



Arduino: ¿Qué es? ¿para qué sirve?

Primer acercamiento a la electrónica con Arduino y no quemar nada en el camino.





¿Es o no es? Esa es la cuestión...

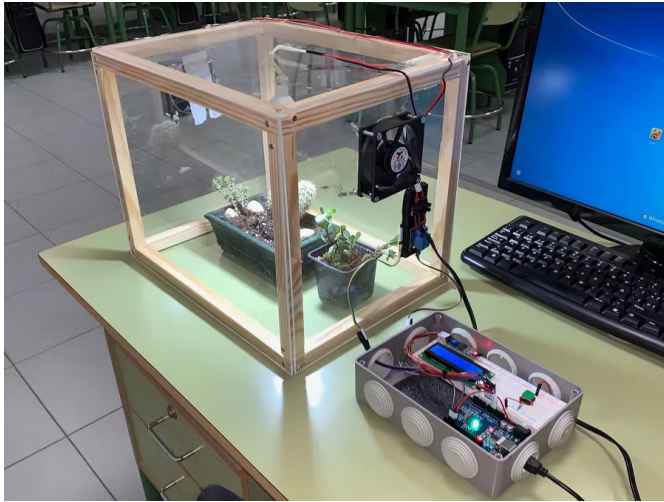
Arduino es una compañía de desarrollo de software y hardware libres, así como una comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos electrónicos, digitales interactivos que puedan detectar y controlar objetos del mundo real.

<https://www.arduino.cc/>

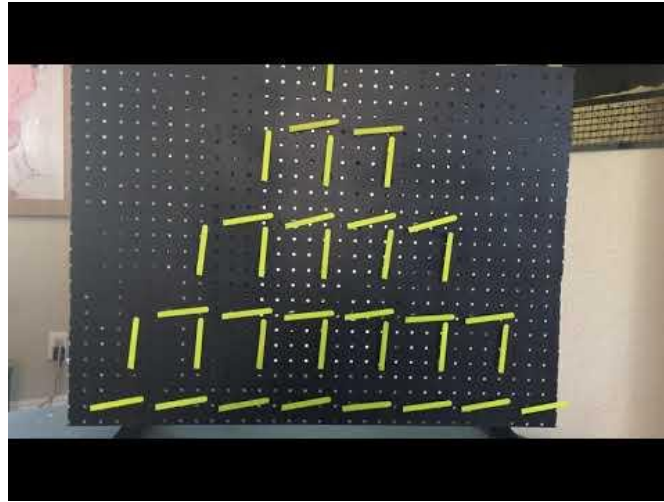
Fuente: Wikipedia uwu

Por ejemplo:

Invernaderos



Animar objetos con servo motores



Por ejemplo:

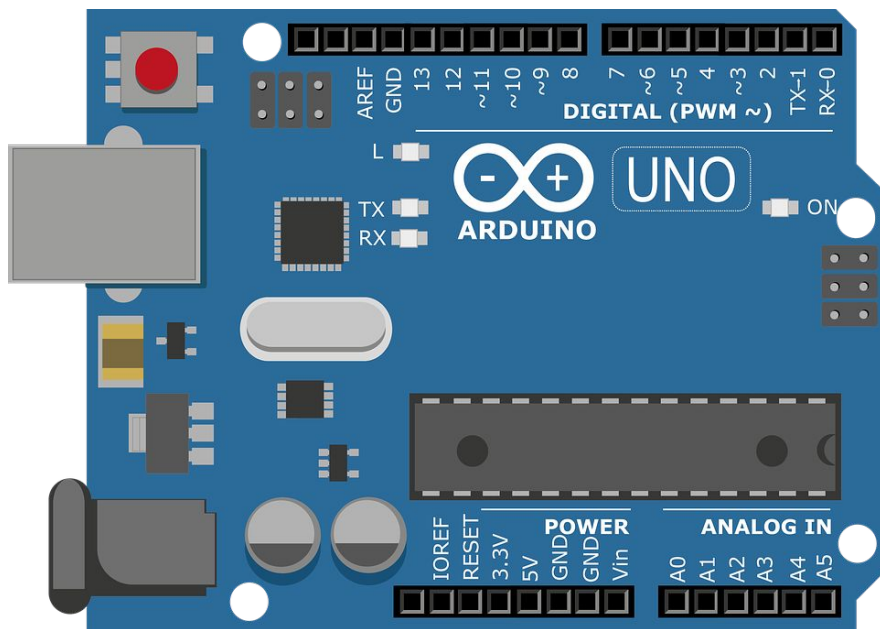
Espejos transformantes <3



Objetos de diseño



Fuente: <https://www.instructables.com/Vibrating-Pixels-Mechanical-Mirror/> | <https://www.youtube.com/watch?v=YKLJcu8xkvg>



- **USB:** transferencia de datos.

- **Plug de alimentación:** positivo al centro,  voltajes entre 9 y 12 preferentemente.

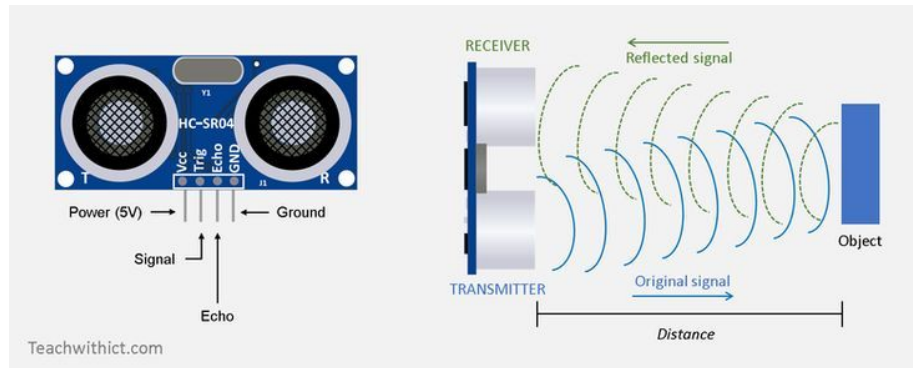
- **Regulador de voltaje:** mantiene el voltaje constante en toda la placa, se puede calentar si se sobrecarga.

