



TRABAJO INTEGRADOR

2DO QUIMESTRE

Nombres: Eric Jacob

Apellidos: Zuberbühler León

Curso: 6to informática

Docente: Henry Muñoz

2022 - 2023

INDICE

ARDUINO.....	3
TINKERCARD	7
EJERCICIOS SIMULADOR TINKERCARD ...	10
ANEXOS COMPONENTES.....	70

1. ARDUINO

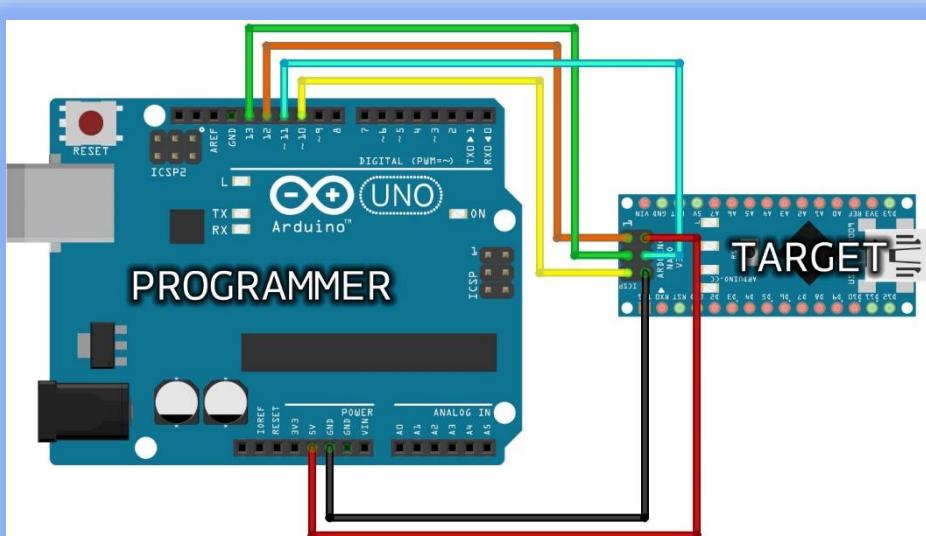
¿QUE ES ARDUINO?

Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso.



¿Cómo funciona?

Para poder entender este concepto, primero vas a tener que entender los conceptos de hardware libre y el software libre. El hardware libre son los dispositivos cuyas especificaciones y diagramas son de acceso público, de manera que cualquiera puede replicarlos. Esto quiere decir que Arduino ofrece las bases para que cualquier otra persona o empresa pueda crear sus propias placas, pudiendo ser diferentes entre ellas, pero igualmente funcionales al partir de la misma base.



TIPOS DE TARJETAS

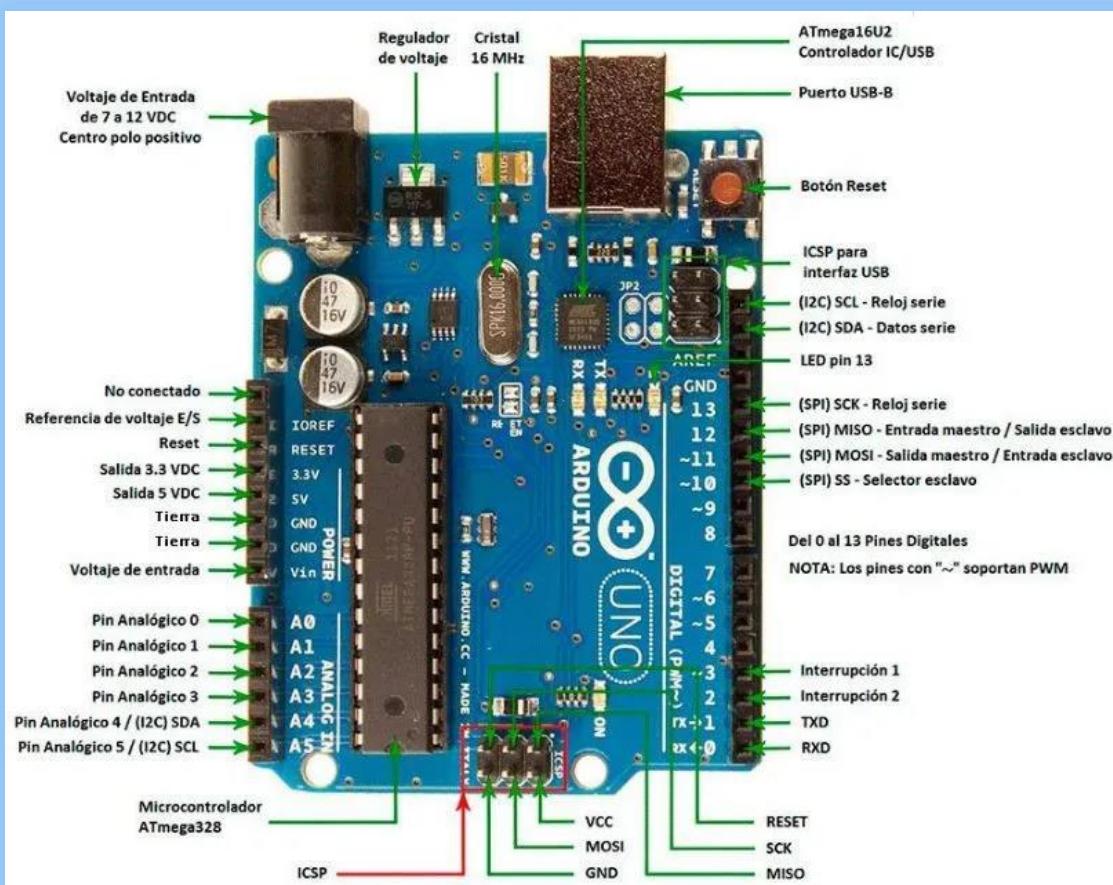


ARDUINO UNO

Arduino UNO es una placa basada en el microcontrolador ATmega328P. Tiene 14 pines de entrada/salida digital (de los cuales 6 pueden ser usando con PWM), 6 entradas analógicas, un cristal de 16Mhz, conexión USB, conector jack de alimentación, terminales para conexión ICSP y un botón de reseteo. Tiene toda la electrónica necesaria para que el microcontrolador opere, simplemente hay que conectarlo a la energía por el puerto USB o con un transformador AC-DC.



PARTES DE LA TARJETA

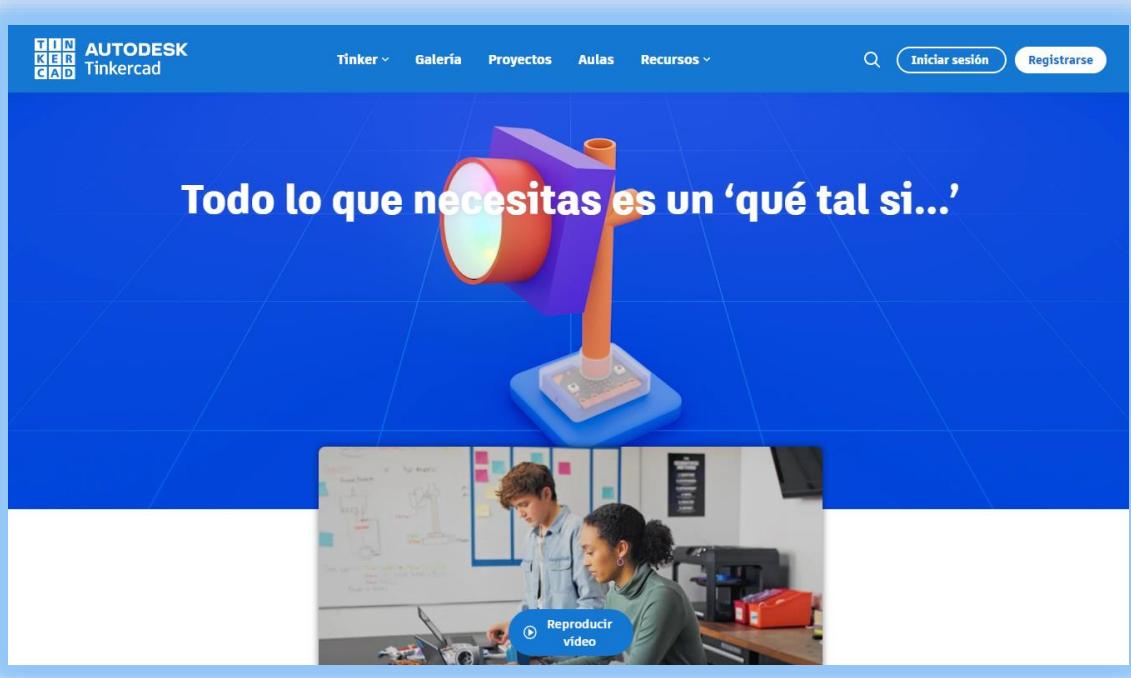


2. TINKERCARD

¿Qué es Tinkercard?

Tinkercad es un software gratuito de diseño y modelado 3D que encanta a todos por su facilidad de uso.

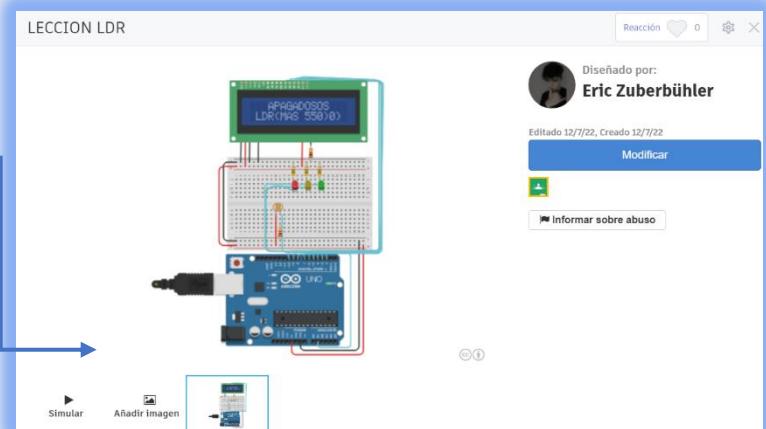
Permite diseñar cualquier objeto con volumen de forma intuitiva y después llevarlo a la realidad mediante una impresora 3D. Por ello hoy en día Tinkercad es la opción favorita de millones de Makers para fabricar todo tipo de objetos según sus propias ideas y en su propia casa.



Herramientas

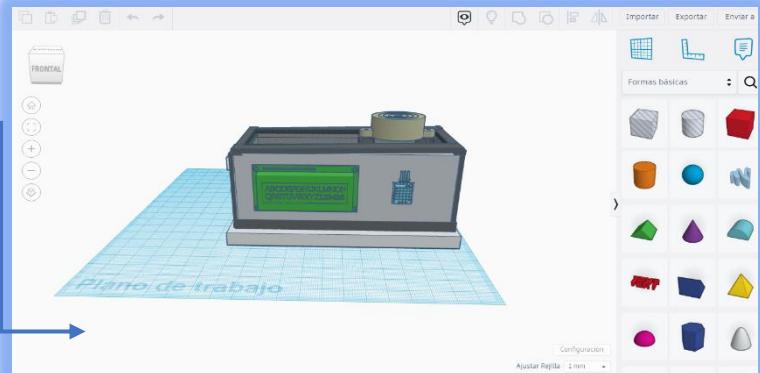
CIRCUITO

Con este simulador podemos aprender a utilizar estos componentes electrónicos.



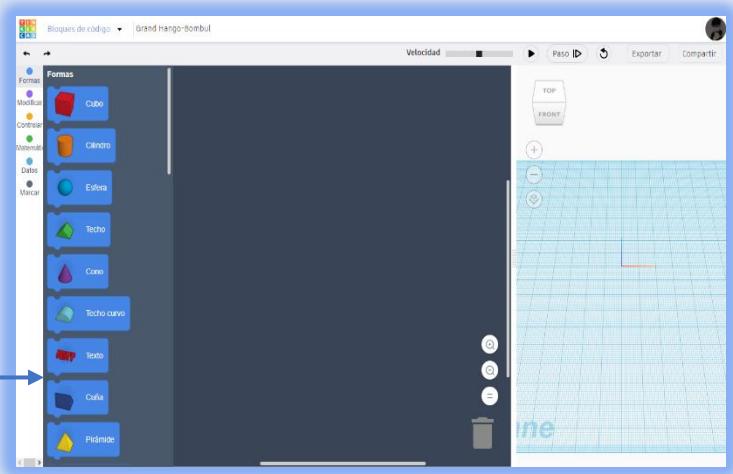
DISEÑO 3D

Estos proyectos para principiantes podemos realizar diseños en 3D de muchas cosas con muchos objetos diferentes,



CODIGO DE BLOQUES

Haga una canasta imprimible en 3D usando los bloques crear variable y cuenta con bloques.



3. EJERCICIOS

SIMULADOR

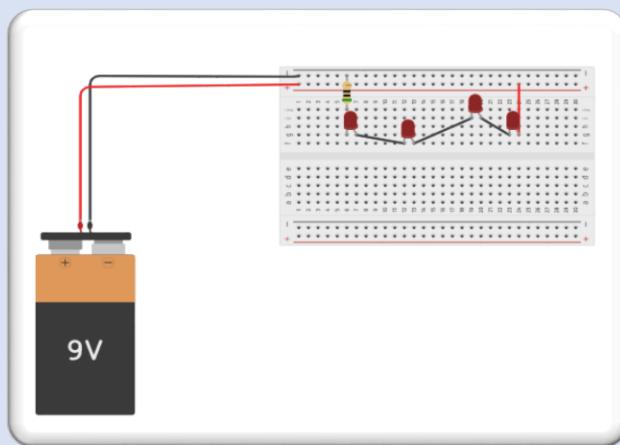
TINKERCARD

ARDUINO

Ejercicio # 1

UNA PILA DE 9V CONECTADA EN UN PROTOBOARD CON UN LED CON SU RESISTENCIA.

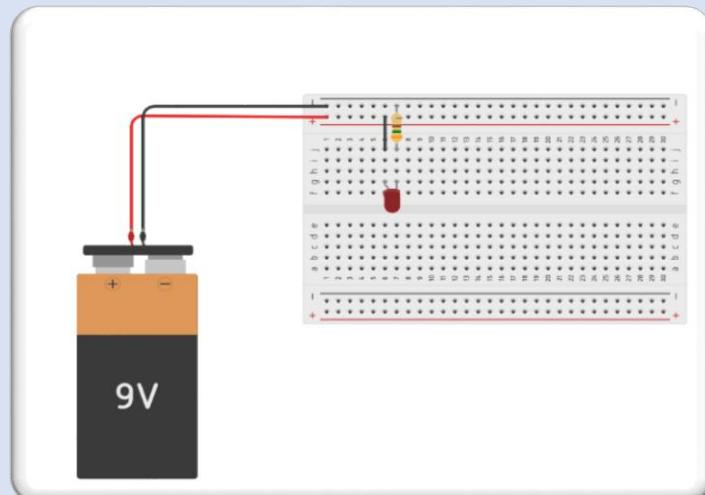
- DIAGRAMA



Ejercicio # 2

UNA PILA DE 9V CONECTADA EN UN PROTOBOARD CON 4 LEDs EN SERIES CON UNA RESISTENCIA.

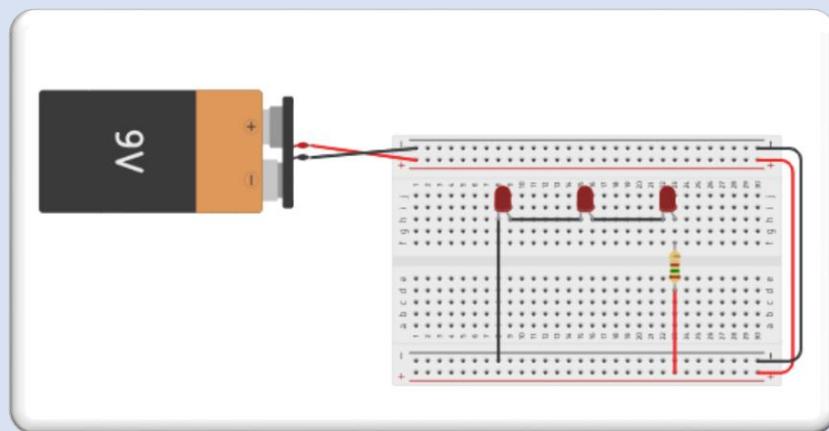
- DIAGRAMA



Ejercicio # 3

UNA PILA DE 9V CONECTADA EN UN PROTOBOARD CON 3 LEDS
CONECTADOS EN SERIE CON UNA RESISTENCIA DE 150
OHMIOS.

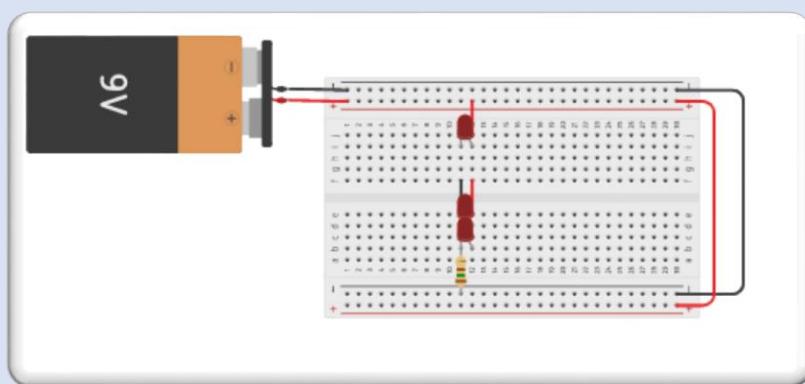
- DIAGRAMA



Ejercicio # 4

UNA PILA DE 9V CONECTADA EN UN PROTOBOARD CON 3 LEDS
CONECTADOS EN PARALELO CON UNA RESISTENCIA DE 150
OHMIOS.

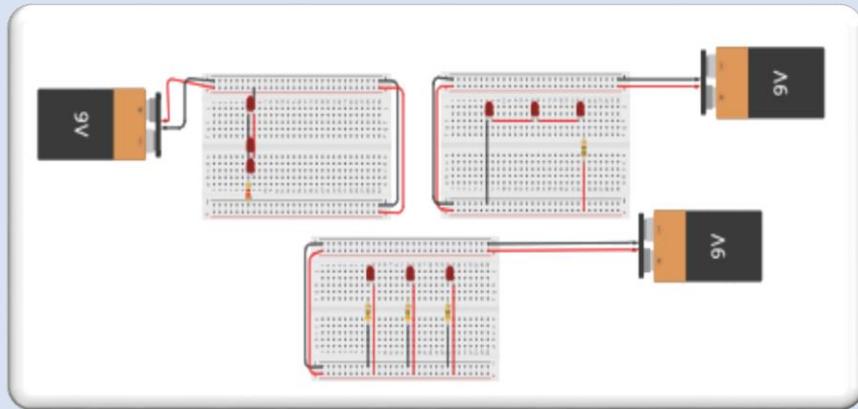
- DIAGRAMA



Ejercicio # 5

3 PILAS DE 9V CONECTADAS EN PROTOBOARDS INDIVIDUALES CON LEDS CONECTADOS EN SERIE, PARALELO E INDIVIDUAL.

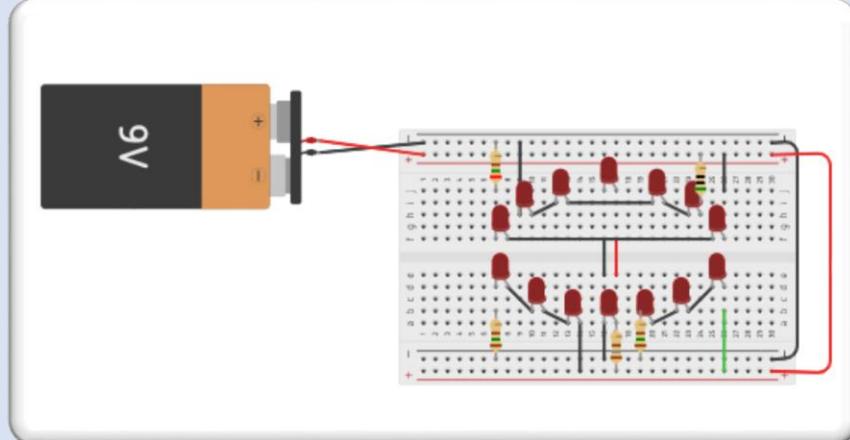
- **DIAGRAMA**



Ejercicio # 6

UNA PILA DE 9V CONECTADA EN PROTOBOARD CON LEDS FORMANDO UN CIRCULOS CON SUS RESPECTIVAS RESISTENCIAS.

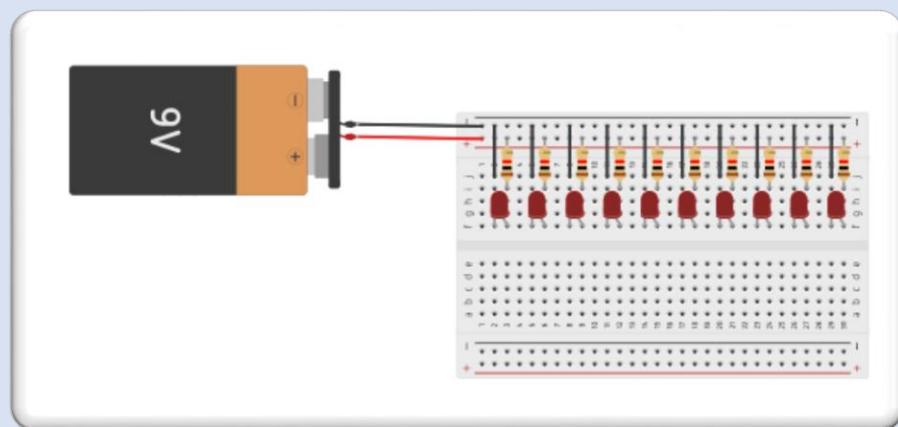
- **DIAGRAMA**



Ejercicio # 7

UNA PILA DE 9V CONECTADA CON UN PROTOBOARD Y LEDS CON SUS RESISTENCIAS.

