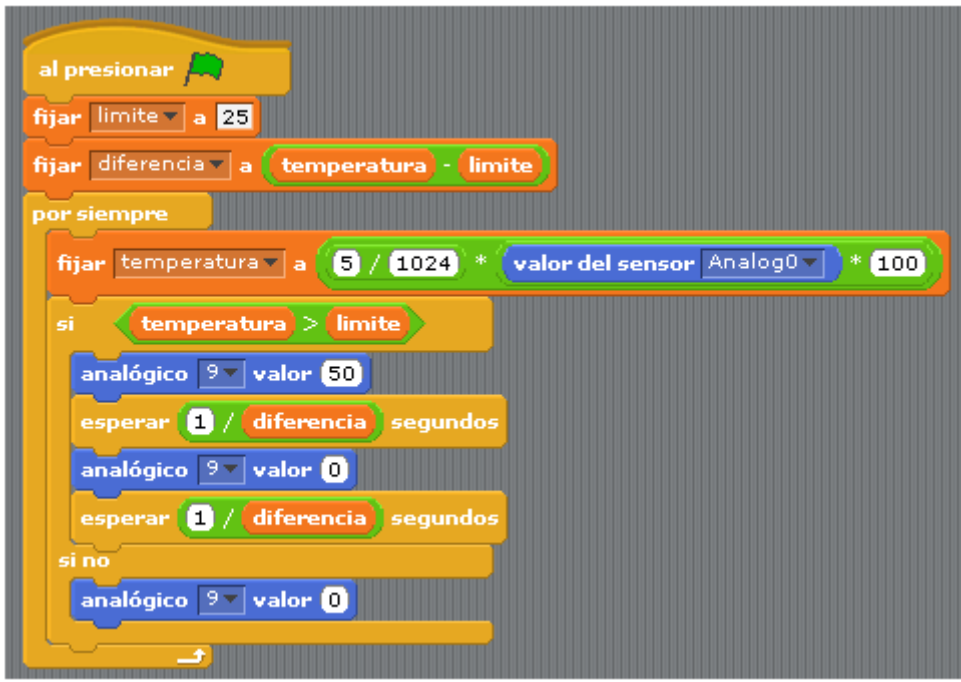


La programación para que suene el buzzer sería muy parecida a la que utilizamos en esa sesión del sensor de temperatura.



Ahora solo nos falta modificarlo para que suene cada vez con intermitencias más rápidas a medida que la temperatura vaya subiendo por encima del límite. Para ello lo que hacemos es que los tiempos de espera sean inversamente proporcionales a la diferencia de la temperatura y el límite.

- Inversamente proporcional significa que, a medida que crezca la variable que estamos midiendo (diferencia de temperatura) el valor de la que queremos controlar (el tiempo) disminuya.



The image shows a Scratch script for controlling a temperature sensor. It starts with an 'al presionar' (when clicked) event block. This is followed by two 'fijar' (set) blocks: one for 'limite' (limit) set to 25, and another for 'diferencia' (difference) set to 'temperatura - limite'. A 'por siempre' (forever) loop block contains the main logic. Inside the loop, a 'fijar' block sets 'temperatura' to the formula $\frac{5}{1024} * \text{valor del sensor Analog0} * 100$. This is followed by an 'if' block: 'si' (if) 'temperatura > limite'. If true, it executes two 'analógico' (analog) blocks: first set 'valor' to 50, then 'esperar' (wait) '1 / diferencia' seconds, then set 'valor' to 0, and wait '1 / diferencia' seconds again. If false ('si no'), it sets 'analógico' 'valor' to 0. The loop ends with a return arrow.

```
al presionar
  fijar limite a 25
  fijar diferencia a temperatura - limite
  por siempre
    fijar temperatura a  $\frac{5}{1024} * \text{valor del sensor Analog0} * 100$ 
    si temperatura > limite
      analógico 9 valor 50
      esperar 1 / diferencia segundos
      analógico 9 valor 0
      esperar 1 / diferencia segundos
    si no
      analógico 9 valor 0
```