- **Botón de reset:** reinicia el Arduino.

- **Chip ATMEGA328:** conviene conseguir un Arduino con chip desmontable por si se quema directamente se cambia eso.

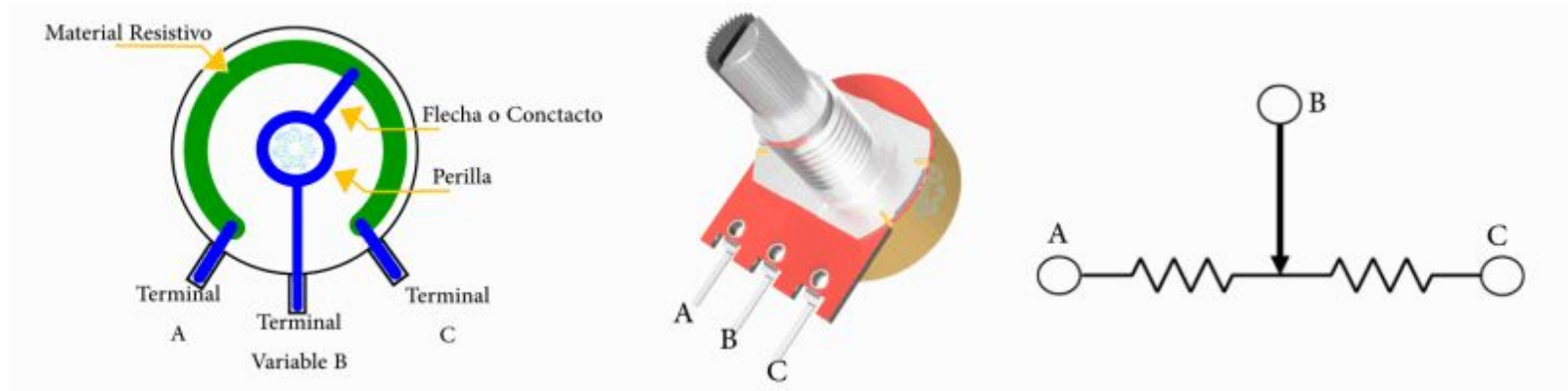
¿Cómo funciona un sensor ultrasónico?

Sensores que miden la distancia mediante el uso de ondas **ultrasónicas**. El cabezal emite una onda **ultrasónica** y recibe la onda reflejada que retorna desde el objeto. Los **sensores ultrasónicos** miden la distancia al objeto contando el tiempo entre la emisión y la recepción.



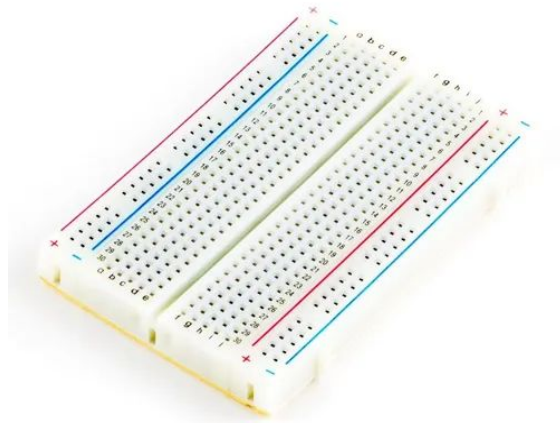
¿Cómo funciona un potenciómetro?

Un potenciómetro es una resistencia variable. Con Arduino, se puede emplear como un sensor de entrada analógica para ajustar variables como la luminosidad de una pantalla o la velocidad de un motor.



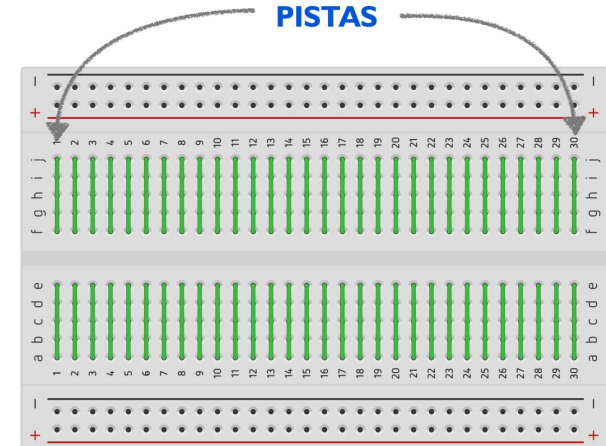
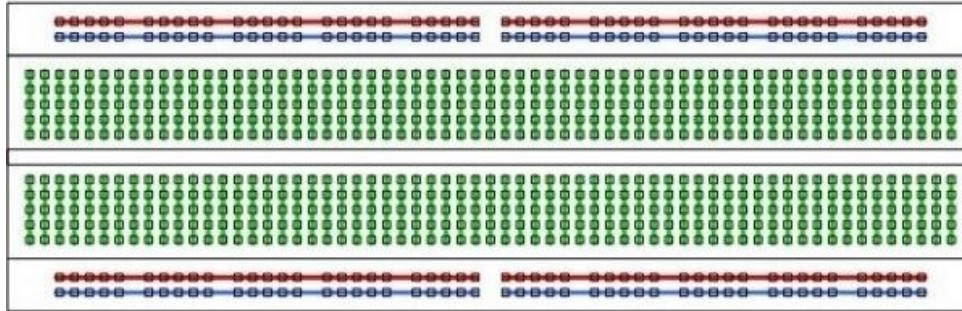
¿Cómo funciona una protoboard?