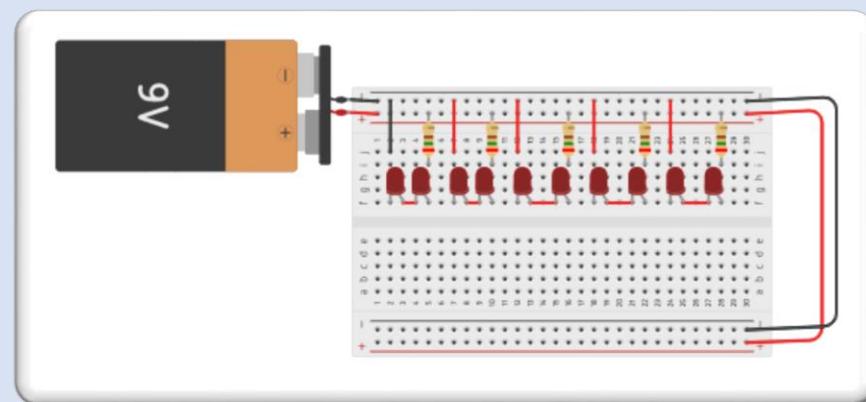
- DIAGRAMA



Ejercicio # 8

UNA PILA DE 9V CONECTADA CON UN PROTOBOARD Y LEDS EN SERIE CON SUS RESISTENCIAS.

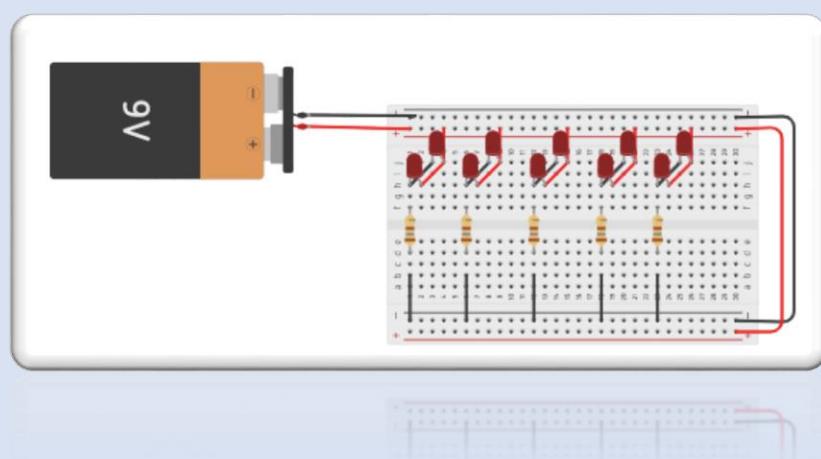
- DIAGRAMA



Ejercicio # 9

UNA PILA DE 9V CONECTADA CON UN PROTOBOARD Y LEDS EN PARALELO CON SUS RESISTENCIAS.

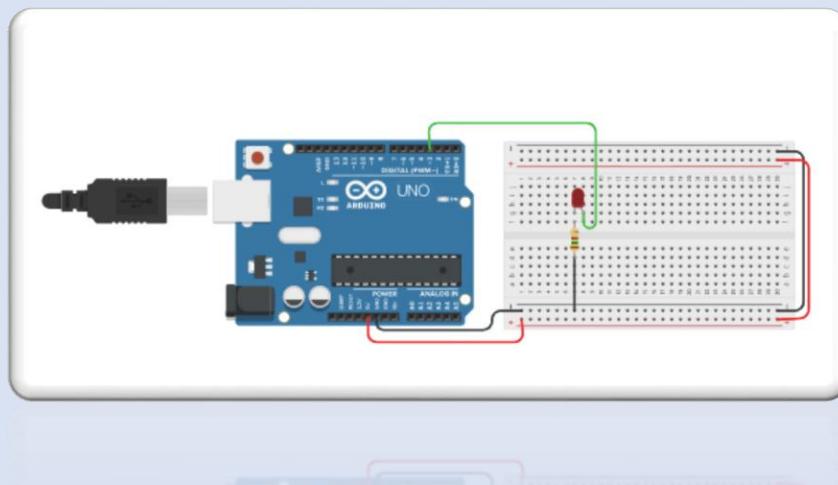
- **DIAGRAMA**



Ejercicio # 10

UN ARDUINO UNO CONECTADO EN UN PROTOBOARD Y UN LED CON SU RESISTENCIA.

- DIAGRAMA



- CÓDIGO

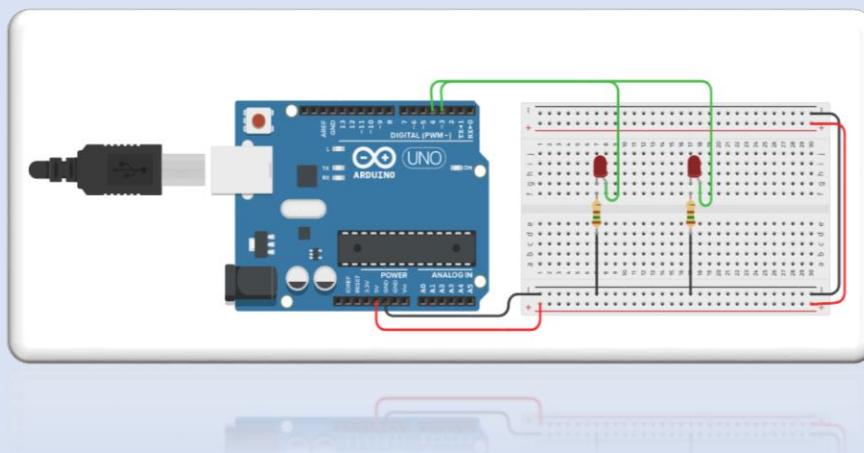
```
// C++ code
//
void setup()
{
    pinMode(3, OUTPUT);
}

void loop()
{
    digitalWrite(3, HIGH);
    delay(1000);
    digitalWrite(3, LOW);
    delay(1000);
}
```

Ejercicio # 11

UN ARDUINO UNO CONECTADO EN UN PROTOBOARD Y 2 LEDS CON SUS RESISTENCIA.

- DIAGRAMA



- CÓDIGO

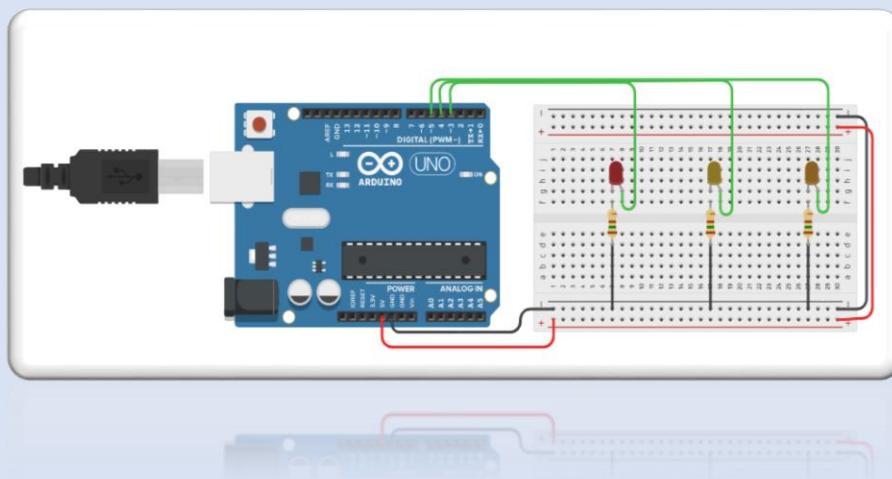
```
// C++ code
//
void setup()
{
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
}

void loop()
{
    digitalWrite(3, HIGH);
    digitalWrite(4, LOW);
    delay(1000);
    digitalWrite(3, LOW);
    digitalWrite(4, HIGH);
    delay(1000);
}
```

Ejercicio # 12

UN ARDUINO UNO CONECTADO EN UN PROTOBOARD Y 3 LEDS DE DIFERENTE COLOR CON SUS RESISTENCIA.

- **DIAGRAMA**



- **CÓDIGO**

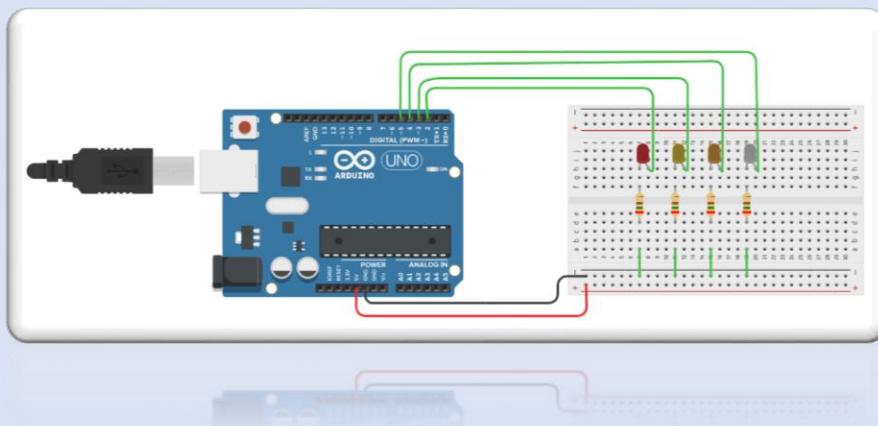
```
// C++ code
//
void setup()
{
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
}

void loop()
{
    digitalWrite(3, HIGH);
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
    delay(1000);
    digitalWrite(3, LOW);
    digitalWrite(4, HIGH);
    digitalWrite(5, HIGH);
    delay(1000);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
    digitalWrite(5, HIGH);
    delay(1000);
}
```

Ejercicio # 13

UN ARDUINO UNO CONECTADO EN UN PROTOBOARD Y 4 LEDS DE DIFERENTE COLOR CON SUS RESISTENCIA.

- **DIAGRAMA**



- **CÓDIGO**

```
// C++ code
//

int rojo=2;
int amarrillo=3;
int naranja=4;
int blanco=5;

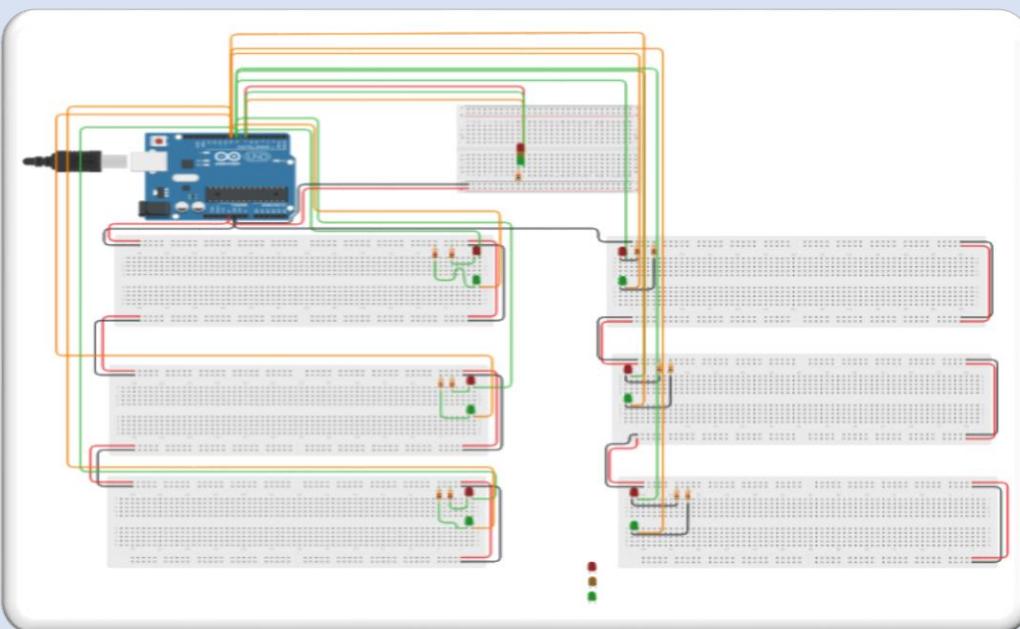
void setup()
{
    pinMode(rojo, OUTPUT);
    pinMode(amarrillo, OUTPUT);
    pinMode(naranja, OUTPUT);
    pinMode(blanco, OUTPUT);
}

void loop()
{
    digitalWrite(rojo, HIGH);
    digitalWrite(amarrillo, HIGH);
    digitalWrite(naranja, LOW);
    digitalWrite(blanco, LOW);
    delay(5000); // Wait for 1000 millisecond(s)
    digitalWrite(naranja, HIGH);
    digitalWrite(blanco, HIGH);
    digitalWrite(rojo, LOW);
    digitalWrite(amarrillo, LOW);
    delay(5000); // Wait for 1000 millisecond(s)
}
```

Ejercicio # 14

UN ARDUINO UNO CONECTADO CON 7 PROTOBOARD Y LEDS DE DIFERENTE COLOR CON SUS RESISTENCIA.

- **DIAGRAMA**



- **CÓDIGO**

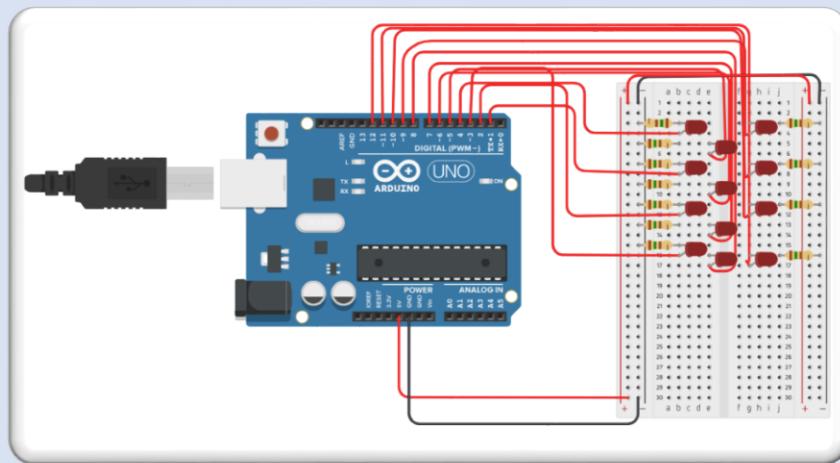
```
// C++ code
//
void setup()
{
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);
}

void loop()
{
    /////
    digitalWrite(2, HIGH);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
    digitalWrite(7, HIGH);
    delay(8000);
    digitalWrite(2, LOW);
    digitalWrite(3, HIGH);
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
    digitalWrite(7, HIGH);
    delay(4000);
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(4, HIGH);
    digitalWrite(5, HIGH);
    digitalWrite(7, LOW);
    delay(4000);
}
```

Ejercicio # 15

UN ARDUINO UNO CONECTADO EN UN PROTOBOARD Y LEDS DE DIFERENTE COLOR CON SUS RESISTENCIA INDIVIDUALES.

- DIAGRAMA



- CÓDIGO

```
// C++ code
//
void setup()
{
    pinMode(4, OUTPUT);
    pinMode(1, OUTPUT);
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);
    pinMode(9, OUTPUT);
    pinMode(10, OUTPUT);
    pinMode(11, OUTPUT);
    pinMode(12, OUTPUT);
}

void loop()
{
    digitalWrite(4, HIGH);
    digitalWrite(1, LOW);
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    digitalWrite(11, LOW);
    digitalWrite(12, LOW);
    delay(1000);

    digitalWrite(4, LOW);
    digitalWrite(1, HIGH);
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    digitalWrite(11, LOW);
    digitalWrite(12, LOW);
    delay(1000);
```

```
digitalWrite(4, LOW);
digitalWrite(1, LOW);
digitalWrite(2, LOW);
digitalWrite(3, HIGH);
digitalWrite(5, LOW);
digitalWrite(6, LOW);
digitalWrite(7, LOW);
digitalWrite(8, LOW);
digitalWrite(9, LOW);
digitalWrite(10, LOW);
digitalWrite(11, LOW);
digitalWrite(12, LOW);
delay(1000);

digitalWrite(4, LOW);
digitalWrite(1, LOW);
digitalWrite(2, LOW);
digitalWrite(3, LOW);
digitalWrite(5, HIGH);
digitalWrite(6, LOW);
digitalWrite(7, LOW);
digitalWrite(8, LOW);
digitalWrite(9, LOW);
digitalWrite(10, LOW);
digitalWrite(11, LOW);
digitalWrite(12, LOW);
delay(1000);

    digitalWrite(4, LOW);
    digitalWrite(1, LOW);
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
    digitalWrite(7, HIGH);
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    digitalWrite(11, LOW);
    digitalWrite(12, LOW);
    delay(1000);

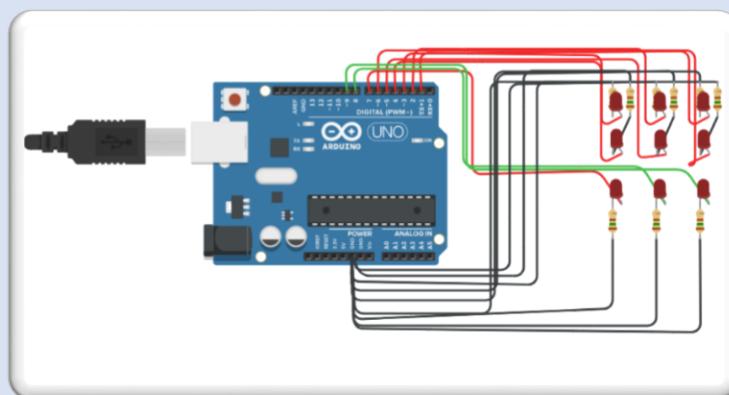
    digitalWrite(4, LOW);
    digitalWrite(1, LOW);
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    digitalWrite(10, HIGH);
    digitalWrite(11, LOW);
    digitalWrite(12, LOW);
    delay(1000);

    digitalWrite(4, LOW);
    digitalWrite(1, LOW);
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    digitalWrite(11, HIGH);
    digitalWrite(12, LOW);
    delay(1000);
```

EJERCICIO # 16

EJERCICIO CON 9 LEDS EN ARDUINO MOSTRANDO LAS LETRAS X, Y, Z.

- DIAGRAMA



- CÓDIGO

```
Texto
1 // C++ code
2 //
3 void setup()
4 {
5     pinMode(1, OUTPUT);
6     pinMode(2, OUTPUT);
7     pinMode(3, OUTPUT);
8     pinMode(4, OUTPUT);
9     pinMode(5, OUTPUT);
10    pinMode(6, OUTPUT);
11    pinMode(7, OUTPUT);
12    pinMode(8, OUTPUT);
13    pinMode(9, OUTPUT);
14 }
15
16 void loop()
17 {
18
19     digitalWrite(1, HIGH);
20     digitalWrite(2, LOW);
21     digitalWrite(3, HIGH);
22     digitalWrite(4, LOW);
23     digitalWrite(5, HIGH);
24     digitalWrite(6, LOW);
25     digitalWrite(7, HIGH);
26     digitalWrite(8, LOW);
27     digitalWrite(9, HIGH);
28     delay(3000);
29
30 }
```

Monitor en serie

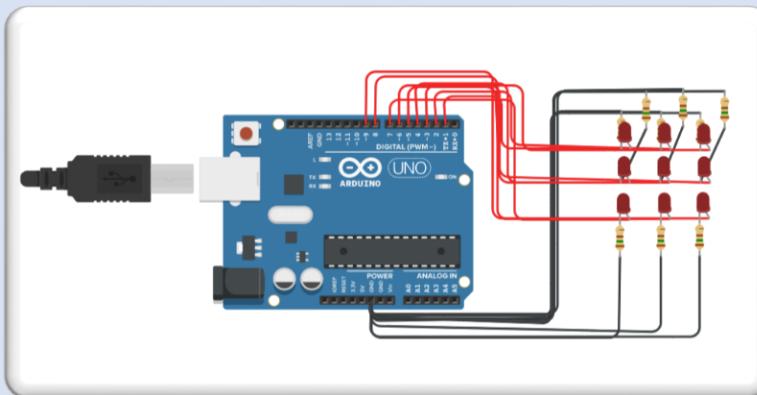
```
1 (Arduino Uno R3)
42     digitalWrite(1, HIGH);
43     digitalWrite(2, HIGH);
44     digitalWrite(3, HIGH);
45     digitalWrite(4, LOW);
46     digitalWrite(5, HIGH);
47     digitalWrite(6, LOW);
48     digitalWrite(7, HIGH);
49     digitalWrite(8, HIGH);
50     digitalWrite(9, HIGH);
51     delay(3000);
52
53
54
55
56
57 }
```

Monitor en serie

EJERCICIO #17

EJERCICIO CON 9 LEDS EN ARDUINO FORMANDO LAS LETRAS X,
Y, Z Y LA INICIAL DE MI APELLIDO (Z).