Una placa de pruebas es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna.



¿Cómo funciona una protoboard?

Una placa de pruebas es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna.





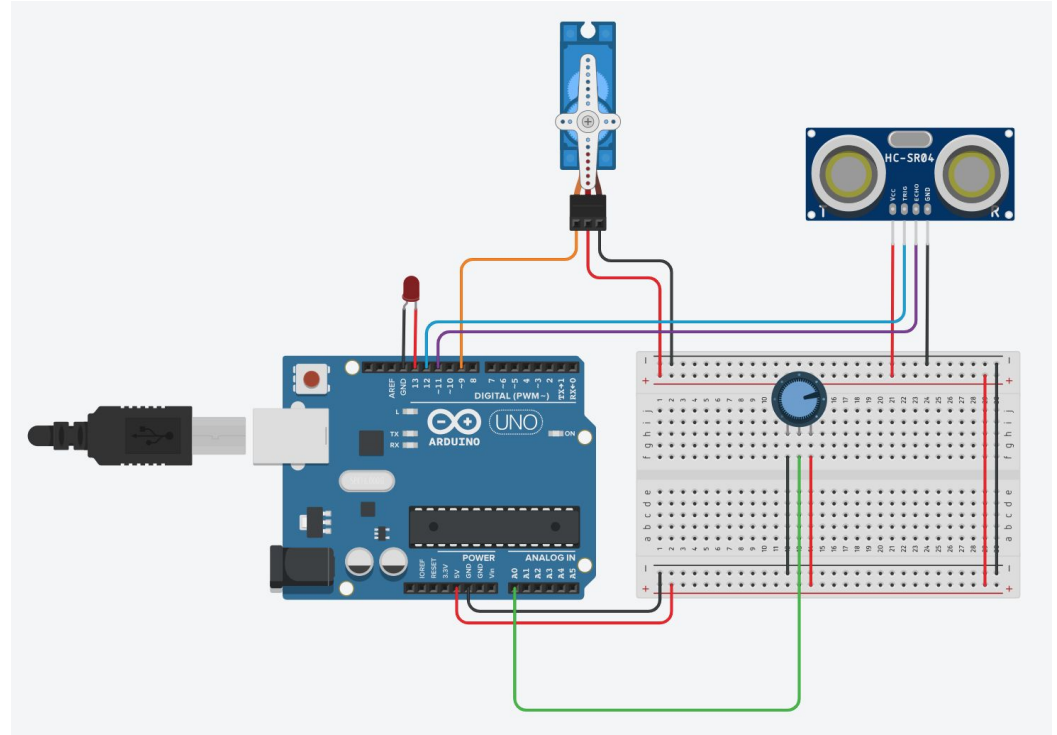
Comenzamos con los proyectos

- 1- Encender un led
- 2- Servo con ultrasónico
- 3- Servo con potenciómetro

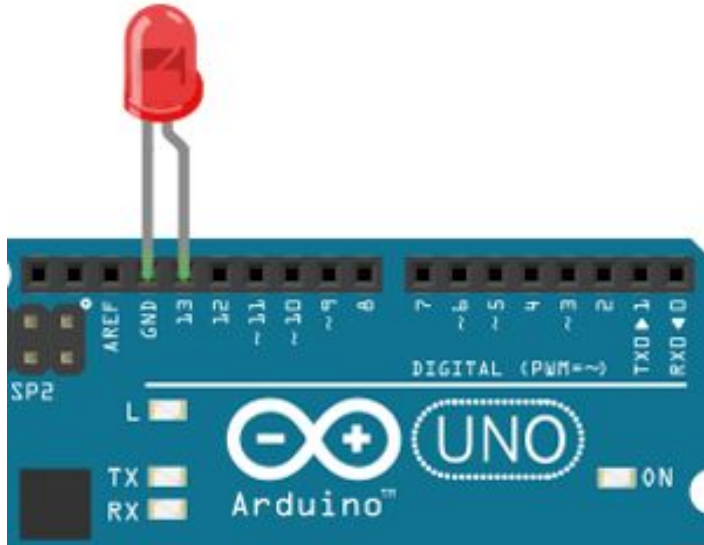
Esquema y circuito

1. Arduino + Cable USB.
2. Servo motor (los azules).
3. Ultrasónico (ojitos).
4. Potenciómetro.
5. Computadora :)
6. Protoboard (opcional).
7. Led (opcional).

Si usas protoboard, los cables deben ser macho-macho, si no usas debes tener macho-macho y macho-hembra.

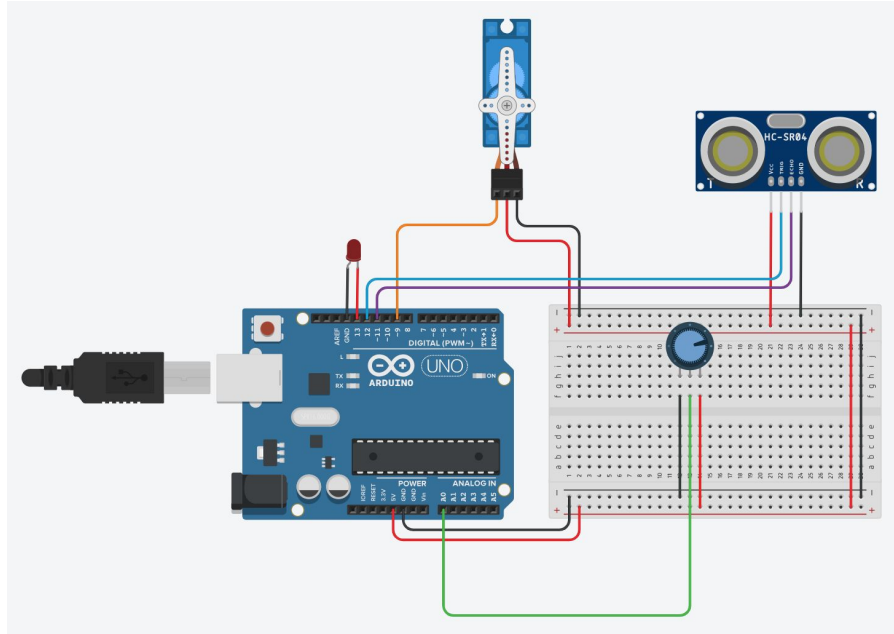


1- Encendemos el led 13



```
✓ → ⚙ Arduino Uno
encender_un_led.ino
1  const int LED = 13;
2
3  void setup()
4  {
5      pinMode(LED, OUTPUT);
6  }
7  void loop()
8  {
9      digitalWrite(LED, HIGH);
10     delay(50);
11     digitalWrite(LED, LOW);
12     delay(50);
13 }
14
```

2- Servo con ultrasónico



```
servo_con_ultrasonico §  
#include <Servo.h>  
#include <NewPing.h>  
  
const int ServoPin = 9;  
const int TriggerPin = 12;  
const int EchoPin = 11;  
const int LED = 13;  
  
// 100 = maxDistance  
NewPing sonar (TriggerPin, EchoPin, 100);  
Servo servo;  
  
void setup() {  
  Serial.begin(9600);  
  pinMode(LED, OUTPUT);  
  servo.attach(ServoPin);  
}  
  
void loop() {  
  int cm = sonar.ping_cm();  
  Serial.println(cm);  
  
  int angle = map(cm, 2, 15, 2, 150);  
  servo.write(angle);  
  if (cm < 20) {  
    digitalWrite(LED, HIGH);  
  } else {  
    digitalWrite(LED, LOW);  
  }  
}
```



Paso a paso para este proyecto

Agregamos las bibliotecas,
programamos el void setup, el void
loop, los objetos y el condicional.



Bibliotecas

Las instalamos y las llamamos.

```
#include <Servo.h>
```

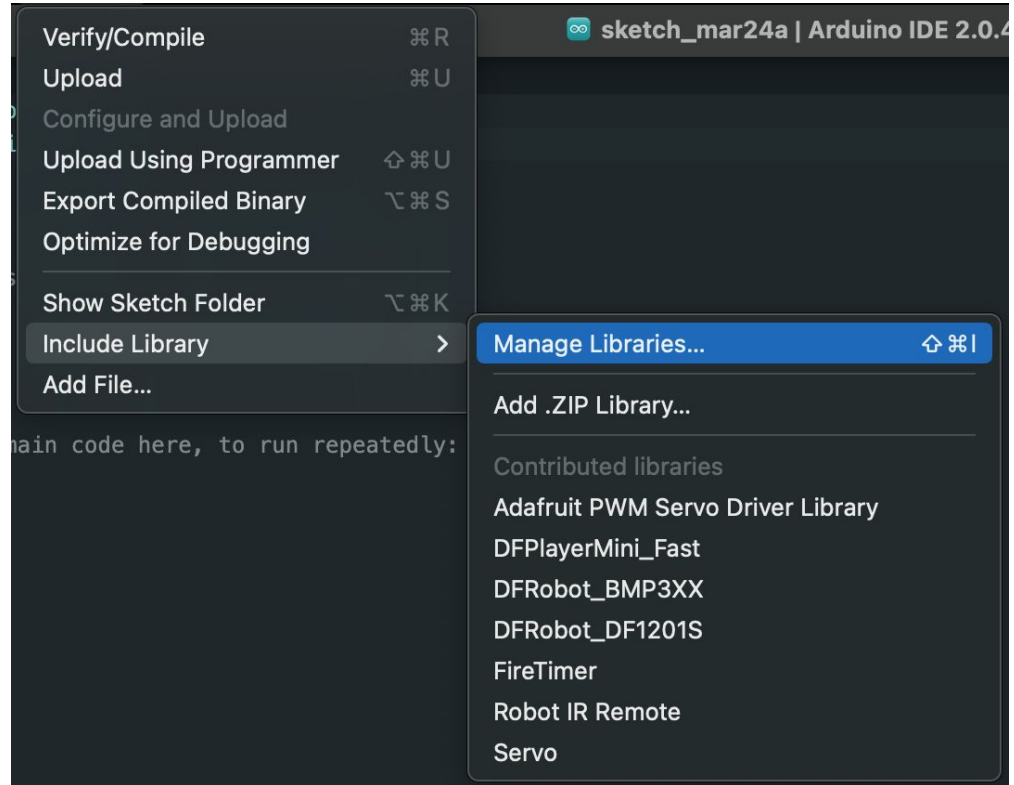
```
#include <NewPing.h>
```

sketch_mar24a.ino

```
1  #include <Servo.h>
2  #include <NewPing.h>
3
4
5  void setup() {
6      // put your setup code here, to run once:
7
8  }
9
10 void loop() {
11     // put your main code here, to run repeatedly:
12
13 }
14
```


Bibliotecas

Sketch > Include Library > Manage Libraries > Filter your search: buscamos el nombre de las librerías, en este caso Servo y NewPing.