- DIAGRAMA



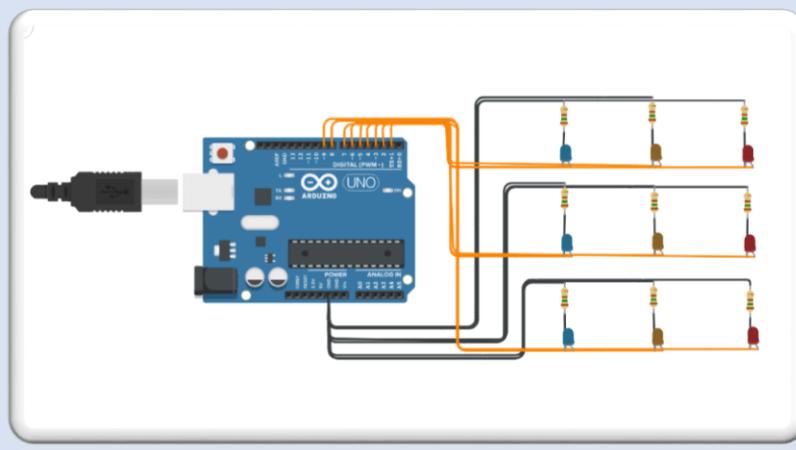
- CÓDIGO

```
Texto
1 // C++ code
2 //
3 void setup()
4 {
5     pinMode(1, OUTPUT);
6     pinMode(2, OUTPUT);
7     pinMode(3, OUTPUT);
8     pinMode(4, OUTPUT);
9     pinMode(5, OUTPUT);
10    pinMode(6, OUTPUT);
11    pinMode(7, OUTPUT);
12    pinMode(8, OUTPUT);
13    pinMode(9, OUTPUT);
14 }
15 }
16
17 void loop()
18 {
19     digitalWrite(1, HIGH);
20     digitalWrite(2, LOW);
21     digitalWrite(3, HIGH);
22     digitalWrite(4, LOW);
23     digitalWrite(5, HIGH);
24     digitalWrite(6, LOW);
25     digitalWrite(7, HIGH);
26     digitalWrite(8, LOW);
27     digitalWrite(9, HIGH);
28     delay(5000);
29
30     digitalWrite(1, HIGH);
31
32     digitalWrite(5, HIGH);
33     digitalWrite(6, LOW);
34     digitalWrite(7, LOW);
35     digitalWrite(8, HIGH);
36     digitalWrite(9, LOW);
37
38     delay(5000);
39
40     digitalWrite(1, HIGH);
41     digitalWrite(2, HIGH);
42     digitalWrite(3, HIGH);
43     digitalWrite(4, LOW);
44     digitalWrite(5, HIGH);
45     digitalWrite(6, LOW);
46     digitalWrite(7, HIGH);
47     digitalWrite(8, HIGH);
48     digitalWrite(9, HIGH);
49
50     delay(5000);
51
52     digitalWrite(1, HIGH);
53     digitalWrite(2, HIGH);
54     digitalWrite(3, HIGH);
55     digitalWrite(4, LOW);
56     digitalWrite(5, HIGH);
57     digitalWrite(6, LOW);
58     digitalWrite(7, HIGH);
59     digitalWrite(8, HIGH);
60     digitalWrite(9, LOW);
61
62
63 }
```

EJERCICIO #18

EJERCICIO CON 9 LEDS EN ARDUINO FORMANDO NUMEROS 1, 2, 3, 4, 5.

- **DIAGRAMA**



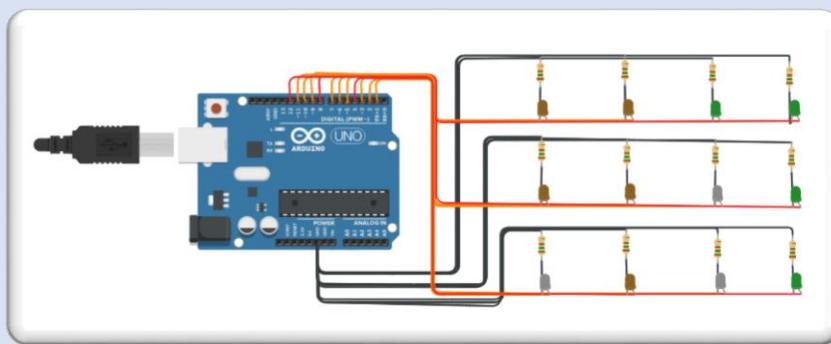
- **CÓDIGO**

```
Texto
1 // C++ code
2 //
3 void setup()
4 {
5     pinMode(1, OUTPUT);
6     pinMode(2, OUTPUT);
7     pinMode(3, OUTPUT);
8     pinMode(4, OUTPUT);
9     pinMode(5, OUTPUT);
10    pinMode(6, OUTPUT);
11    pinMode(7, OUTPUT);
12    pinMode(8, OUTPUT);
13    pinMode(9, OUTPUT);
14 }
15
16 void loop()
17 {
18     digitalWrite(1, HIGH);
19     digitalWrite(2, HIGH);
20     digitalWrite(3, LOW);
21     digitalWrite(4, HIGH);
22     digitalWrite(5, HIGH);
23     digitalWrite(6, LOW);
24     digitalWrite(7, HIGH);
25     digitalWrite(8, HIGH);
26     digitalWrite(9, LOW);
27     delay(2000);
28
29
30     digitalWrite(1, LOW);
}
Monitor en serie
1 (Arduino Uno R3)
41     digitalWrite(1, HIGH);
42     digitalWrite(2, HIGH);
43     digitalWrite(3, LOW);
44     digitalWrite(4, HIGH);
45     digitalWrite(5, HIGH);
46     digitalWrite(6, LOW);
47     digitalWrite(7, HIGH);
48     digitalWrite(8, HIGH);
49     digitalWrite(9, LOW);
50     delay(2000);
51
52     digitalWrite(1, HIGH);
53     digitalWrite(2, HIGH);
54     digitalWrite(3, LOW);
55     digitalWrite(4, HIGH);
56     digitalWrite(5, HIGH);
57     digitalWrite(6, LOW);
58     digitalWrite(7, HIGH);
59     digitalWrite(8, HIGH);
60     digitalWrite(9, LOW);
61     delay(2000);
62
63     digitalWrite(1, LOW);
64     digitalWrite(2, HIGH);
65     digitalWrite(3, LOW);
66     digitalWrite(4, HIGH);
67     digitalWrite(5, HIGH);
68     digitalWrite(6, LOW);
69     digitalWrite(7, LOW);
70     digitalWrite(8, HIGH);
}
Monitor en serie
```

EJERCICIO #19

EJERCICIO CON 12 LEDS EN ARDUINO FORMANDO NUMEROS DE DOS DIGITOS.

- **DIAGRAMA**



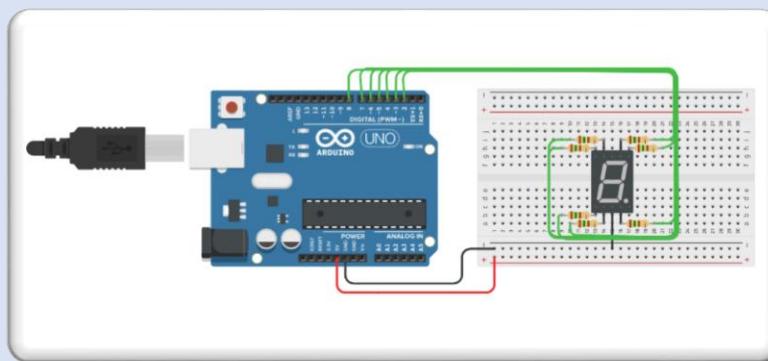
- **CÓDIGO**

```
Texto
3 void setup()
4 {
5     pinMode(1, OUTPUT);
6     pinMode(2, OUTPUT);
7     pinMode(3, OUTPUT);
8     pinMode(4, OUTPUT);
9     pinMode(5, OUTPUT);
10    pinMode(6, OUTPUT);
11    pinMode(7, OUTPUT);
12    pinMode(8, OUTPUT);
13    pinMode(9, OUTPUT);
14    pinMode(10, OUTPUT);
15    pinMode(11, OUTPUT);
16    pinMode(12, OUTPUT);
17 }
18
19 void loop()
20 {
21     digitalWrite(1, HIGH);
22     digitalWrite(2, HIGH);
23     digitalWrite(3, HIGH);
24     digitalWrite(4, HIGH);
25     digitalWrite(5, HIGH);
26     digitalWrite(6, HIGH);
27     digitalWrite(7, LOW);
28     digitalWrite(8, HIGH);
29     digitalWrite(9, LOW);
30     digitalWrite(10, HIGH);
31     digitalWrite(11, LOW);
32     digitalWrite(12, HIGH);
}
Monitor en serie
```

EJERCICIO #20

EJERCICIO CON VISUALIZADOR DE 7 SEGMENTOS QUE MUESTRE DEL 0 AL 5.

- **DIAGRAMA**



- **CÓDIGO**

```
Texto 1 (Arduino Uno R3)

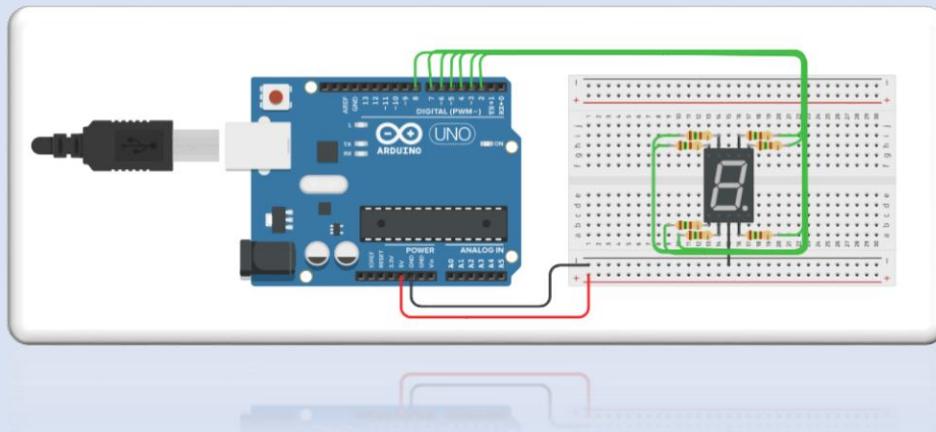
3 int a=2;
4 int b=3;
5 int c=4;
6 int d=5;
7 int e=6;
8 int f=7;
9 int g=8;
10
11 void setup()
12 {
13     pinMode(a, OUTPUT);
14     pinMode(b, OUTPUT);
15     pinMode(c, OUTPUT);
16     pinMode(d, OUTPUT);
17     pinMode(e, OUTPUT);
18     pinMode(f, OUTPUT);
19     pinMode(g, OUTPUT);
20 }
21
22 void loop()
23 {
24     digitalWrite(a, HIGH);
25     digitalWrite(b, HIGH);
26     digitalWrite(c, HIGH);
27     digitalWrite(d, HIGH);
28     digitalWrite(e, HIGH);
29     digitalWrite(f, HIGH);
30     digitalWrite(g, LOW);
31     delay(1000);
32 }

43     digitalWrite(a, HIGH);
44     digitalWrite(b, LOW);
45     digitalWrite(d, HIGH);
46     digitalWrite(e, HIGH);
47     digitalWrite(f, LOW);
48     digitalWrite(g, HIGH);
49     delay(1000);
50
51     digitalWrite(a, HIGH);
52     digitalWrite(b, HIGH);
53     digitalWrite(c, HIGH);
54     digitalWrite(d, HIGH);
55     digitalWrite(e, LOW);
56     digitalWrite(f, LOW);
57     digitalWrite(g, HIGH);
58     delay(1000);
59
60     digitalWrite(a, LOW);
61     digitalWrite(b, HIGH);
62     digitalWrite(c, HIGH);
63     digitalWrite(d, LOW);
64     digitalWrite(e, LOW);
65     digitalWrite(f, HIGH);
66     digitalWrite(g, HIGH);
67     delay(1000);
68
69     digitalWrite(a, HIGH);
70     digitalWrite(b, LOW);
71     digitalWrite(c, HIGH);
```

EJERCICIO #21

EJERCICIO CON VISUALIZADOR DE 7 SEGMENTOS QUE MUESTRE LA LETRA A AL D MAYUSCULAS Y MINUSCULAS.

- **DIAGRAMA**



- **CÓDIGO**

```
Texto

4 int b=3;
5 int c=4;
6 int d=5;
7 int e=6;
8 int f=7;
9 int g=8;
10
11 void setup()
12 {
13   pinMode(a, OUTPUT);
14   pinMode(b, OUTPUT);
15   pinMode(c, OUTPUT);
16   pinMode(d, OUTPUT);
17   pinMode(e, OUTPUT);
18   pinMode(f, OUTPUT);
19   pinMode(g, OUTPUT);
20 }
21
22 void loop()
23 {
24   digitalWrite(a, HIGH);
25   digitalWrite(b, HIGH);
26   digitalWrite(c, HIGH);
27   digitalWrite(d, HIGH);
28   digitalWrite(e, HIGH);
29   digitalWrite(f, LOW);
30   digitalWrite(g, HIGH);
31   delay(1000);
32
33   digitalWrite(a, LOW);
34
35   Monitor en serie
```

```
24   digitalWrite(a, HIGH);
25   digitalWrite(b, HIGH);
26   digitalWrite(c, HIGH);
27   digitalWrite(d, HIGH);
28   digitalWrite(e, HIGH);
29   digitalWrite(f, LOW);
30   digitalWrite(g, HIGH);
31   delay(1000);
32
33   digitalWrite(a, LOW);
34
35   digitalWrite(a, LOW);
36   digitalWrite(b, LOW);
37   digitalWrite(c, HIGH);
38   digitalWrite(d, HIGH);
39   digitalWrite(e, HIGH);
40   digitalWrite(f, HIGH);
41   digitalWrite(g, HIGH);
42   delay(1000);
43
44   digitalWrite(a, HIGH);
45   digitalWrite(b, HIGH);
46   digitalWrite(c, HIGH);
47   digitalWrite(d, HIGH);
48   digitalWrite(e, HIGH);
49   digitalWrite(f, LOW);
50   digitalWrite(g, HIGH);
51   delay(1000);
52
53   digitalWrite(a, HIGH);
54   digitalWrite(b, HIGH);
55   digitalWrite(c, HIGH);
56   digitalWrite(d, HIGH);
57   digitalWrite(e, HIGH);
58   digitalWrite(f, HIGH);
59   digitalWrite(g, HIGH);
60   delay(1000);
61
62   Monitor en serie
```

```
digitalWrite(a, HIGH);
digitalWrite(b, HIGH);
digitalWrite(c, LOW);
digitalWrite(d, HIGH);
digitalWrite(e, HIGH);
digitalWrite(f, HIGH);
digitalWrite(g, HIGH);
delay(1000);

//mayusculas

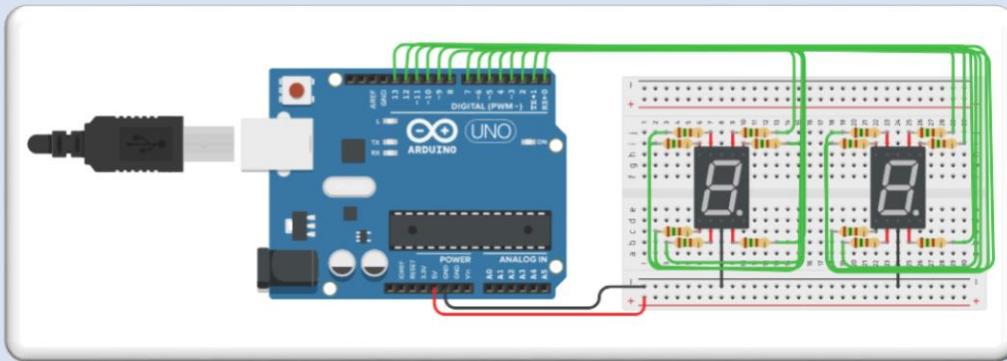
digitalWrite(a, HIGH);
digitalWrite(b, HIGH);
digitalWrite(c, HIGH);
digitalWrite(d, LOW);
digitalWrite(e, HIGH);
digitalWrite(f, HIGH);
digitalWrite(g, HIGH);
delay(1000);

digitalWrite(a, HIGH);
digitalWrite(b, HIGH);
digitalWrite(c, HIGH);
digitalWrite(d, HIGH);
digitalWrite(e, HIGH);
digitalWrite(f, HIGH);
digitalWrite(g, HIGH);
delay(1000);
```

EJERCICIO #22

EJERCICIO CON 2 VISUALIZADORES DE 7 SEGMENTOS QUE MUESTRE 1 EN LETRAS Y 1 EN NUMEROS.

- DIAGRAMA



- CÓDIGO

```
Texto
3 int a=2;
4 int b=3;
5 int c=4;
6 int d=5;
7 int e=6;
8 int f=7;
9 int g=8;
10
11 int a1=9;
12 int b1=10;
13 int c1=11;
14 int d1=12;
15 int e1=13;
16 int f1=0;
17 int g1=1;
18
19
20
21 void setup()
22 {
23   pinMode(a, OUTPUT);
24   pinMode(b, OUTPUT);
25   pinMode(c, OUTPUT);
26   pinMode(d, OUTPUT);
27   pinMode(e, OUTPUT);
28   pinMode(f, OUTPUT);
29   pinMode(g, OUTPUT);
30
31   pinMode(a1, OUTPUT);
32   pinMode(b1, OUTPUT);
33 }

void loop()
{
  digitalWrite(a, HIGH);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, LOW);
  digitalWrite(e, HIGH);
  digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);

  digitalWrite(a1, HIGH);
  digitalWrite(b1, LOW);
  digitalWrite(c1, HIGH);
  digitalWrite(d1, HIGH);
  digitalWrite(e1, LOW);
  digitalWrite(f1, HIGH);
  digitalWrite(g1, HIGH);
  delay(1000);

  digitalWrite(a, HIGH);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, HIGH);
  digitalWrite(e, HIGH);
  digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);

  digitalWrite(a1, LOW);
  digitalWrite(b1, HIGH);
  digitalWrite(c1, HIGH);
  digitalWrite(d1, HIGH);
  digitalWrite(e1, LOW);
  digitalWrite(f1, HIGH);
  digitalWrite(g1, HIGH);
  delay(1000);

  digitalWrite(a, HIGH);
  digitalWrite(b, LOW);
  digitalWrite(c, LOW);
  digitalWrite(d, HIGH);
  digitalWrite(e, HIGH);
  digitalWrite(f, HIGH);
  digitalWrite(g, LOW);

  digitalWrite(a1, HIGH);
  digitalWrite(b1, HIGH);
  digitalWrite(c1, HIGH);
  digitalWrite(d1, HIGH);
  digitalWrite(e1, LOW);

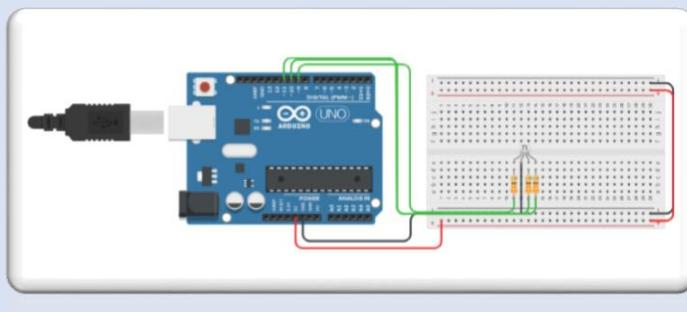
}

Monitor en serie
Monitor en serie
```

EJERCICIO #23

EJERCICIO CON LED RGB EN ARDUINO CON SU PROTOBOARD.

- **DIAGRAMA**



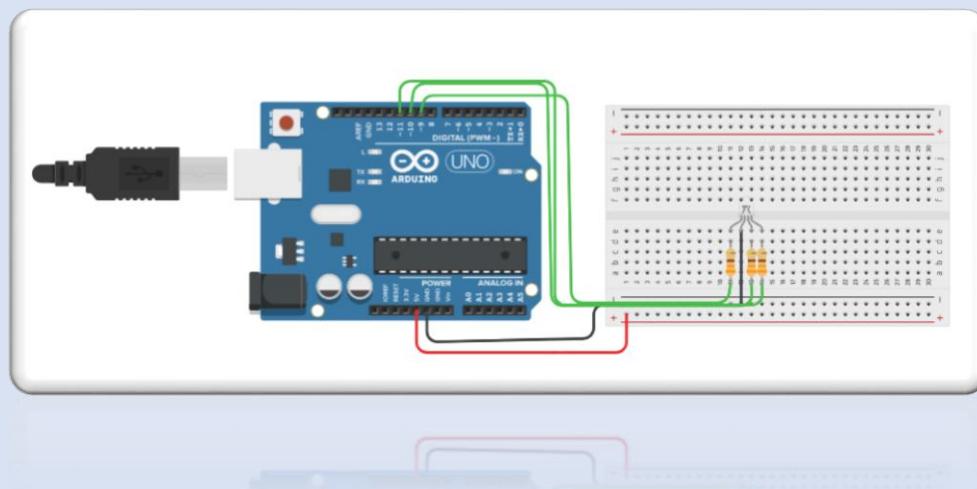
- **CÓDIGO**

```
Texto          ↓  ↳  A  1 (Arduino Uno R3) ▾
1 // C++ code
2 //
3 int rojo=9;
4 int azul=10;
5 int verde=11;
6 void setup()
7 {
8     pinMode(rojo, OUTPUT);
9     pinMode(azul, OUTPUT);
10    pinMode(verde, OUTPUT);
11 }
12
13 void loop()
14 {
15     analogWrite(rojo, 224);
16     analogWrite(verde, 45);
17     analogWrite(azul, 25);
18     delay(1000);
19
20
21
22 }
```

EJERCICIO #24

EJERCICIO CON LED RGB EN ARDUINO CON SUS RESPECTIVOS COLORES.

- **DIAGRAMA**



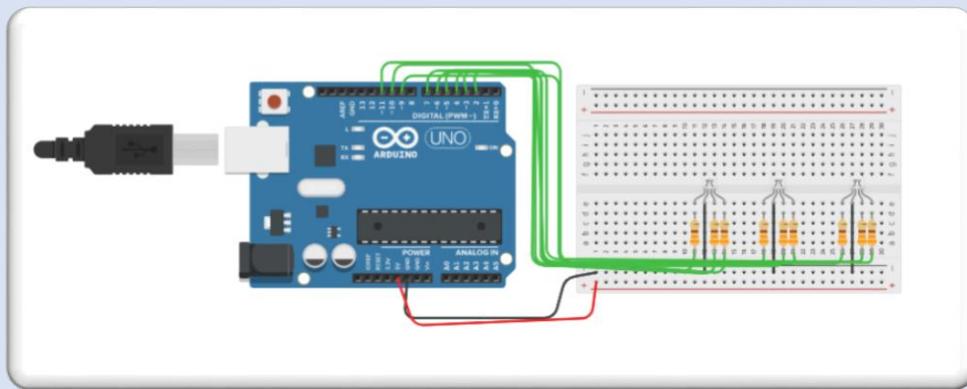
- **CÓDIGO**

```
Texto 1 (Arduino Uno R3)
1 // C++ code
2 //
3 int rojo=9;
4 int azul=10;
5 int verde=11;
6 void setup()
7 {
8     pinMode(rojo, OUTPUT);
9     pinMode(azul, OUTPUT);
10    pinMode(verde, OUTPUT);
11 }
12
13 void loop()
14 {
15     analogWrite(rojo, 224);
16     analogWrite(verde, 45);
17     analogWrite(azul, 25);
18     delay(1000);
19
20     analogWrite(rojo, 255);
21     analogWrite(verde, 211);
22     analogWrite(azul, 170);
23     delay(1000);
24
25     analogWrite(rojo, 170);
26     analogWrite(verde, 240);
27     analogWrite(azul, 255);
28     delay(1000);
29
30 }
```

EJERCICIO #25

EJERCICIO CON 3 LEDS RGB EN ARDUINO.

- **DIAGRAMA**



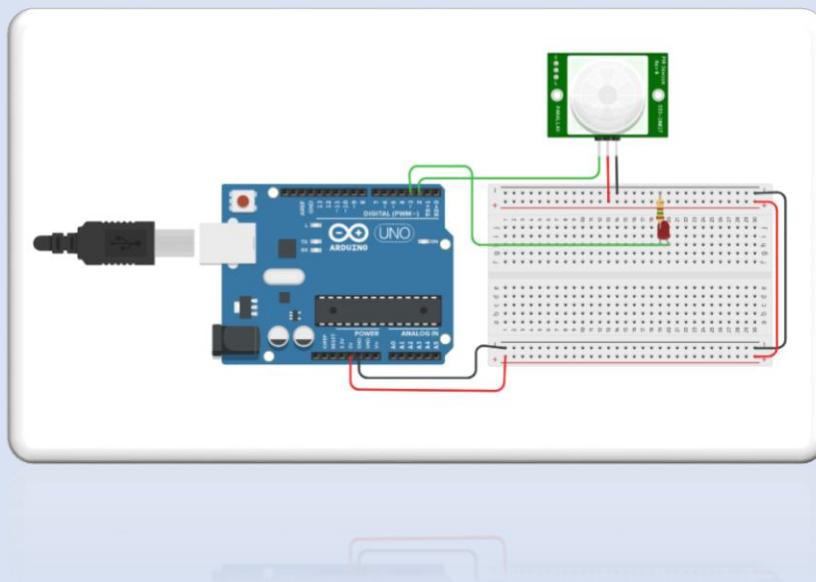
- **CÓDIGO**

```
Texto
1 int rojo1=9;
2 int azul1=10;
3 int verde1=11;
4 int rojo2=2;
5 int azul2=3;
6 int verde2=4;
7 int rojo3=5;
8 int azul3=6;
9 int verde3=7;
10 void setup()
11 {
12   pinMode(rojo1, OUTPUT);
13   pinMode(azul1, OUTPUT);
14   pinMode(verde1, OUTPUT);
15 }
16
17 void loop()
18 {
19   analogWrite(rojo1, 224);
20   analogWrite(verde1, 45);
21   analogWrite(azul1, 25);
22
23   analogWrite(rojo2, 255);
24   analogWrite(verde2, 211);
25   analogWrite(azul2, 170);
26
27   analogWrite(rojo3, 170);
28   analogWrite(verde3, 240);
29   analogWrite(azul3, 255);
30 }
```

EJERCICIO #26

EJERCICIO CON SENSOR PIR EN ARDUINO.

- **DIAGRAMA**



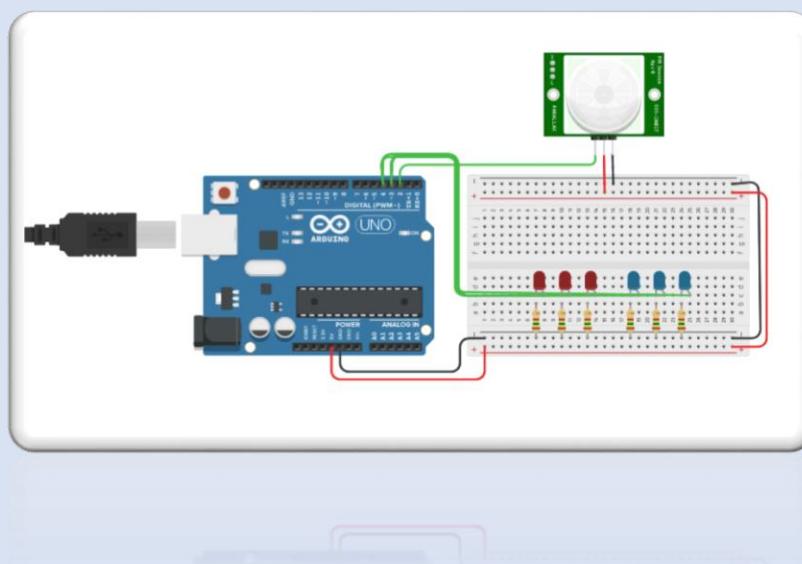
- **CÓDIGO**

```
Texto          ▾  ↴  ↵  AA  ▾  1 (Arduino Uno R3) ▾  
1 int pir=2;  
2 int led1=3;  
3 int dt;  
4 void setup()  
5 {  
6   pinMode(pir, INPUT);  
7   pinMode(led1, OUTPUT);  
8  
9 }  
10  
11 void loop()  
12 {  
13   dt=digitalRead(pir);  
14  
15   if(dt==HIGH)  
16   {  
17     digitalWrite(led1, HIGH);  
18     delay(1000); // Wait for 1000 millisecond(s)  
19   }  
20   else  
21   {  
22     digitalWrite(led1, LOW);  
23   }  
24  
25 }
```

EJERCICIO #27

EJERCICIO CON SENSOR PIR EN ARDUINO COLOCANDO LEDS.

- **DIAGRAMA**



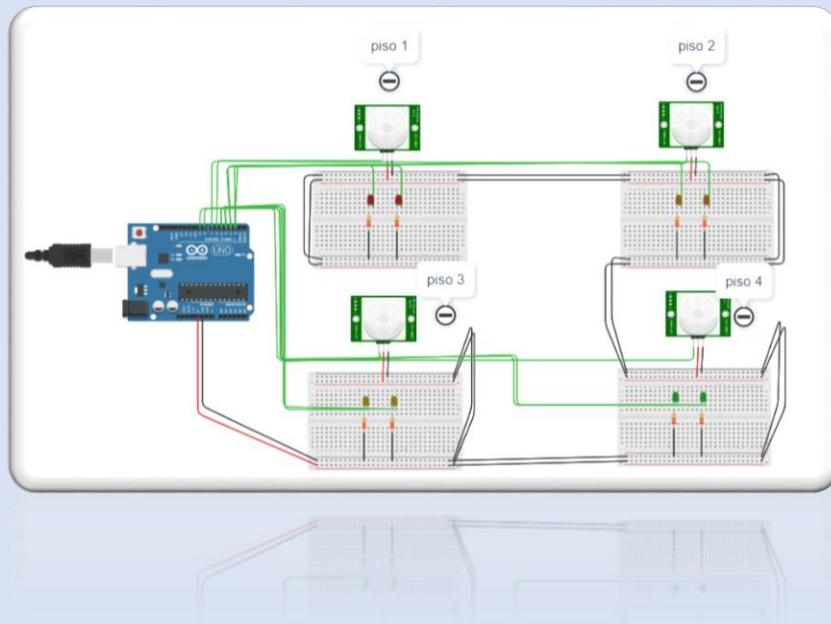
- **CÓDIGO**

```
Texto          ↓  ↴  AA  1 (Arduino Uno R3) ▾  
1 int pir=2;  
2 int led1=3;  
3 int led2=4;  
4  
5 int dt;  
6 void setup()  
{  
7   pinMode(pir, INPUT);  
8   pinMode(led1, OUTPUT);  
9   pinMode(led2, OUTPUT);  
10  
11 }  
12  
13 void loop()  
14 {  
15   dt=digitalRead(pir);  
16  
17   if(dt==HIGH)  
18   {  
19     digitalWrite(led1, HIGH);  
20     digitalWrite(led2, LOW);  
21     delay(1000);  
22   }  
23   else  
24   {  
25     digitalWrite(led1, LOW);  
26     digitalWrite(led2, HIGH);  
27     delay(1000);  
28   }  
29  
30 }  
Monitor en serie
```

EJERCICIO #28

EJERCICIO CON 4 SENSORES PIR CON SUS RESPECTIVOS LEDS EN ARDUINO.

- **DIAGRAMA**



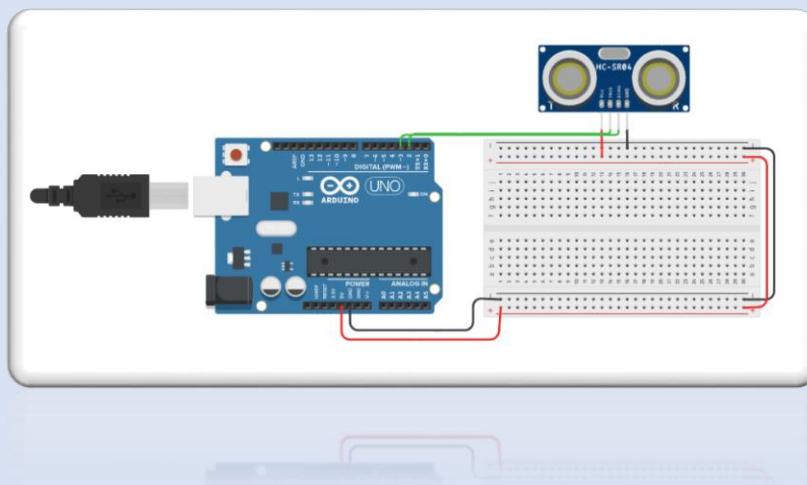
- **CÓDIGO**

```
Texto          1 (Arduino Uno R3)
1 int r= 2;
2 int n= 3;
3 int a= 4;
4 int v= 5;
5 int s1= 6;
6 int s2= 7;
7 int s3= 8;
8 int s4= 9;
9 int d1,d2,d3,d4;
10
11 int t=200;
12
13 void setup()
14 {
15   pinMode(r, OUTPUT);
16   pinMode(n, OUTPUT);
17   pinMode(a, OUTPUT);
18   pinMode(v, OUTPUT);
19   pinMode(s1, INPUT);
20   pinMode(s2, INPUT);
21   pinMode(s3, INPUT);
22   pinMode(s4, INPUT);
23 }
24
25 }
26
27 void loop()
28 {
29   //s1
30   d1= digitalRead(s1);
31   if(d1==HIGH)
32   {
33     digitalWrite(r, LOW);
34     delay(1);
35   }
36   else
37   {
38     digitalWrite(r, HIGH);
39     delay(t);
40   }
41   //s2
42   d2= digitalRead(s2);
43   if(d2==HIGH)
44   {
45     digitalWrite(n, LOW);
46     delay(1);
47   }
48   else
49   {
50     digitalWrite(n, HIGH);
51     delay(t);
52   }
53 }
54
55 //s3
56 d3= digitalRead(s3);
57 if(d3==HIGH)
58 {
59   digitalWrite(a, HIGH);
60   delay(1);
61 }
62 else
63 {
64   digitalWrite(a, LOW);
65   delay(t);
66 }
67
68 //s4
69 d4= digitalRead(s4);
70 if(d4==HIGH)
71 {
72   digitalWrite(v, HIGH);
73   delay(1);
74 }
75 else
76 {
77   digitalWrite(v, LOW);
78   delay(t);
79 }
80
81 }
```

EJERCICIO #29

EJERCICIO CON SENSOR DE DISTANCIA ULTRASONICO EN ARDUINO.

- DIAGRAMA



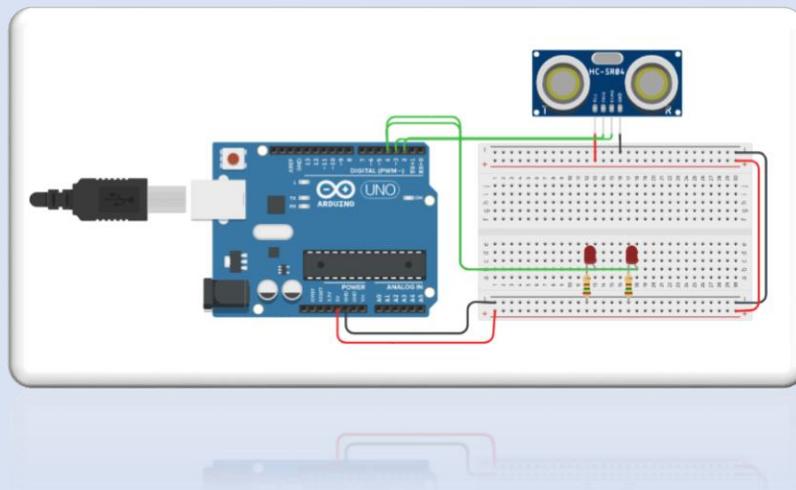
- CÓDIGO

```
Texto 1 (Arduino Uno R3)
1 int trig=2;
2 int eco=3;
3 int led1=4;
4 int t;
5 int d;
6
7 void setup()
8 {
9   pinMode(led1, OUTPUT);
10  pinMode(trig, OUTPUT);
11  pinMode(eco, INPUT);
12  Serial.begin(9600);
13
14
15 }
16
17 void loop()
18 {
19   digitalWrite(trig, HIGH);
20   delay(1000);
21   digitalWrite(trig, LOW);
22
23   t= pulseIn(eco,HIGH);
24   d= t/58.2;
25
26   Serial.println(d);
27   Serial.println("cm");
28   delay(500);
29 }
30 }
```

EJERCICIO #30

EJERCICIO CON SENSOR DE DISTANCIA ULTRASONICO CON LEDS EN ARDUINO.

- **DIAGRAMA**



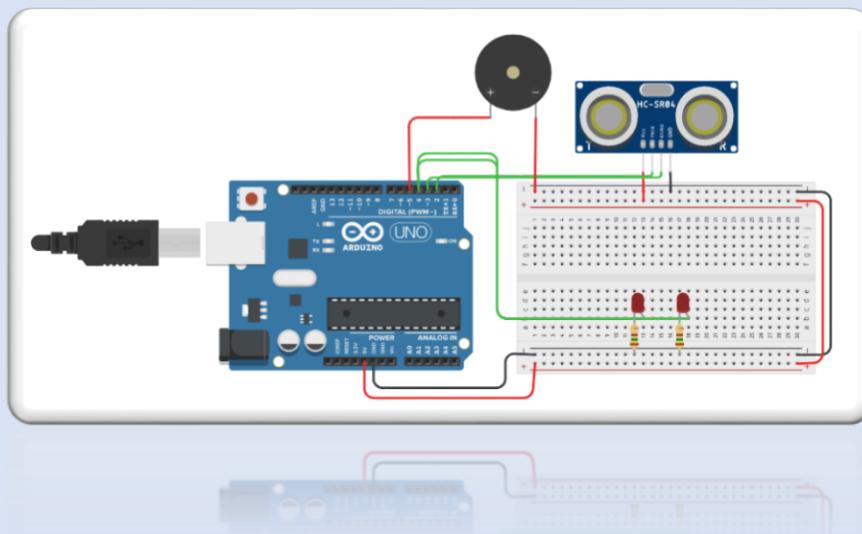
- **CÓDIGO**

```
1 int trig=2;
2 int eco=3;
3 int led1=4;
4 int t;
5 int d;
6
7 void setup()
8 {
9   pinMode(led1, OUTPUT);
10  pinMode(trig, OUTPUT);
11  pinMode(eco, INPUT);
12  Serial.begin(9600);
13
14
15
16 }
17
18 void loop()
19 {
20   digitalWrite(trig, HIGH);
21   delay(1);
22   digitalWrite(trig, LOW);
23
24   t= pulseIn(eco,HIGH);
25   d= t/58.2;
26
27   Serial.println(d);
28   Serial.println("cm");
29   delay(50);
30
31   if(d<100)
32   {
33     digitalWrite(led1, HIGH);
34     delay(1);
35   }
36   else
37   {
38     digitalWrite(led1, LOW);
39   }
40
41
42 }
```

EJERCICIO #31

EJERCICIO CON SENSOR DE DISTANCIA ULTRASONICO CON LEDS Y BUZZER EN ARDUINO.

- **DIAGRAMA**



- **CÓDIGO**

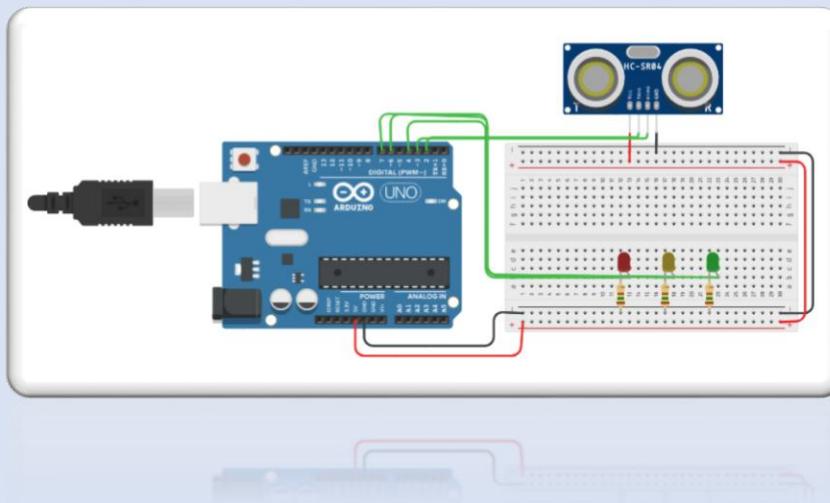
```
Texto          ↓  ↴  ↵  ↵  ↵  1 (Arduino Uno R3)

1 int trig=2;
2 int eco=3;
3 int led1=4;
4 int t;
5 int d;
6 int buz=5;
7
8 void setup()
9 {
10   pinMode(led1, OUTPUT);
11   pinMode(trig, OUTPUT);
12   pinMode(buz, OUTPUT);
13
14   pinMode(eco, INPUT);
15   Serial.begin(9600);
16
17
18
19 }
20
21 void loop()
22 {
23   digitalWrite(trig, HIGH);
24   delay(1);
25   digitalWrite(trig, LOW);
26
27   t= pulseIn(eco,HIGH);
28   d= t/58.2;
29
30   Serial.println(d);
31
32
33
34   if(d<100)
35   {
36     digitalWrite(led1, HIGH);
37     tone(buz,1000,10);
38     delay(1);
39   }
40   else
41   {
42     digitalWrite(led1, LOW);
43     noTone(buz);
44   }
45
46
47 }
```

EJERCICIO #32

EJERCICIO CON SENSOR DE DISTANCIA ULTRASONICO CON LEDS Y SUS RESPECTIVAS DISTANCIAS EN ARDUINO.