Declaramos las constantes

Pines digitales: el 9 es el Servomotor, el 12 es el Trigger del sensor, el 11 es el Echo y el 13 es el led.

```
1  #include <Servo.h>
2  #include <NewPing.h>
3
4  const int ServoPin = 9;
5  const int TriggerPin = 12;
6  const int EchoPin = 11;
7  const int LED = 13;
8
9  void setup() {
10     // put your setup code here, to run once:
11
12 }
13
14 void loop() {
15     // put your main code here, to run repeatedly:
16
17 }
18
```

Definimos objetos

Definimos el objeto NewPing que va a contener el trigger, echo y su valor va a ser máximo 100. Definimos el objeto servo.

```
1  #include <Servo.h>
2  #include <NewPing.h>
3
4  const int ServoPin = 9;
5  const int TriggerPin = 12;
6  const int EchoPin = 11;
7  const int LED = 13;
8
9  NewPing sonar (TriggerPin, EchoPin, 100);
10 Servo servo;
11
12 void setup() {
13     Serial.begin(9600);
14     pinMode(LED, OUTPUT);
15     servo.attach(ServoPin);
16
17 }
```



Void Setup

Declaramos el puerto 9600, el output del led y el pin del servo.

Para ambos llamamos los const creados al principio.

```
1  #include <Servo.h>
2  #include <NewPing.h>
3
4  const int ServoPin = 9;
5  const int TriggerPin = 12;
6  const int EchoPin = 11;
7  const int LED = 13;
8
9  NewPing sonar (TriggerPin, EchoPin, 100);
10 Servo servo;
11
12 void setup() {
13     Serial.begin(9600);
14     pinMode(LED, OUTPUT);
15     servo.attach(ServoPin);
16
17 }
```



Void Loop

Declaramos que cm es igual al objeto sonar cuya lectura es ping_cm (es decir traducimos el valor del sensor en cm con la librería.

Hacemos un print line de la distancia del sensor.

```
18 void loop() {  
19     int cm = sonar.ping_cm();  
20     Serial.println (cm);  
21  
22     int angle = map(cm, 2, 15, 2, 150);  
23     servo.write(angle);  
24 }  
25
```



Void Loop

Declaramos el ángulo del motor a partir de un mapeo: a lo largo que el valor en centímetros varía de 2 a 15, el ángulo va de 2 a 150 grados.

Plasmamos la escritura del ángulo del servo.

```
18 void loop() {  
19     int cm = sonar.ping_cm();  
20     Serial.println (cm);  
21  
22     int angle = map(cm, 2, 15, 2, 150);  
23     servo.write(angle);  
24 }  
25
```



Void Loop

Hacemos un condicional que diga que si el valor en cm es menor a 20, se prende el led y si no, se apague.

```
18 void loop() {  
19     int cm = sonar.ping_cm();  
20     Serial.println (cm);  
21  
22     int angle = map(cm, 2, 15, 2, 150);  
23     servo.write(angle);  
24  
25     if (cm < 20){  
26         digitalWrite(LED, HIGH);  
27     } else {  
28         digitalWrite (LED, LOW);  
29     }  
30 }
```

Por último...

- 1) Verificamos el código a ver si tiene errores.
- 2) Luego conectamos la placa y la elegimos en el desplegable.
- 3) Finalmente subimos el código a la placa.