- **DIAGRAMA**



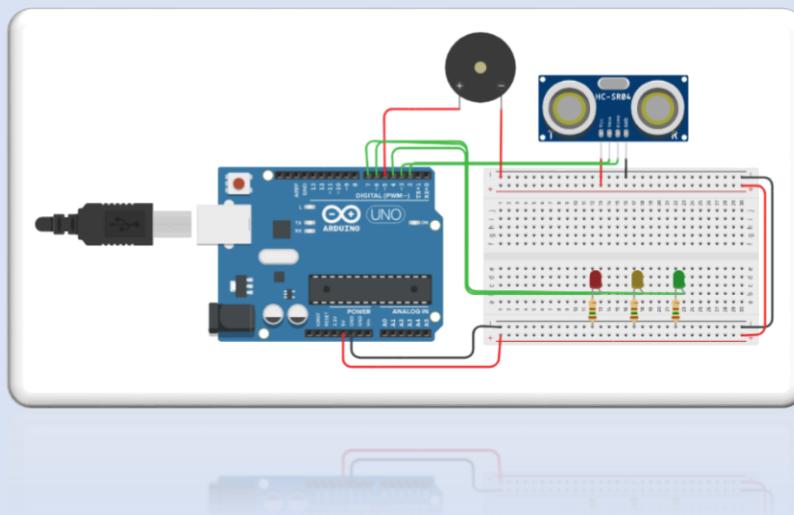
- **CÓDIGO**

```
Texto          ↓  ↳  A  ↳  1 (Arduino Uno R3) ↳  
1 int trig=2;  
2 int eco=3;  
3 int led1=4;  
4 int led2=6;  
5 int led3=7;  
6 int t;  
7 int d;  
8 int buz=5;  
9  
10 void setup()  
11 {  
12   pinMode(led1, OUTPUT);  
13   pinMode(led2, OUTPUT);  
14   pinMode(led3, OUTPUT);  
15   pinMode(trig, OUTPUT);  
16   pinMode(buz, OUTPUT);  
17  
18   pinMode(eco, INPUT);  
19   Serial.begin(9600);  
20  
21  
22 }  
23  
24 void loop()  
25 {  
26   digitalWrite(trig, HIGH);  
27   delay(1);  
28   digitalWrite(trig, LOW);  
29  
30   Monitor en serie  
31  
32   if(d<50)  
33   {  
34     Serial.println(d);  
35     Serial.println("cm");  
36     delay(50);  
37  
38     if(d<50)  
39     {  
40       digitalWrite(led1, HIGH);  
41       digitalWrite(led2, HIGH);  
42       digitalWrite(led3, HIGH);  
43     }  
44     else if(d<100)  
45     {  
46       digitalWrite(led1, HIGH);  
47       digitalWrite(led2, HIGH);  
48       digitalWrite(led3, LOW);  
49     }  
50  
51     else if(d<150)  
52     {  
53       digitalWrite(led1, HIGH);  
54       digitalWrite(led2, LOW);  
55       digitalWrite(led3, LOW);  
56     }  
57     else  
58     {  
59       digitalWrite(led1, LOW);  
60       digitalWrite(led2, LOW);  
61       digitalWrite(led3, LOW);  
62     }  
63  
64   Monitor en serie
```

EJERCICIO #33

EJERCICIO CON SENSOR DE DISTANCIA ULTRASONICO CON LEDS Y BUZZER EN ARDUINO.

- **DIAGRAMA**



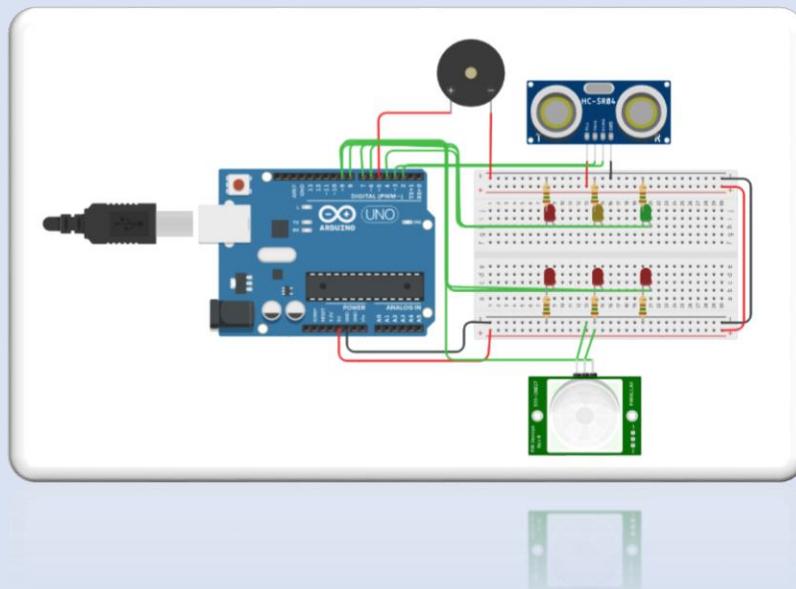
- **CÓDIGO**

```
Texto          1 (Arduino Uno R3)
1 int trig=2;
2 int eco=3;
3 int led1=4;
4 int led2=5;
5 int led3=6;
6 int t;
7 int d;
8 int buz=7;
9
10 void setup()
11 {
12   pinMode(led1, OUTPUT);
13   pinMode(led2, OUTPUT);
14   pinMode(led3, OUTPUT);
15   pinMode(trig, OUTPUT);
16   pinMode(buz, OUTPUT);
17
18   pinMode(eco, INPUT);
19   Serial.begin(9600);
20
21
22 }
23
24 void loop()
25 {
26   digitalWrite(trig, HIGH);
27   delay(1);
28   digitalWrite(trig, LOW);
29
30   t= pulseIn(eco,HIGH);
31   d= t/58.2;
32
33   Serial.println(d);
34   Serial.println("cm");
35   delay(50);
36
37   if(d<50)
38   {
39     digitalWrite(led1, HIGH);
40     digitalWrite(led2, HIGH);
41     digitalWrite(led3, HIGH);
42     tone(buz,1000,10);
43   }
44   else if(d<100)
45   {
46     digitalWrite(led1, HIGH);
47     digitalWrite(led2, HIGH);
48     digitalWrite(led3, LOW);
49     noTone(buz);
50   }
51
52   else if(d<150)
53   {
54     digitalWrite(led1, HIGH);
55     digitalWrite(led2, LOW);
56     digitalWrite(led3, LOW);
57     noTone(buz);
58   }
59   else
60   {
61     digitalWrite(led1, LOW);
62     digitalWrite(led2, LOW);
63     digitalWrite(led3, LOW);
64     noTone(buz);
65   }
66
67   }
68
69   }
70
71
72
73 }
```

EJERCICIO #34

EJERCICIO CON SENSOR DE DISTANCIA ULTRASONICO, SENSOR PIR Y BUZZER CON LEDS EN ARDUINO.

- **DIAGRAMA**



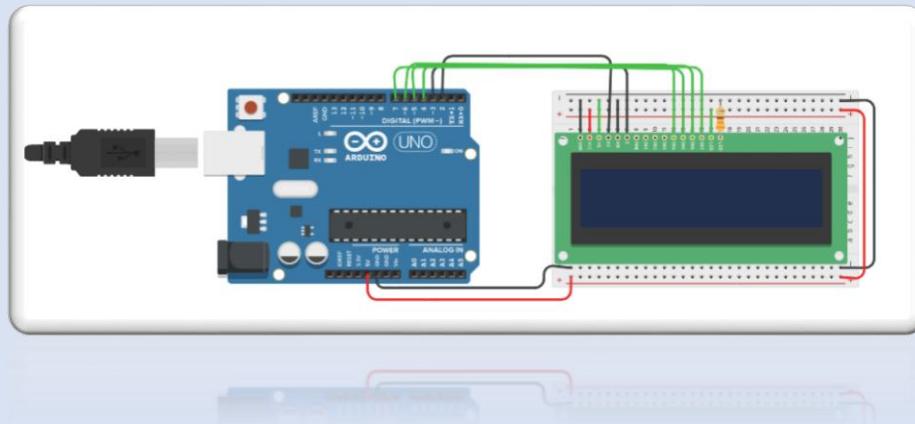
- **CÓDIGO**

```
Texto 1 (Arduino Uno R3)
1 int trig=2;
2 int eco=3;
3 int led1=4;
4 int led3=6;
5 int led2=7;
6 int leds=9;
7 int pir=8;
8 int dt;
9 int t;
10 int d;
11 int buz=5;
12
13
14 void setup()
15 {
16     pinMode(led1, OUTPUT);
17     pinMode(led2, OUTPUT);
18     pinMode(led3, OUTPUT);
19     pinMode(leds, OUTPUT);
20
21     pinMode(trig, OUTPUT);
22     pinMode(buz, OUTPUT);
23     pinMode(pir, INPUT);
24     pinMode(eco, INPUT);
25     serial.begin(9600);
26
27
28
29 }
30
31 Monitor en serie
32
33 void loop()
34 {
35     dt= digitalRead(pir);
36     digitalWrite(trig, HIGH);
37     delay(1);
38     digitalWrite(trig, LOW);
39     t= pulseIn(eco,HIGH);
40     d= t/58.2;
41
42     Serial.println(d);
43     Serial.println("cm");
44     delay(50);
45     if(dt==HIGH)
46     {
47         digitalWrite(leds, HIGH);
48     }
49     else
50     {
51         digitalWrite(leds, LOW);
52     }
53
54     if(d<50)
55     {
56         digitalWrite(led1, HIGH);
57         digitalWrite(led2, HIGH);
58         digitalWrite(led3, HIGH);
59         tone(buz,1000.10);
60     }
61
62     else if(d<100)
63     {
64         digitalWrite(led1, HIGH);
65         digitalWrite(led2, HIGH);
66         digitalWrite(led3, LOW);
67         noTone(buz);
68     }
69
70     else if(d<150)
71     {
72         digitalWrite(led1, HIGH);
73         digitalWrite(led2, LOW);
74         digitalWrite(led3, LOW);
75         noTone(buz);
76     }
77
78     else
79     {
80         digitalWrite(led1, LOW);
81         digitalWrite(led2, LOW);
82         digitalWrite(led3, LOW);
83
84
85         noTone(buz);
86     }
87
88
89 }
90
91 Monitor en serie
92
93
94
95 }
```

EJERCICIO #35

EJERCICIO CON LCD 16X2 EN ARDUINO.

- **DIAGRAMA**



- **CÓDIGO**

```
Texto 1 (Arduino Uno R3)

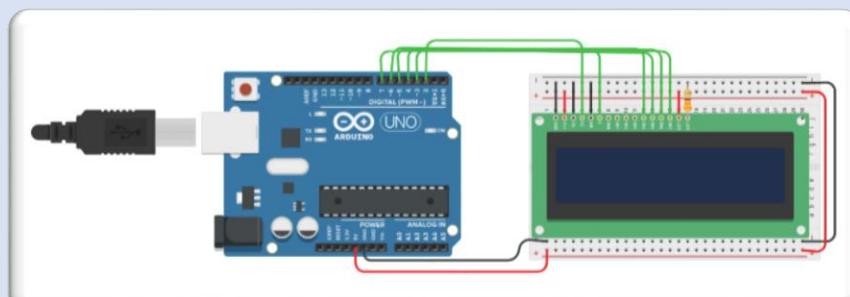
1 #include <LiquidCrystal.h>
2
3 LiquidCrystal a(2, 3, 4, 5, 6, 7);
4
5 void setup()
6 {
7     a.begin (16,2);
8 }
9
10 void loop()
11 {
12     a.setCursor(6,0),
13     a.print("hola");
14     delay(2000);
15     a.clear();
16
17     a.setCursor(3,1),
18     a.print("Aprendiendo");
19     delay(2000);
20     a.clear();
21
22
23     a.setCursor(3,0),
24     a.print("A programar");
25     delay(2000);
26     a.clear();
27
28
29     a.setCursor(2,1),
30     a.print("Con el mejor");
31     delay(2000);
32     a.clear();
33
34
35     a.setCursor(2,0),
36     a.print("Mister Henry");
37     delay(2000);
38     a.clear();
39
40
41
42 }
```

The code uses the LiquidCrystal library to control a 16x2 LCD. It initializes the LCD with `a.begin (16,2);`. In the `loop()` function, it prints "hola" at cursor position (6,0), "Aprendiendo" at (3,1), "A programar" at (3,0), and "Con el mejor" at (2,1). It then prints "Mister Henry" at (2,0). Each message is displayed for 2 seconds before clearing the screen.

EJERCICIO #36

EJERCICIO CON LCD 16X2 EN ARDUINO.

- **DIAGRAMA**



- **CÓDIGO**

The screenshot shows the Arduino IDE interface with the code for the LCD 16x2 exercise. The code uses the LiquidCrystal library to print "holo" to the LCD screen. The code is as follows:

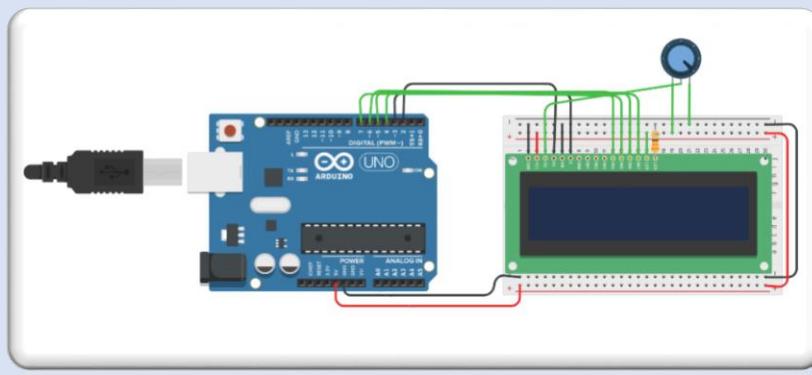
```
1 #include <LiquidCrystal.h>
2
3 LiquidCrystal a(2, 3, 4, 5, 6, 7);
4
5 void setup()
6 {
7     a.begin (16,2);
8 }
9
10 void loop()
11 {
12     a.setCursor(6,0),
13     a.print("holo");
14     delay(2000);
15     a.clear();
16
17
18 }
```

The status bar at the bottom of the IDE indicates "Monitor en serie".

EJERCICIO #37

EJERCICIO CON LCD 16X2 Y POTENCIOMETRO EN ARDUINO.

- **DIAGRAMA**



- **CÓDIGO**

```
Texto          1 (Arduino Uno R3) ▾

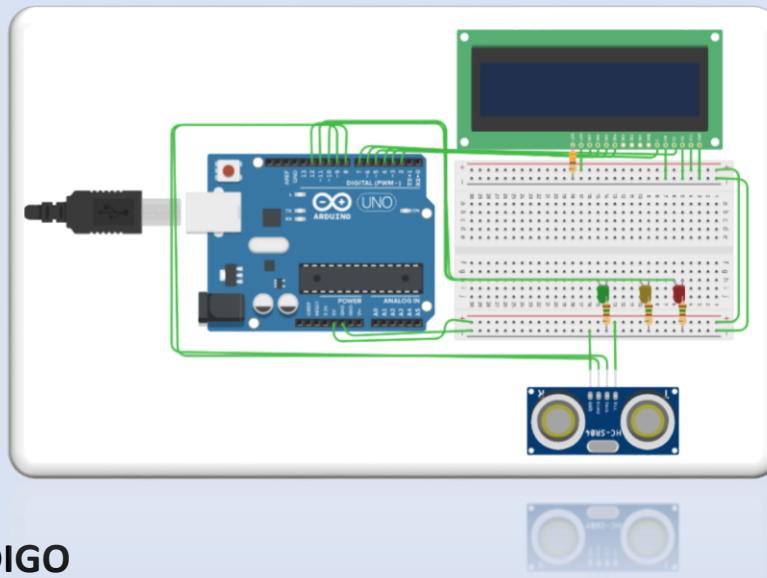
1 #include <LiquidCrystal.h>
2
3 LiquidCrystal a(2, 3, 4, 5, 6, 7);
4
5 void setup()
6 {
7     a.begin (16,2);
8 }
9
10 void loop()
11 {
12     a.setCursor(4,0),
13     a.print("eric");
14     delay(2000);
15     a.clear();
16
17     a.setCursor(5,1),
18     a.print("Zuberbuhler");
19     delay(2000);
20     a.clear();
21
22
23     a.setCursor(5,0),
24     a.print("Leon");
25     delay(2000);
26     a.clear();
27
28
29     a.setCursor(7,1),
30     a.print("3ro");
31     delay(2000);
32     a.clear();
33
34
35     a.setCursor(3,0),
36     a.print("Informatica");
37     delay(2000);
38     a.clear();
39
40
41
42 }
```

Monitor en serie

EJERCICIO #38

EJERCICIO CON SENSOR DE DISTANCIA ULTRASONICO, SENSOR PIR CON LEDS EN ARDUINO.

- **DIAGRAMA**



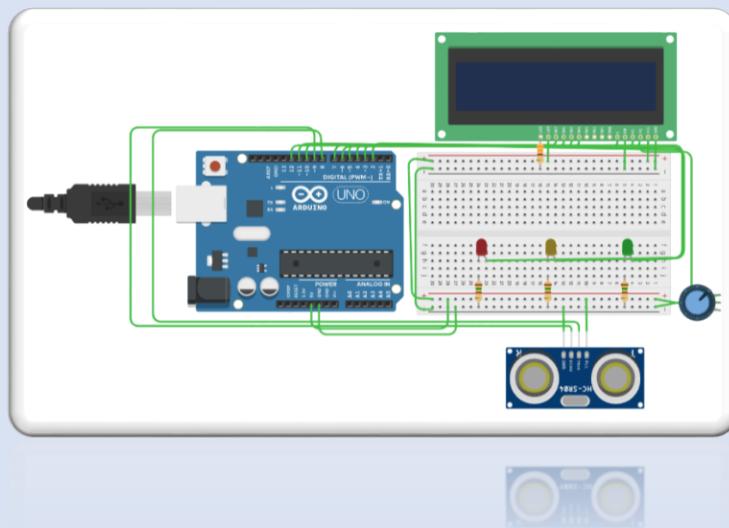
- CÓDIGO

```
1 #include <LiquidCrystal.h>
2 LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
3 int eco= 9;
4 int trig=8;
5 int t;
6 int d;
7 int l1=10;
8 int l2=11;
9 int l3=12;
10 int l4=13;
11 int l5=1;
12
13 void setup()
14 {
15     pinMode(trig, OUTPUT);
16     pinMode(eco, INPUT);
17     pinMode(l1, OUTPUT);
18     pinMode(l2, OUTPUT);
19     pinMode(l3, OUTPUT);
20     pinMode(l4, OUTPUT);
21     pinMode(l5, OUTPUT);
22
23     lcd.begin(16,2);
24 }
25
26 void loop()
27 {
28     digitalWrite(trig,HIGH);
29
30     if(d<50)
31     {
32         digitalWrite(l1,HIGH);
33         digitalWrite(l2,HIGH);
34         digitalWrite(l3,HIGH);
35
36         lcd.setCursor(0,0);
37         lcd.print("Todos encendidos");
38         lcd.setCursor(6,1);
39         lcd.print(d);
40         lcd.print("cm");
41         delay(500);
42         lcd.clear();
43     }
44     else if(d<100)
45     {
46         digitalWrite(l1,HIGH);
47         digitalWrite(l2,HIGH);
48         digitalWrite(l3,LOW);
49
50         lcd.setCursor(0,0);
51         lcd.print("Verde y amarillo");
52         lcd.setCursor(6,1);
53         lcd.print(d);
54         lcd.print("cm");
55         delay(500);
56         lcd.clear();
57     }
58     else if(d<150)
59     {
60         digitalWrite(l1,LOW);
61         digitalWrite(l2,LOW);
62         digitalWrite(l3,LOW);
63
64         lcd.setCursor(0,0);
65         lcd.print("Rojo encendidos");
66         lcd.setCursor(6,1);
67         lcd.print(d);
68         lcd.print("cm");
69         delay(500);
70         lcd.clear();
71     }
72     else
73     {
74         digitalWrite(l1,LOW);
75         digitalWrite(l2,LOW);
76         digitalWrite(l3,LOW);
77
78         lcd.setCursor(4,1);
79         lcd.print("Apagados");
80         delay(500);
81         lcd.clear();
82     }
83 }
```

EJERCICIO #39

EJERCICIO CON SENSOR DE DISTANCIA ULTRASONICO, SENSOR PIR Y BUZZER CON LEDS EN ARDUINO.

- **DIAGRAMA**

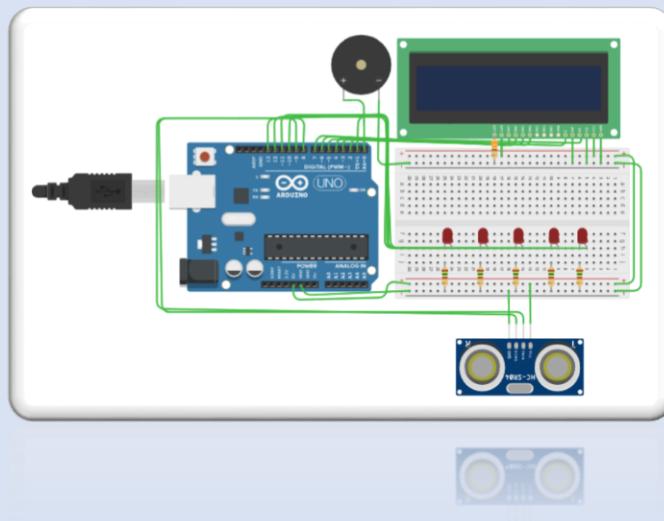


- CÓDIGO

EJERCICIO #40

EJERCICIO CON SENSOR DE DISTANCIA ULTRASONICO CON LEDS EN ARDUINO.

- **DIAGRAMA**



- **CÓDIGO**

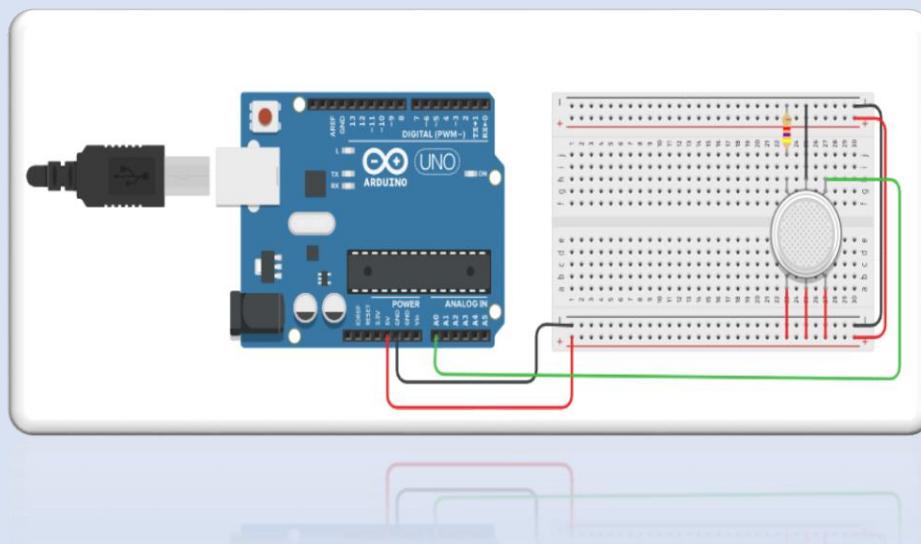
```
Texto          1 (Arduino Uno R3)

1 #include <LiquidCrystal.h>
2 LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
3 int eco= 9;
4 int trig=8;
5 int t;
6 int d;
7 int l1=10;
8 int l2=11;
9 int l3=12;
10 int l4=13;
11 int l5=1;
12 int buz=0;
13
14 void setup()
15 {
16     pinMode(trig, OUTPUT);
17     pinMode(eco, INPUT);
18     pinMode(buz, OUTPUT);
19     pinMode(l1, OUTPUT);
20     pinMode(l2, OUTPUT);
21     pinMode(l3, OUTPUT);
22     pinMode(l4, OUTPUT);
23     pinMode(l5, OUTPUT);
24
25
26     lcd.begin(16,2);
27 }
28
29
30 void loop()
31 {
32     digitalWrite(trig,HIGH);
33     delay(1);
34     digitalWrite(trig,LOW);
35
36     t= pulseIn(eco,HIGH);
37     d= t/58.2;
38
39     if(d<50)
40     {
41         digitalWrite(l1,HIGH);
42         digitalWrite(l2,HIGH);
43         digitalWrite(l3,HIGH);
44         digitalWrite(l4,HIGH);
45         digitalWrite(l5,HIGH);
46         tone(buz,2050);
47         lcd.setCursor(4,1);
48         lcd.print("Encendido");
49         delay(500);
50         lcd.clear();
51     }
52     else
53     {
54         digitalWrite(l1,LOW);
55         digitalWrite(l2,LOW);
56         digitalWrite(l3,LOW);
57         digitalWrite(l4,LOW);
58         digitalWrite(l5,LOW);
59         lcd.setCursor(4,1);
60     }
61 }
```

Ejercicio # 41

EJERCICIO CON SENSOR DE GAS EN ARDUINO.

- DIAGRAMA



- CÓDIGO

```
int gas;

void setup()
{
    pinMode(A0, INPUT);
    Serial.begin(9600);
}

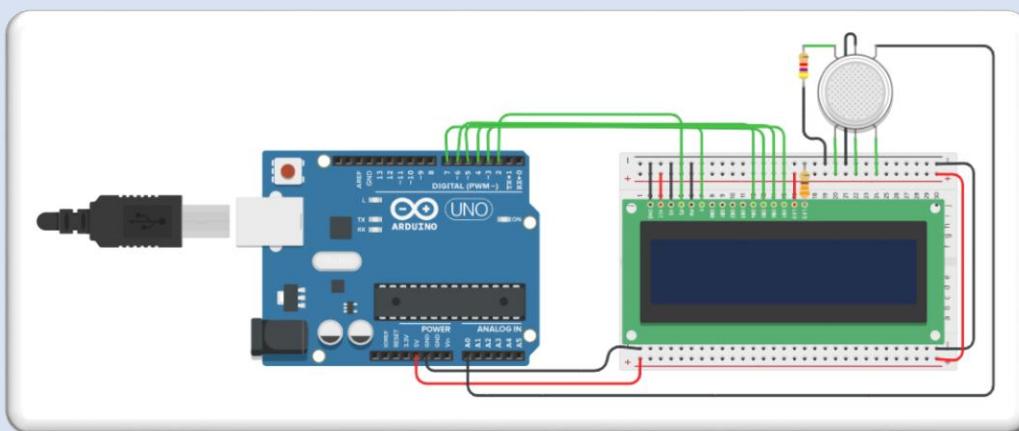
void loop()
{
    gas=analogRead(A0);
    Serial.println(gas);

}
```

Ejercicio #42

EJERCICIO CON SENSOR DE GAS CON LCD 16X2 EN ARDUINO.

- **DIAGRAMA**



- **CÓDIGO**

```
#include <LiquidCrystal.h>

LiquidCrystal e(2, 3, 4, 5, 6, 7);
int gas;

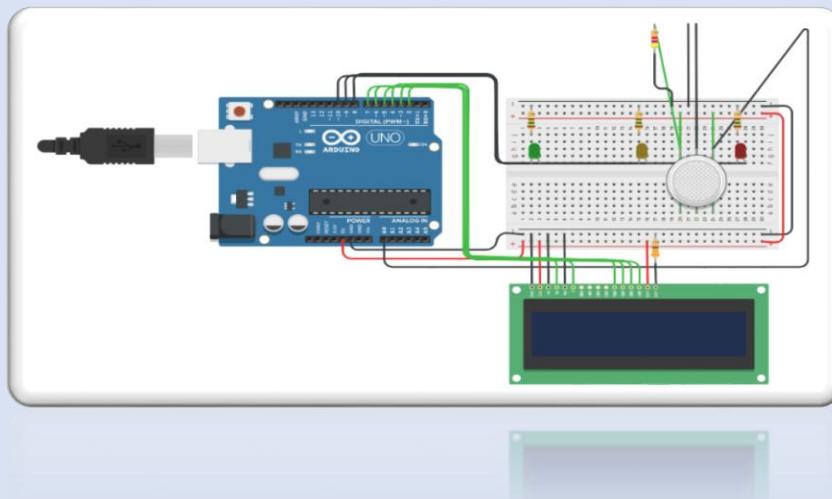
void setup()
{
    e.begin (16,2);
    pinMode(A0, INPUT);
}

void loop()
{
    gas= analogRead(A0);
    e.setCursor(6,0),
    e.print(gas);
    delay(1000);
    e.clear();
}
```

Ejercicio # 43

EJERCICIO CON SENSOR DE GAS, LCD 16X2 CON LEDS EN ARDUINO.

- DIAGRAMA



- CÓDIGO

```
/* Ejercicio #43 con Sensor de Gas y LCD 16x2 */

LiquidCrystal e(2, 3, 4, 5, 6, 7);
int gas;
int r=10;
int v=8;
int a=9;

void setup()
{
    e.begin (16,2);
    pinMode(A0,INPUT);
}

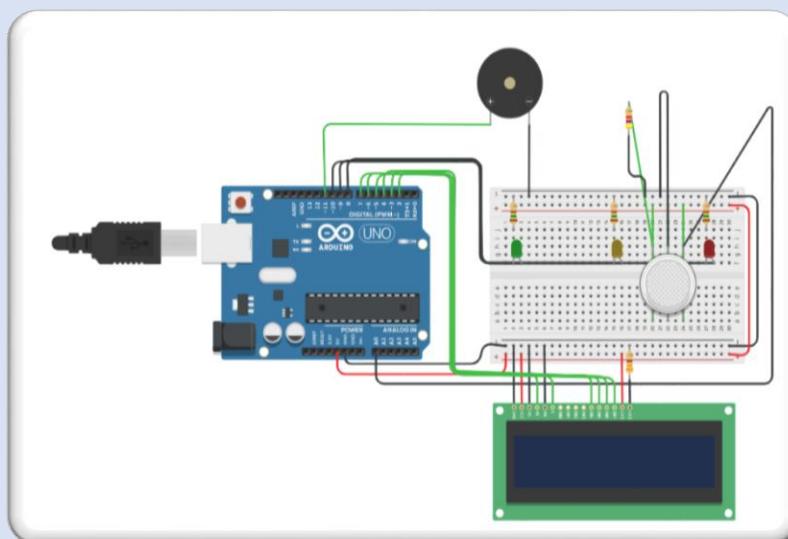
void loop()
{
    gas= analogRead(A0);
    e.setCursor(6,0);
    e.print(gas);
    delay(1000);
    e.clear();

    if(gas>506)
    {
        digitalWrite(r,HIGH);
        digitalWrite(a,LOW);
        digitalWrite(v,LOW);
        e.setCursor(6,0);
        e.print(gas);
        e.setCursor(4,1);
        e.print("peligro");
        delay(1000);
        e.clear();
    }
    else if(gas>436)
    {
        digitalWrite(a,HIGH);
        digitalWrite(r,LOW);
        digitalWrite(v,LOW);
        e.setCursor(6,0);
        e.print(gas);
        e.setCursor(2,1);
        e.print("Advertencia");
        delay(1000);
        e.clear();
    }
    else if(gas>306)
    {
        digitalWrite(v,HIGH);
        digitalWrite(a,LOW);
        digitalWrite(r,LOW);
        e.setCursor(6,0);
        e.print(gas);
        e.setCursor(0,1);
        e.print("Fuera de peligro");
        delay(1000);
        e.clear();
    }
}
```

Ejercicio # 44

EJERCICIO CON SENSOR DE GAS, LCD 16X2, BUZZER CON LEDS EN ARDUINO.

- **DIAGRAMA**



- **CÓDIGO**

```
#include <LiquidCrystal.h>

LiquidCrystal e(2, 3, 4, 5, 6, 7);
int gas;
int r=10;
int v=8;
int a=9;
int te=11;

void setup()
{
  e.begin (16,2);
  pinMode(A0,INPUT);
  pinMode(te,OUTPUT);
}

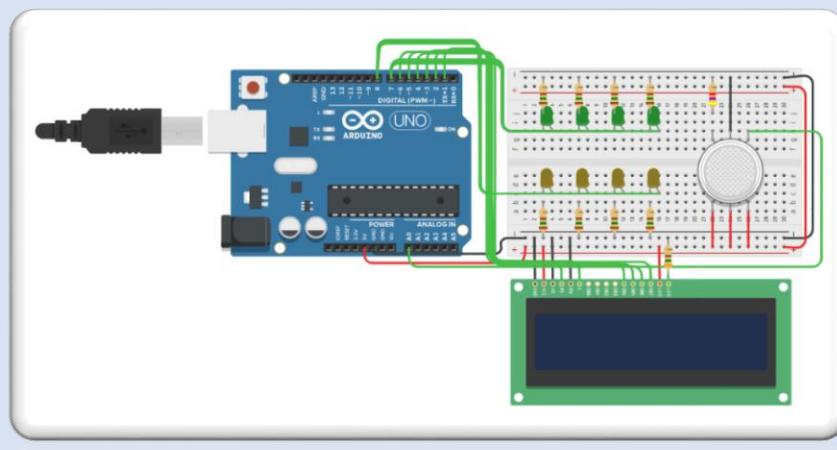
void loop()
{
  gas= analogRead(A0);
  e.setCursor(6,0);
  e.print(gas);
  delay(1000);
  e.clear();

  if(gas>506)
  {
    digitalWrite(r,HIGH);
    digitalWrite(a,LOW);
    digitalWrite(v,LOW);
    tone(te,2050);
    e.setCursor(6,0);
    e.print(gas);
    e.setCursor(4,1);
    e.print("peligro");
    delay(1000);
    e.clear();
  }
  else if(gas>436)
  {
    digitalWrite(a,HIGH);
    digitalWrite(r,LOW);
    digitalWrite(v,LOW);
    tone(te,2050);
    e.setCursor(6,0);
    e.print(gas);
    e.setCursor(2,1);
    e.print("Advertencia");
    delay(1000);
    e.clear();
  }
  else if(gas>306)
  {
    digitalWrite(v,HIGH);
    digitalWrite(a,LOW);
    digitalWrite(r,LOW);
    noTone(te);
    e.setCursor(6,0);
    e.print(gas);
    e.setCursor(0,1);
    e.print("Fuera de peligro");
    delay(1000);
    e.clear();
  }
}
```

Ejercicio # 45

EJERCICIO CON SENSOR DE GAS Y LCD 16X2 CON LEDS EN ARDUINO.

- DIAGRAMA



- CÓDIGO

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(1,2,3,4,5,6);
int gas;
int lv=7;
int la=8;

void setup()
{
    pinMode(lv, OUTPUT);
    pinMode(la, OUTPUT);
    lcd.begin(16,2);
    pinMode(A0, INPUT);
}

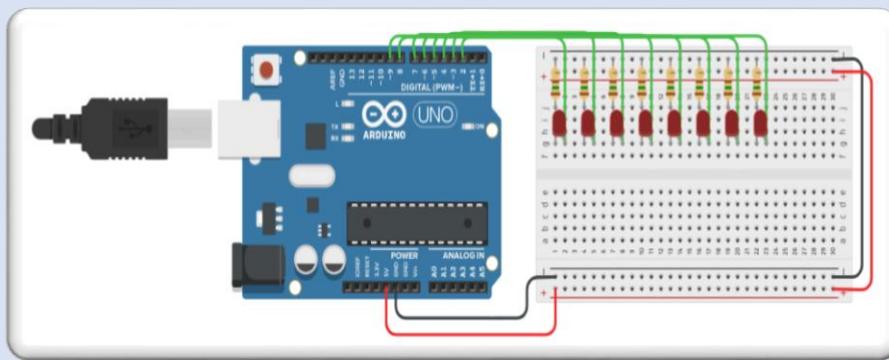
void loop()
{
    gas= analogRead(A0);

    if(gas<10)
    {
        digitalWrite(lv, HIGH);
        digitalWrite(la, LOW);
        delay(1000);
        lcd.setCursor(3,0);
        lcd.print("led verde ");
        lcd.setCursor(3,1);
        lcd.print("encendido");
        delay(1000);
        lcd.clear();
    }
    else
    {
        digitalWrite(la, HIGH);
        digitalWrite(lv, LOW);
        delay(1000);
        lcd.setCursor(1,0);
        lcd.print("led amarillo ");
        lcd.setCursor(3,1);
        lcd.print("encendido");
        delay(1000);
        lcd.clear();
    }
}
```

Ejercicio # 46

EJERCICIO CON SENSOR DE GAS Y LCD 16X2 CON LEDS EN ARDUINO.

- **DIAGRAMA**



- **CÓDIGO**

```
int d=300;
void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
}

void loop()
{
  digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
  delay(d);

  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
  delay(d);

  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, HIGH);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
  delay(d);

  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, HIGH);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
  delay(d);

  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
  delay(d);

  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, HIGH);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
  delay(d);

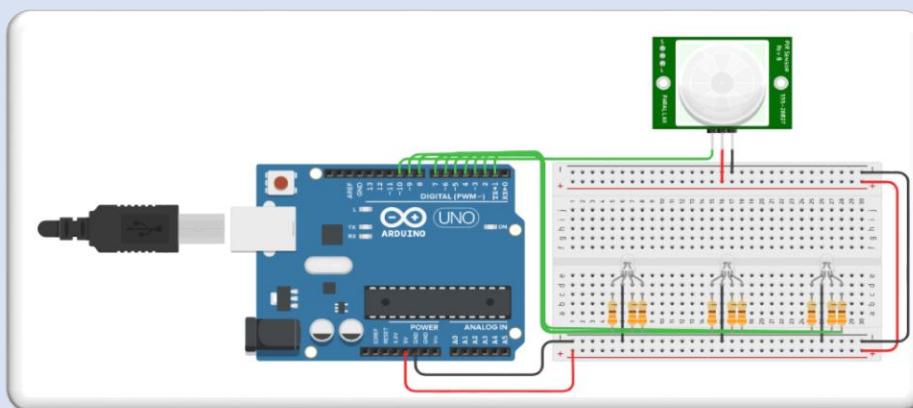
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, HIGH);
  digitalWrite(9, LOW);
  delay(d);

  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, HIGH);
  delay(d);
}
```

Ejercicio # 47

EJERCICIO CON SENSOR PIR Y LED RGB EN ARDUINO.

- DIAGRAMA



- CÓDIGO

```
int r1=1;
int a1=2;
int v1=3;

int r2=4;
int a2=5;
int v2=6;

int r3=7;
int a3=8;
int v3=9;

int pir=10;
int dt;
void setup()
{
    pinMode(pir, INPUT);
    pinMode(r1, OUTPUT);
    pinMode(a1, OUTPUT);
    pinMode(v1, OUTPUT);
}

void loop()
{
    dt=digitalRead(pir);

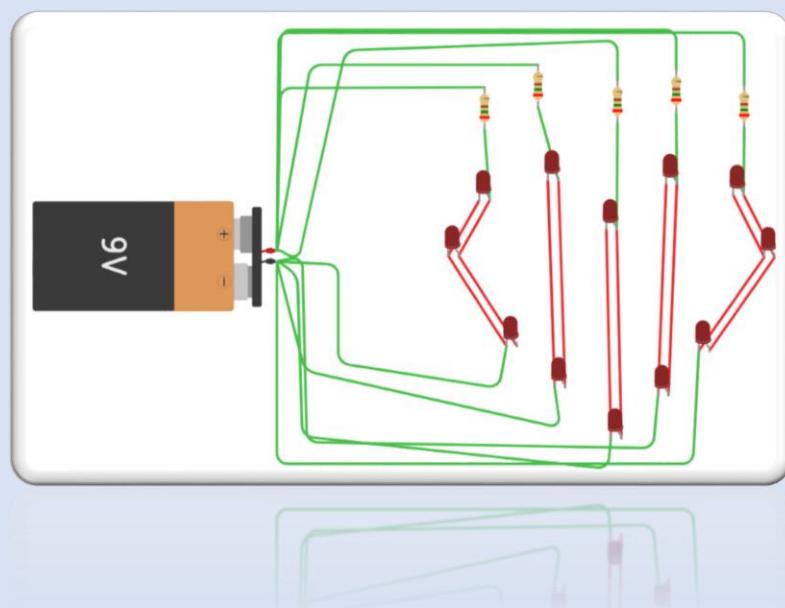
    if(dt==HIGH)
    {
        analogWrite(r1, 10);
        analogWrite(a1, 100);
        analogWrite(v1, 200);
        analogWrite(r2, 100);
        analogWrite(a2, 59);
        analogWrite(v2, 200);
        analogWrite(r3, 200);
        analogWrite(a3, 100);
        analogWrite(v3, 36);
        delay(1000);

        analogWrite(r1, 12);
        analogWrite(a1, 200);
        analogWrite(v1, 15);
        analogWrite(r2, 200);
        analogWrite(a2, 12);
        analogWrite(v2, 200);
        analogWrite(r3, 35);
        analogWrite(a3, 21);
        analogWrite(v3, 152);
        delay(1000);
    }
    else
    {
        analogWrite(r1, 0);
        analogWrite(a1, 0);
        analogWrite(v1, 0);
        analogWrite(r2, 0);
        analogWrite(a2, 0);
        analogWrite(v2, 0);
        analogWrite(r3, 0);
        analogWrite(a3, 0);
        analogWrite(v3, 0);
        delay(1000);
    }
}
```

Ejercicio # 48

UNA PILA DE 9V CON LEDS EN SERIE FORMANDO CORAZON CON SU RESISTENCIAS.

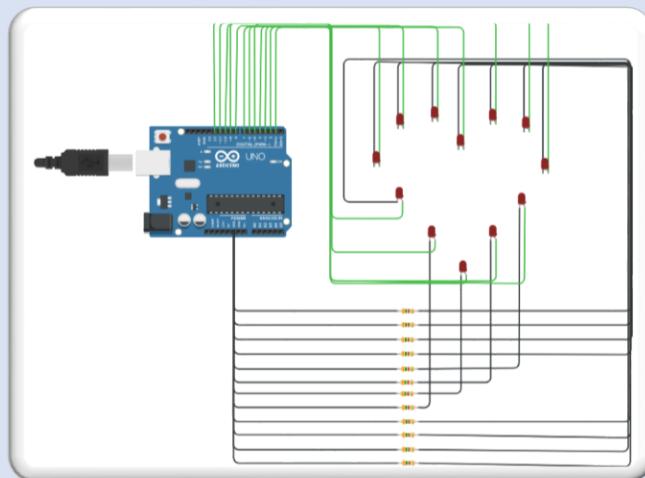
- **DIAGRAMA**



Ejercicio # 49

UN ARDUINO UNO, CONECTADOS CON LEDS FORMANDO CORAZON Y SUS RESISTENCIAS INDIVIDUALES.

- DIAGRAMA



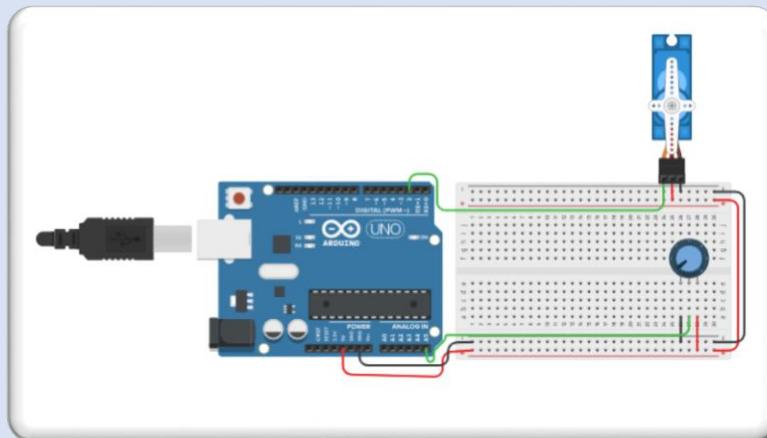
- CÓDIGO

```
....  
void setup()  
{  
    pinMode(1, OUTPUT);  
    pinMode(2, OUTPUT);  
    pinMode(3, OUTPUT);  
    pinMode(4, OUTPUT);  
    pinMode(5, OUTPUT);  
    pinMode(6, OUTPUT);  
    pinMode(7, OUTPUT);  
    pinMode(8, OUTPUT);  
    pinMode(9, OUTPUT);  
    pinMode(10, OUTPUT);  
    pinMode(11, OUTPUT);  
    pinMode(12, OUTPUT);  
  