#1

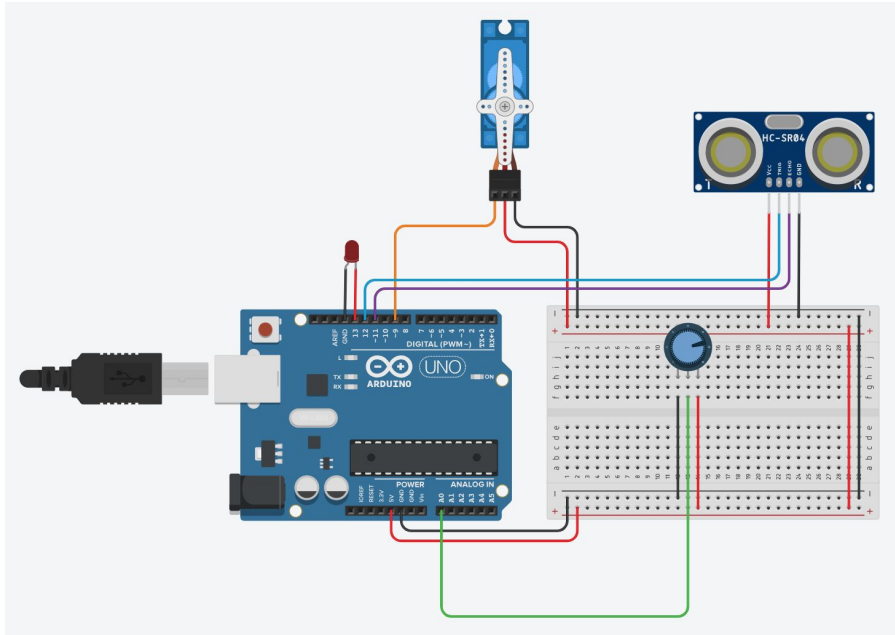
#3

#2



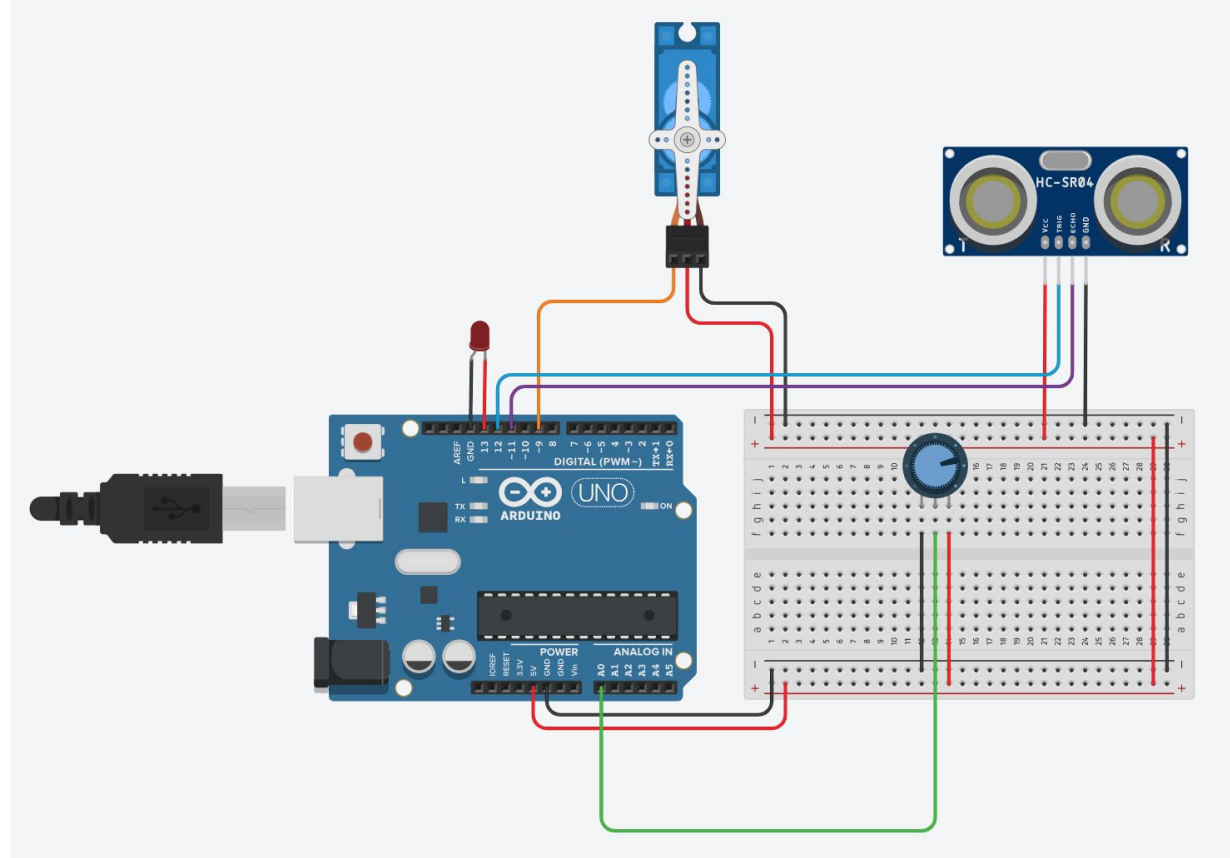
Arduino Uno

3- Servo con potenciómetro

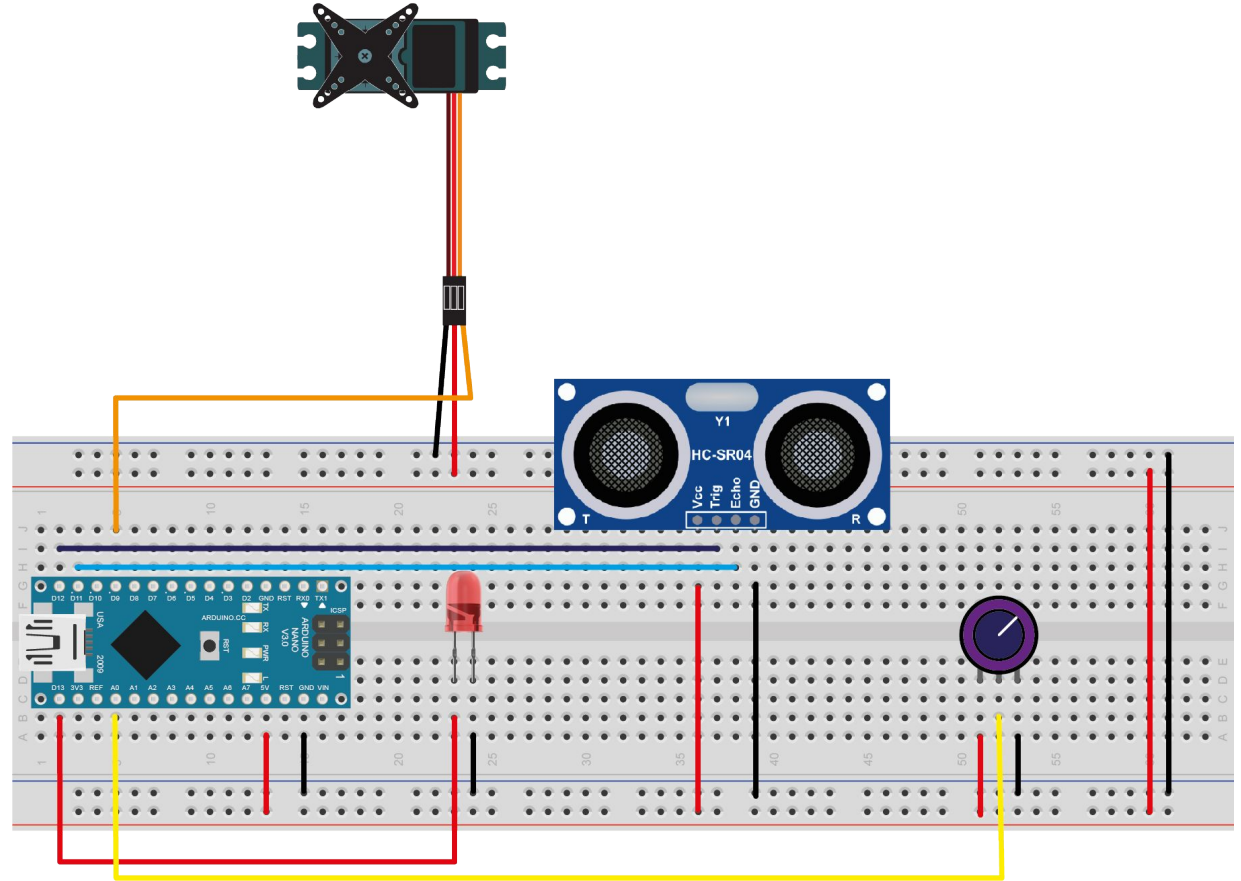


```
Arduino Uno
sketch_apr23a.ino
1  #include <Servo.h>
2  Servo servo;
3  const int LED = 13;
4
5  void setup() {
6      Serial.begin(9600);
7      pinMode(LED, OUTPUT);
8      servo.attach(9);
9  }
10
11 void loop() {
12
13     int valoranalogico = analogRead(A0);
14     int angulo = map(valoranalogico, 0, 1023, 0, 85);
15
16     servo.write(angulo);
17
18     Serial.print("Analog: ");
19     Serial.print(valoranalogico);
20     Serial.print(", Angle: ");
21     Serial.println(angulo);
22     delay(100);
23
24     if (angulo == 0){
25         digitalWrite(LED, HIGH);
26     } else {
27         digitalWrite(LED, LOW);
28     }
29 }
```