}  
  
void loop()  
{  
    digitalWrite(1, HIGH);  
    digitalWrite(2, LOW);  
    digitalWrite(3, LOW);  
    digitalWrite(4, LOW);  
    digitalWrite(5, LOW);  
    digitalWrite(6, LOW);  
    digitalWrite(7, LOW);  
    digitalWrite(8, LOW);  
    digitalWrite(9, LOW);  
    digitalWrite(10, LOW);  
    digitalWrite(11, LOW);  
    digitalWrite(12, LOW);  
    delay(d);  
  
    digitalWrite(1, HIGH);  
    digitalWrite(2, HIGH);  
    digitalWrite(3, HIGH);  
    digitalWrite(4, LOW);  
    digitalWrite(5, LOW);  
    digitalWrite(6, LOW);  
    digitalWrite(7, LOW);  
    digitalWrite(8, LOW);  
    digitalWrite(9, LOW);  
    digitalWrite(10, LOW);  
    digitalWrite(11, LOW);  
    digitalWrite(12, LOW);  
    delay(d);  
  
    digitalWrite(1, HIGH);  
    digitalWrite(2, HIGH);  
    digitalWrite(3, HIGH);  
    digitalWrite(4, HIGH);  
    digitalWrite(5, HIGH);  
    digitalWrite(6, HIGH);  
    digitalWrite(7, HIGH);  
    digitalWrite(8, HIGH);  
    digitalWrite(9, LOW);  
    digitalWrite(10, LOW);  
    digitalWrite(11, LOW);  
    digitalWrite(12, LOW);  
    delay(d);  
  
    digitalWrite(1, HIGH);  
    digitalWrite(2, HIGH);  
    digitalWrite(3, HIGH);  
    digitalWrite(4, HIGH);  
    digitalWrite(5, HIGH);  
    digitalWrite(6, HIGH);  
    digitalWrite(7, HIGH);  
    digitalWrite(8, HIGH);  
    digitalWrite(9, HIGH);  
    digitalWrite(10, HIGH);  
    digitalWrite(11, HIGH);  
    digitalWrite(12, HIGH);  
    delay(d);  
}
```

Ejercicio # 50

EJERCICIO CON UN SERVOMOTOR MAS POTENCIOMETRO EN ARDUINO

- **Diagrama:**



- **CÓDIGO**

```
#include <Servo.h>

Servo motor;//Asignacion de nombre al motor

int grado;

void setup()
{
    Serial.begin (9600);
    motor.attach(2);//Asignacion pin

}

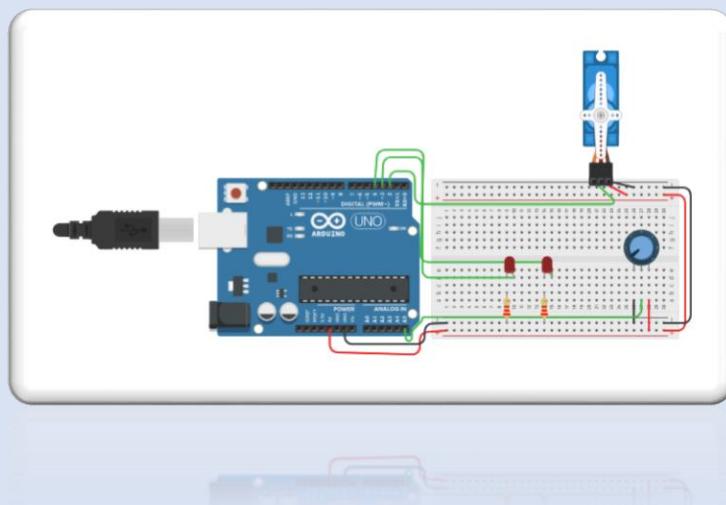
void loop()
{
    grado = map(analogRead(A5),0,1023,0,180);
    Serial.println(grado);
    motor.write(grado);

}
```

Ejercicio # 51

EJERCICIO CON UN SERVOMOTOR Y UN POTENCIOMETRO CON 2 LEDS MAS SUS RESISTENCIAS INDIVIDUALES EN ARDUINO.

- **DIAGRAMA**



- **CÓDIGO**

```
#include <Servo.h>

Servo motor;//Asignacion de nombre al motor

int grado;+
int

void setup()
{
    Serial.begin (9600);
    motor.attach(2);//Asignacion pin

}

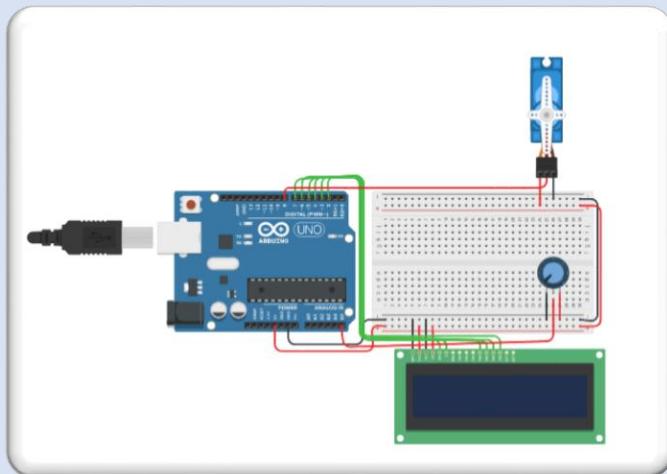
void loop()
{
    grado = map(analogRead(A5),0,1023,0,180);
    Serial.println(grado);
    motor.write(grado);

}
```

Ejercicio # 53

EJERCICIO CON UN LCD 16X2 Y UN SERVOMOTOR EN ARDUINO.

- DIAGRAMA



- CÓDIGO

```
#include <LiquidCrystal.h>
LiquidCrystal lcd (2,3,4,5,6,7);
#include <Servo.h>
Servo motor
int d = 0;

void setup()
{
  lcd.begin (16,2);
  d = (0);

}

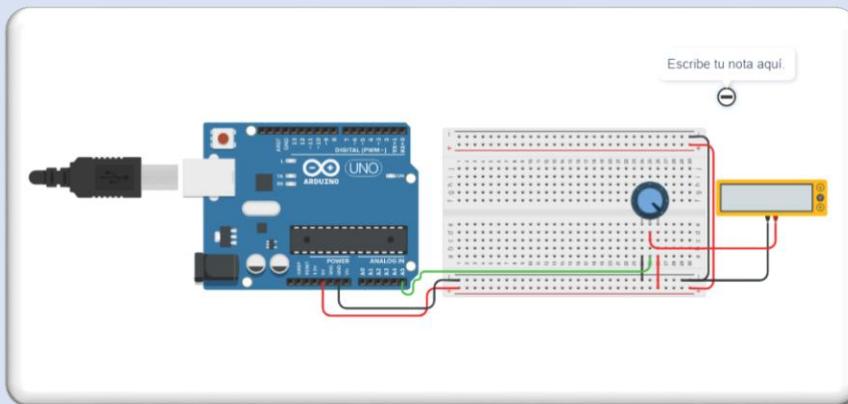
void loop()
{
  map = (analogRead(A5)0,1023,1,180);
  servo motor = d;
  lcd.print(d);

  lcd.print(d);
  (6,1)
```

Ejercicio # 54

EJERCICIO CON UN MULTIMETRO Y UN POTENCIOMETRO EN ARDUINO.

- **DIAGRAMA**



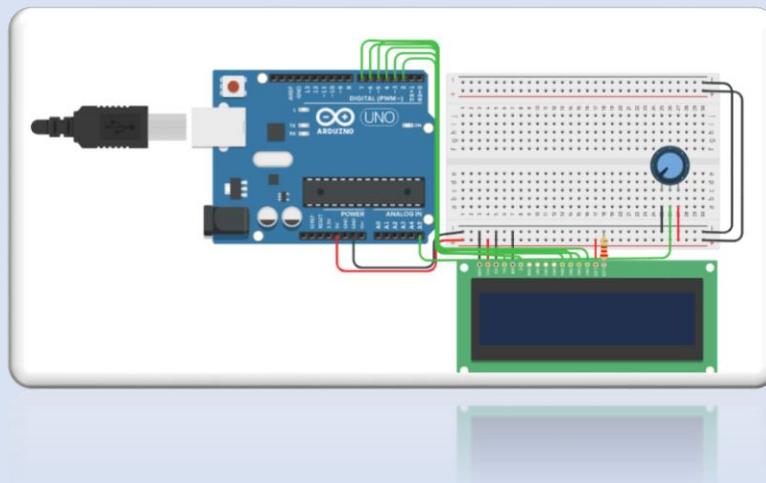
- **CÓDIGO**

```
int pot = 0;  
void setup()  
{  
    Serial.begin(9600);  
}  
  
void loop()  
{  
    pot = analogRead(A5);  
    Serial.println(pot);  
}
```

Ejercicio # 55

EJERCICIO CON UN LCD 16X2 Y POTENCIOMETRO EN ARDUINO.

- DIAGRAMA



- CÓDIGO

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(2,3,4,5,6,7);
int d;

void setup()
{
    lcd.begin (16,2);
}

void loop()
{
    d = map(analogRead(A5),0,1023,1,500);

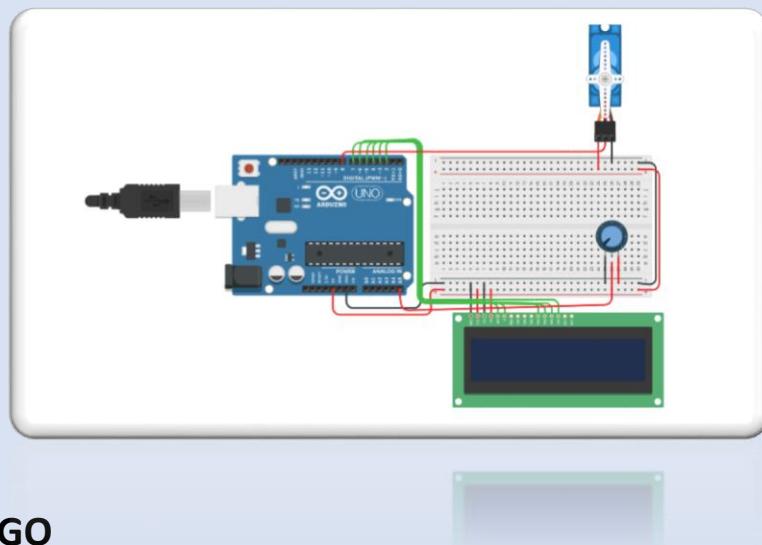
    lcd.setCursor(1,0);
    lcd.print("Potenciómetro");

    lcd.setCursor(2,1);
    lcd.print("valor = ");
    lcd.print(d);
    delay(500);
    lcd.clear();
}
```

Ejercicio # 56

EJERCICIO CON UN LCD 16X2, SERVOMOTOR Y POTENCIOMETRO EN ARDUINO.

- **DIAGRAMA**



- **CÓDIGO**

```
#include <LiquidCrystal.h>
LiquidCrystal lcd (2,3,4,5,6,7);
#include <Servo.h>
Servo motor
int = d ;

void setup()
{
    lcd.begin (16,2);
    d = (0);

}

void loop()
{

    map = (analogRead(A5)0,1023,1,180);
    servo motor = d;
    lcd.print(d);

    lcd.print(d);
    (6,1)
```

Ejercicio # 57

EJERCICIO CON UN KEYPAD 4X4 EN ARDUINO.

- DIAGRAMA



- CÓDIGO

```
#include <Keypad.h>

const byte filas = 4;
const byte columnas = 4;
byte pinesFilas[] = {11,10,9,8};
byte pinesColumnas[] = {7,6,5,4};
char teclas[4][4] = {{'1','2','3','A'},
                     {'4','5','6','B'},
                     {'7','8','9','C'},
                     {'*','0','#','D'}};

Keypad teclado1 = Keypad( makeKeymap(teclas),
                         pinesFilas, pinesColumnas, filas, columnas);
void setup() {

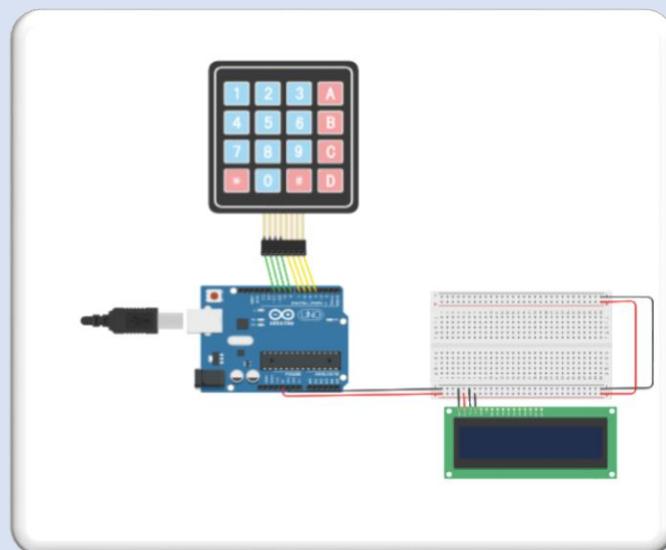
    Serial.begin(9600);
    Serial.println("Teclado 4x4 con Biblioteca Keypad");
    Serial.println();
}

void loop() {
    //Verifica si alguna tecla fue presionada
    char tecla_presionada = teclado1.getKey();
    //Monitor Serial
    if (tecla_presionada)
    {
        Serial.print("Tecla: ");
        Serial.println(tecla_presionada);
    }
}
```

Ejercicio # 58

EJERCICIO CON UN KEYPAD 4X4 Y LCD 16X2 EN ARDUINO.

- DIAGRAMA



- CÓDIGO

```
1 #include <Keypad.h>
2
3 const byte filas = 4;
4 const byte columnas = 4;
5 byte pinesFilas[] = {11,10,9,8};
6 byte pinesColumnas[] = {7,6,5,4};
7 char teclas[4][4] = {{'1','2','3','A'},
8                         {'4','5','6','B'},
9                         {'7','8','9','C'},
0                         {'*','0','#','D'}};
```

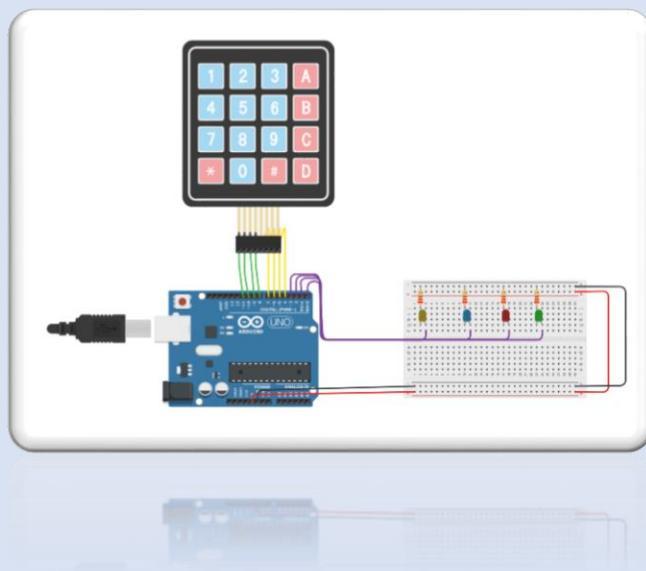
```
void loop() {
    //Verifica si alguna tecla fue presionada
    char tecla_presionada = teclado1.getKey();

    //Monitor Serial
    if (tecla_presionada)
    {
        Serial.print("Tecla: ");
        Serial.println(tecla_presionada);
    }
}
```

Ejercicio # 59

EJERCICIO CON UN KEYPAD 4X4 CON 4 LEDs DE DIFERENTES COLORES, PERO SUS RESISTENCIAS INDIVIDUALES EN ARDUINO.

- DIAGRAMA



- CÓDIGO

```
#include <LiquidCrystal.h>
#include <Keypad.h>
LiquidCrystal lcd(2,3,12,13,A1,A2);
int lz1 = 0 ;
int lz2 = 1 ;
int lz3 = 2 ;
int lz4 = 3 ;
const byte filas = 4;
const byte columnas = 4;
byte pinesFilas[] = {11,10,9,8};
byte pinesColumnas[] = {7,6,5,4};
char teclas[4][4] = {{'1','2','3','A'},
                     {'4','5','6','B'},
                     {'7','8','9','C'},
                     {'*','0','#','D'}};

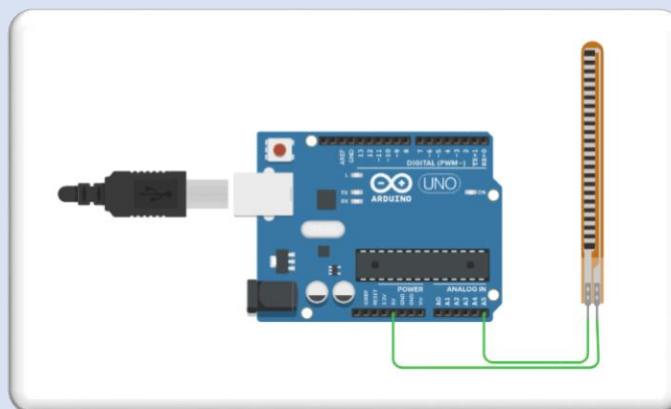
Keypad teclado1 = Keypad( makeKeymap(teclas), pinesFilas, pinesColumnas);

void setup() {
  lcd.begin(16,2);
}
```

Ejercicio # 60

EJERCICIO CON UN SENSOR DE FLEXION EN ARDUINO.

- DIAGRAMA



- CÓDIGO

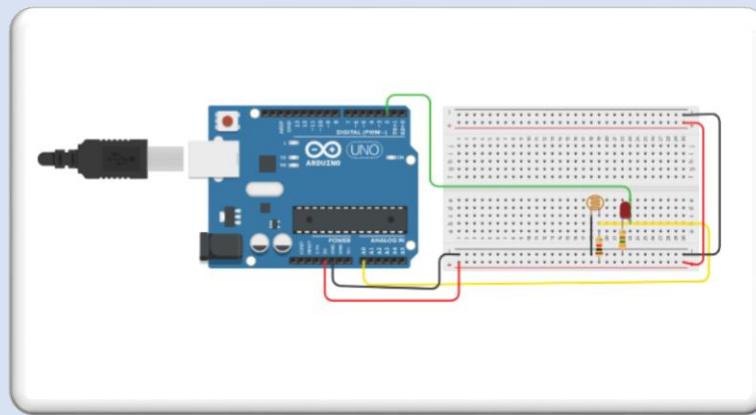
```
int flex= A5;
void setup()
{
    Serial.begin(9600);
    pinMode(flex, OUTPUT);
}

void loop()
{
    Serial.println(flex);
    digitalWrite(flex, HIGH);
    delay(100);
    digitalWrite(flex, LOW);
    delay(100);
}
```

Ejercicio # 61

EJERCICIO CON FOTORRESISTENCIA CON LED Y SU RESISTENCIA EN ARDUINO.

- **DIAGRAMA**



- **CÓDIGO**

```
int ldr=A0;
int LDR=0;
void setup(){
  Serial.begin(9600);
  pinMode(2, OUTPUT);
}
void loop(){
  LDR=analogRead(ldr);
  Serial.println(LDR);
  delay(300);

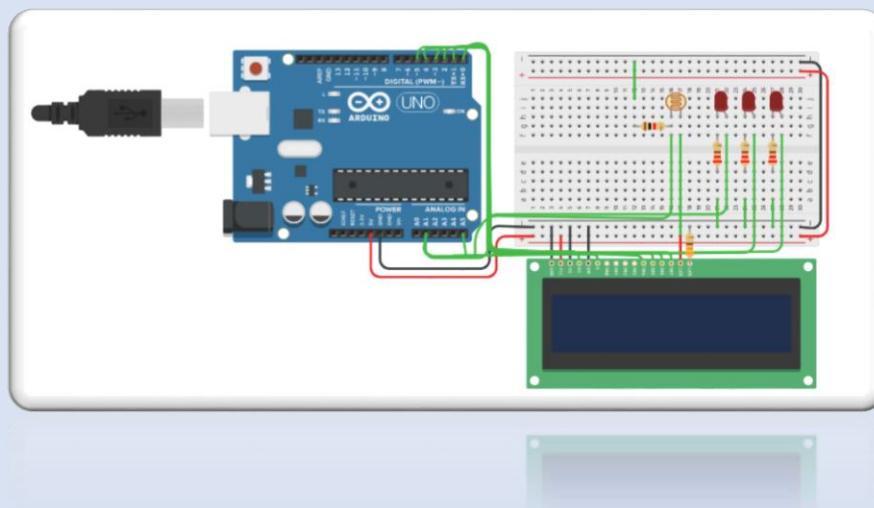
  if(LDR>479)
  {
    digitalWrite(2, HIGH);
  }
  else

  {
    digitalWrite(2, LOW);
  }
}
```

Ejercicio # 62

EJERCICIO CON FOTORRESISTENCIA CON 3 LEDS Y SU RESISTENCIA MAS LCD 16X2 EN ARDUINO.