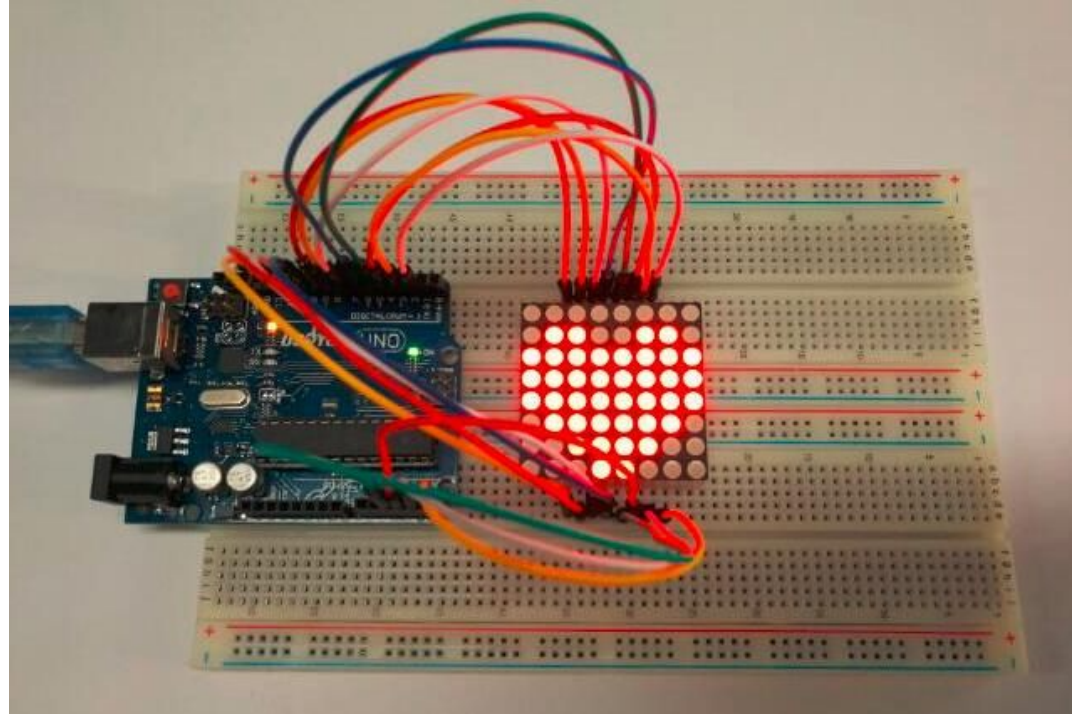
Circuito con Arduino UNO



Circuito con Arduino NANO

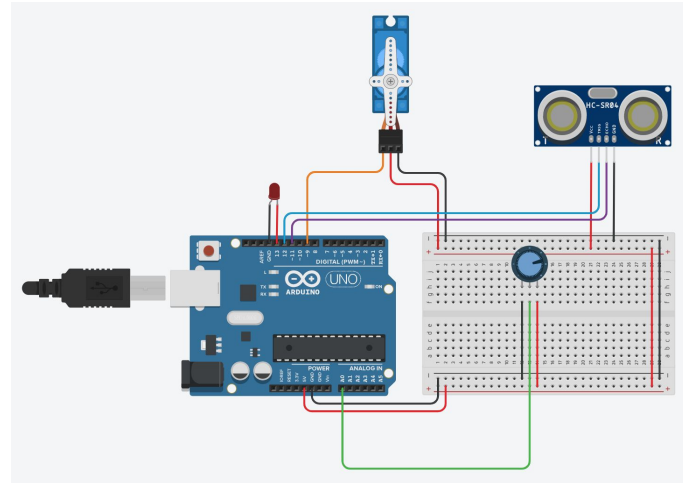
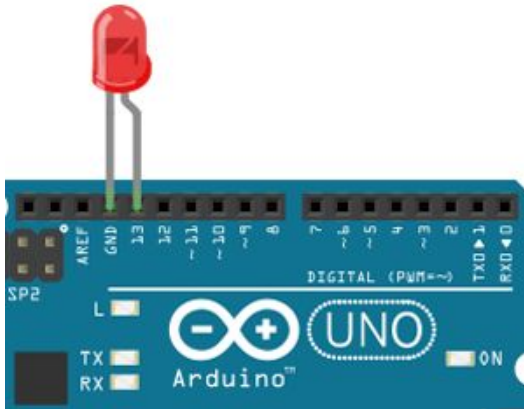



¡GRACIAS!



Empezamos con nuestro Hola Mundo!

Entendiendo esto vamos a hacer en este campo va a ser **encender y controlar el led 13**, seguido de **controlar un servo motor con un sensor ultrasónico y un potenciómetro**.



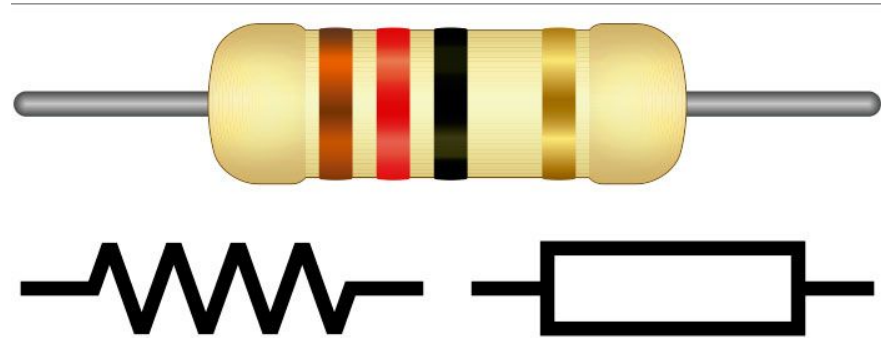
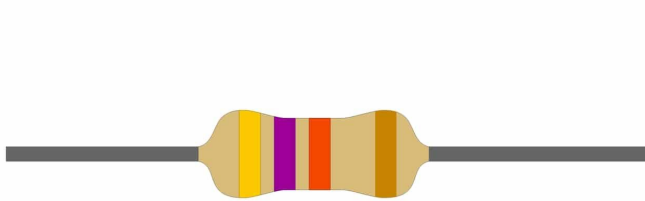


Continuamos con los pines analógicos

Como tercer experimento vamos a controlar un servo con un potenciómetro convencional, y vamos a simular un switcheo entre un sistema u otro. **Lo vamos a hacer a partir de leds y resistencias :)**

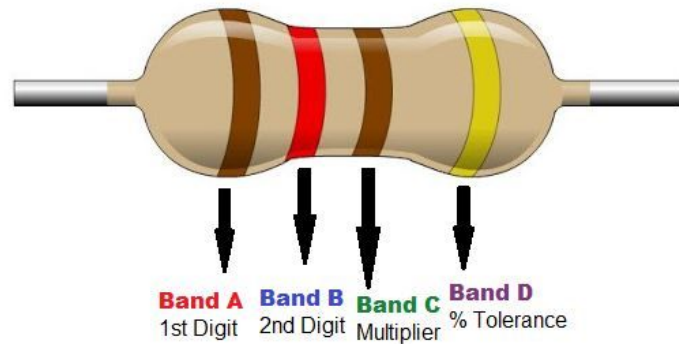
Pero primero: ¿Potenciómetro? ¿Resistencias?

RESISTENCIAS: Una resistencia se puede definir como cualquier medio material que limita el paso de la corriente eléctrica. **Provocan una restricción al paso de la corriente**, limitándola y regulándola.



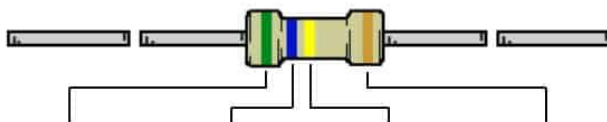
Pero primero: ¿Potenciómetro? ¿Resistencias?

RESISTENCIAS: Las resistencias poseen anillos de colores que indican su valor en Ohms. Si entramos a la App Store de nuestro celular y escribimos “Calculadora de resistencias” o sus derivados podemos encontrar aplicativos que nos ayudan a saber qué valor tienen las resistencias que adquirimos. También hay aplicaciones online.

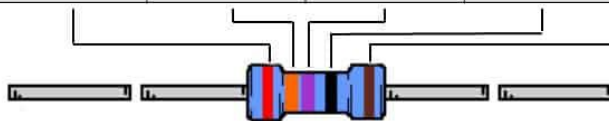




4 Band Code



COLOR	1 st BAND	2 nd BAND	3 rd BAND	MULTIPLIER	TOLERANCE
BLACK	0	0	0	1 Ω	
BROWN	1	1	1	10 Ω	$\pm 1\%$ (F)
RED	2	2	2	100 Ω	$\pm 2\%$ (G)
ORANGE	3	3	3	1K Ω	
YELLOW	4	4	4	10K Ω	
GREEN	5	5	5	100K Ω	$\pm 0.5\%$ (D)
BLUE	6	6	6	1M Ω	$\pm 0.25\%$ (C)
VIOLET	7	7	7	10M Ω	$\pm 0.10\%$ (B)
GREY	8	8	8	100M Ω	$\pm 0.05\%$
WHITE	9	9	9	1G Ω	
GOLD				0.1 Ω	$\pm 5\%$ (J)
SILVER				0.01 Ω	$\pm 10\%$ (K)



5 Band Code

Pero primero: ¿Potenciómetro? ¿Resistencias?

POTENCIÓMETRO: Un potenciómetro es una Resistencia Variable. Los potenciómetros limitan el paso de la corriente eléctrica provocando una caída de tensión en ellos al igual que en una resistencia, pero en este caso el valor de la corriente y la tensión en el potenciómetro las podemos variar solo con cambiar el valor de su resistencia.