- DIAGRAMA



- CÓDIGO

```
#include <LiquidCrystal.h>
LiquidCrystal lcd (0,1,2,3,4,5);

int ldr = A5;
int lz = A1;
int d;

void setup()
{
  pinMode(ldr, INPUT);
  pinMode(lz, OUTPUT);
  lcd.begin(16,2);
}

void loop()
{

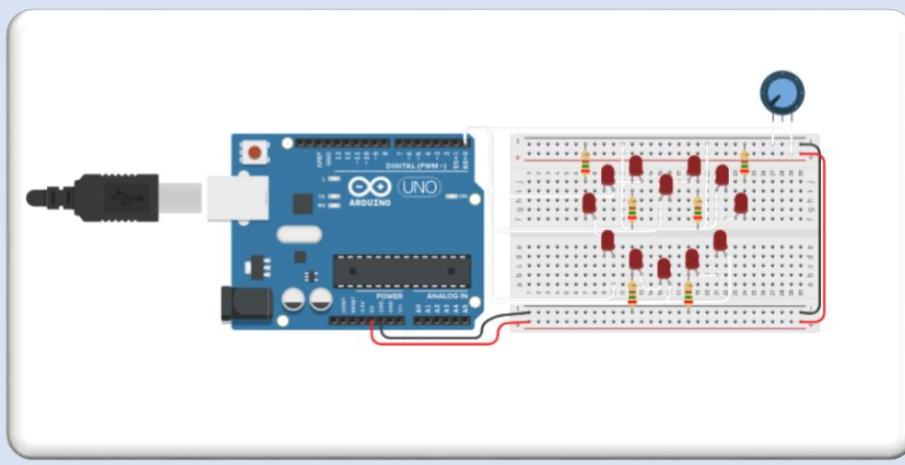
  d = analogRead(A5);
  lcd.print(d);

  if
    d =
  digitalWrite(lz,HIGH);
  Set.cursor(3,1);
  lcd.print("luces encendidas");
}
```

Ejercicio # 63

EJERCICIO CON 4 GRUPOS DE 3 LEDS CON UNA RESISTENCIA DE 150 OHMIOS FORMANDO UN CORAZON Y UN POTENCIOMETRO EN ARDUINO.

- **DIAGRAMA**



- **CÓDIGO**

```
1 #include <LiquidCrystal.h>
2 LiquidCrystal lcd(1,2,3,4,5,6);
3 int d;
4 int lz =0;
5
6
7 void setup()
8 {
9     lcd.begin (16,2);
10    pinMode(lz, OUTPUT);
11 }
12
13 void loop()
14 {
15     d = analogRead(A5);
16
17     if (d == 1023){
18         digitalWrite(lz, HIGH);
19         lcd.setCursor(1,0);
20         lcd.print("Luces");
21         lcd.setCursor(2,1);
22         lcd.print("Encendidas");
23     }
24     else
25         digitalWrite(lz, LOW);
26 }
```

4. ANEXOS COMPONENTES

BATERIA

Las baterías son unos dispositivos especiales para almacenar y generar energía mediante un proceso químico, para luego liberarla en forma de corriente eléctrica continua. Cualquier tipo de batería contiene un electrodo negativo y uno positivo dentro de una cantidad de electrolito líquido, y todo en conjunto se encuentra en un recipiente.



CONEXION

El terminal positivo (+) es un conector circular y el terminal negativo (-), un poco más grande, es hexagonal u octogonal.

TIPOS

- Tipos de Baterías.
- Baterías de plomo-ácido.
- Pila alcalina.
- Baterías de níquel-hierro (Ni-Fe)
- Baterías alcalinas de manganeso.
- Baterías de níquel-cadmio (Ni-Cd)
- Baterías de níquel-hidruro metálico (Ni-MH)
- Baterías de iones de litio (Li-ion)



DIODOS LED

Los diodos son componentes electrónicos que permiten el paso de la corriente en un solo sentido, en sentido contrario no dejan pasar la corriente.

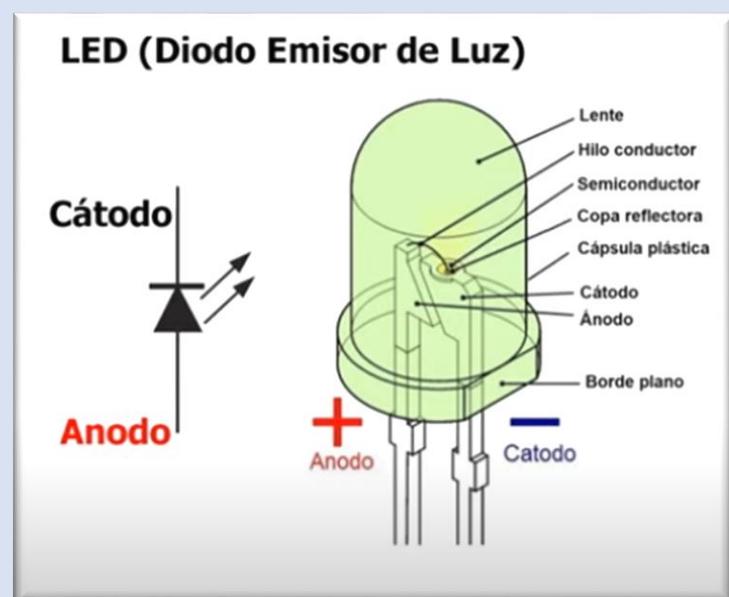
CONEXIÓN



Al conectar varios LEDs en el mismo circuito simplemente se debe respetar la polaridad de los mismos y la conexión debe ser en serie (uno al lado del otro), de manera que el ánodo de cada diodo LED debe conectarse al ánodo del diodo LED que le sigue y así sucesivamente hasta terminar con el número de diodos LED que se conecten

TIPOS

- LED (DIP)
- LED SMD
- LED VANDALICO
- Micro LED.



RESISTENCIAS

Se le denomina resistencia eléctrica a la oposición al flujo de corriente eléctrica a través de un conductor. La unidad de resistencia en el Sistema Internacional es el ohmio, que se representa con la letra griega (Ω), en honor al físico alemán Georg Simon Ohm, quien descubrió el principio que ahora lleva su nombre.



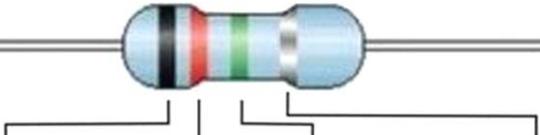
TIPOS

Podemos clasificar las resistencias en tres grandes grupos:

Resistencias fijas: Son las que presentan un valor óhmico que no podemos modificar.

Resistencias variables: Son las que presentan un valor óhmico que nosotros podemos variar modificando la posición de un contacto deslizante.

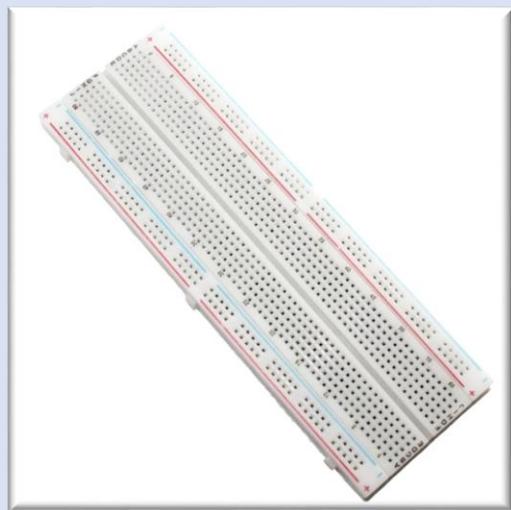
Resistencias especiales: Son las que varían su valor óhmico en función de la estimulación que reciben de un factor externo.

				
Color	1ra. Banda	2da. Banda	3ra. Banda Multiplicador	Tolerancia %
Negro	0	0	x1	
Cafe	1	1	x10	
Rojo	2	2	x100	2%
Naranja	3	3	x1000	
Amarillo	4	4	x10000	
Verde	5	5	x100000	
Azul	6	6	x1000000	
Violeta	7	7	x10000000	
Gris	8	8	x100000000	
Blanco	9	9	x1000000000	
Circuitos Básicos				Dorado 5%
				Plata 10%

Circuitos Básicos				Var. Riesgo
Básico	8	8	x1000000000	Var. Riesgo
Cap	8	8	x1000000000	
Ind	1	1	x100000000	
Volt	0	0	x1000000000	

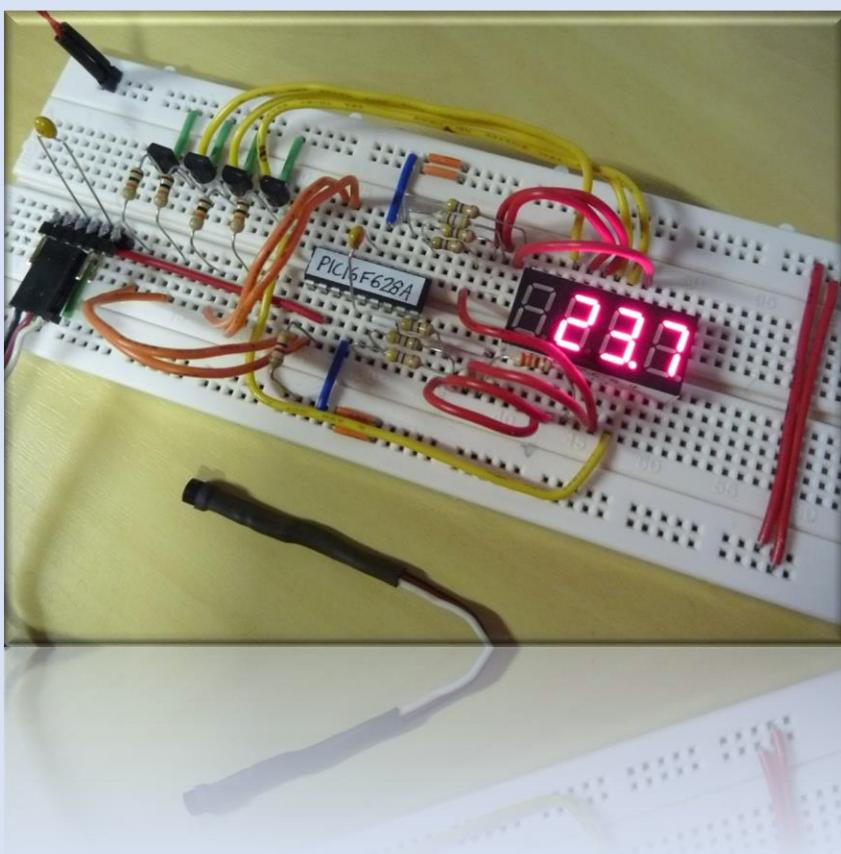
PROTOBOARD

La Protoboard, llamada en inglés breadboard, es una placa de pruebas en los que se pueden insertar elementos electrónicos y cables con los que se arman circuitos sin la necesidad de soldar ninguno de los componentes. Las Protoboard tienen orificios conectados entre sí por medio de pequeñas laminas metálicas. Usualmente, estas placas siguen un arreglo en el que los orificios de una misma fila están conectados entre sí y los orificios en filas diferentes no. Los orificios de las placas normalmente tienen una separación de 2.54 milímetros (0.1 pulgadas).



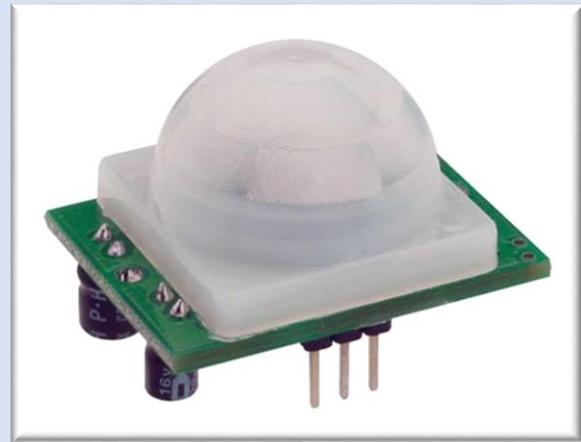
TIPOS

- Protoboard Grande
- Protoboard Mediana
- Protoboard Chica



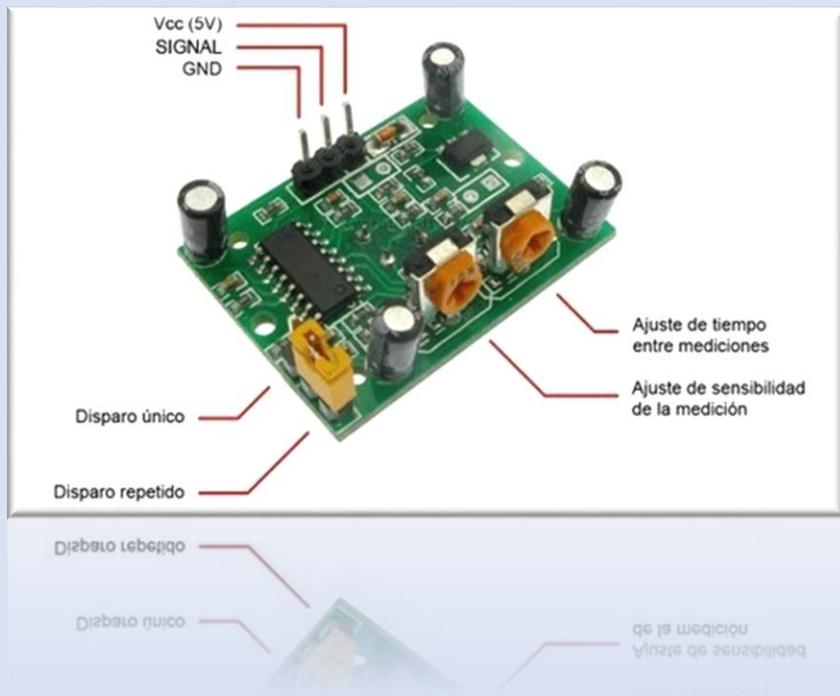
SENSOR PIR

Los detectores pasivos infrarrojos o PIR (Passive Infrared) son dispositivos que detectan variaciones de la radiación infrarroja en el área de cobertura, por lo que son especialmente útiles para detectar la presencia de personas o animales a través del calor que emiten sus cuerpos. El concepto pasivo del nombre de PIR hace referencia a que no generan de forma activa ninguna señal y solo reciben radiaciones para su funcionamiento.



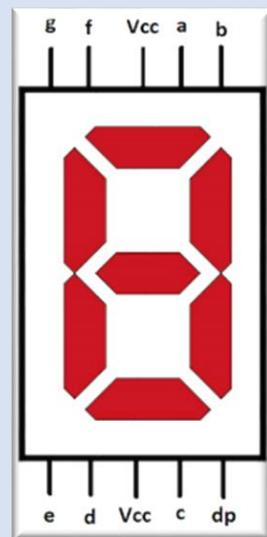
CONEXIÓN

En el circuito conectaremos la salida digital del sensor PIR al pin 11 de nuestro Arduino uno y el pin 12 al LED que está conectado con su respectiva resistencia. Como consecuencia, tendremos nuestro sistema ya montado y listo para programar.



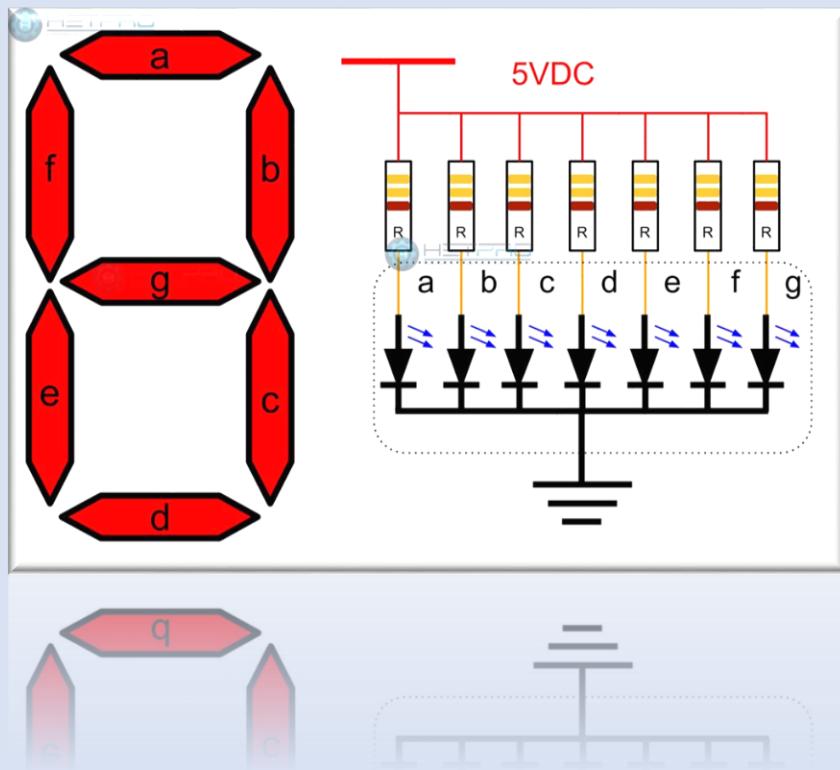
DISPLAY 7 SEGMENTOS

En el circuito conectaremos la salida digital del sensor PIR al pin 11 de nuestro Arduino uno y el pin 12 al LED que está conectado con su respectiva resistencia. Como consecuencia, tendremos nuestro sistema ya montado y listo para programar.



TIPOS

Existen dos tipos diferentes de displays de 7 segmentos: de ánodo común y de cátodo común. En el tipo de ánodo común, todos los ánodos del display están conectados a un pin común, generalmente la fuente de alimentación, y los LED se controlan mediante los cátodos con la conexión a tierra encendida y la potencia pagada.



POTENCIOMETRO

Un potenciómetro se define como un resistor variable de tres terminales cuyo voltaje es ajustable manualmente con la ayuda de un contacto móvil, con el fin de controlar el flujo de corriente eléctrica a través de él.

Todo resistor variable tendrá algún tipo de control mecánico o electrónico para variar su resistencia. El uso más obvio del potenciómetro que la mayoría de nosotros hemos visto es el control del volumen en radios y otros equipos de audio.



CONEXIÓN

Tomamos el potenciómetro, cualquiera de sus “terminales externas” (izquierda o derecha), la conectaremos con la terminal de la resistencia que se encuentra sin conexión alguna.

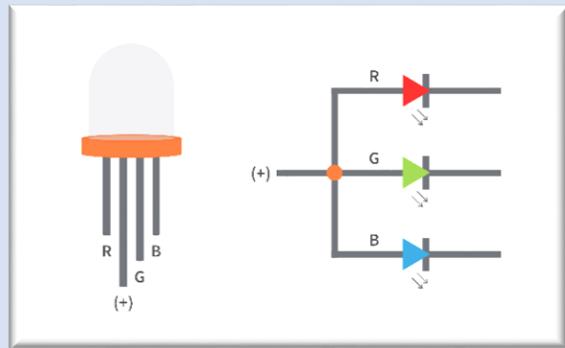
TIPOS

- Potenciómetros lineal
- Potenciómetros logarítmicos
- Potenciómetros digital
- Potenciómetros multivuelta



LED RGB

Un LED (Light Emitting Diode) o Diodo Emisor de Luz, es un componente formado por semiconductores de tipo P y N, que al ser atravesado por una corriente continua emite luz, este fenómeno es conocido como electroluminiscencia.

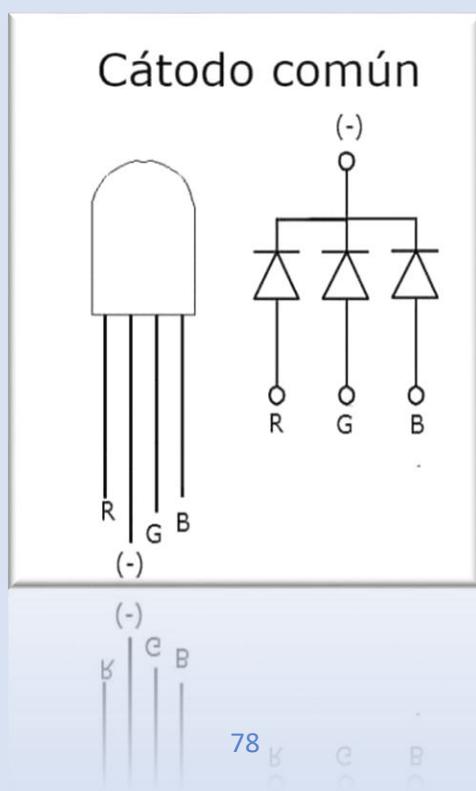


CONEXIÓN

Puesto que tenemos tres pines positivos, vamos a utilizar una resistencia en cada uno de ellos. Esto funciona muy similar a conectar tres leds normales separados. Los cables los he coloreado en consonancia con el color de la luz del LED, de modo que sea más fácil saber en qué pin va cada color.

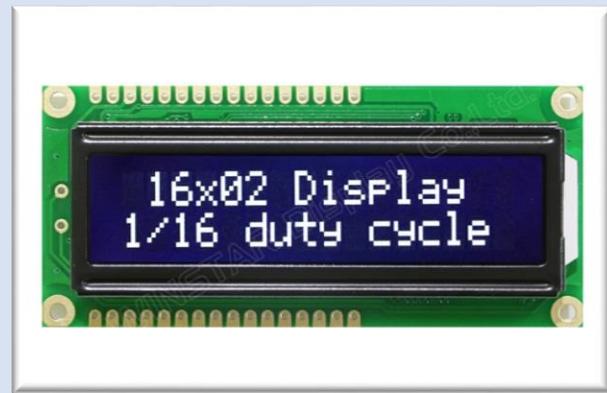
TIPOS

Hay dos tipos de LED RGB: LED de ánodo habitual y LED de cátodo común. En un LED RGB de cátodo común, los tres LED comparten una conexión negativa (cátodo). En un LED RGB de ánodo común, los tres LED comparten una conexión efectiva (ánodo).



PANTALLA LCD

El display LCD 16x2 es ideal para utilizarse como dispositivo de salida e interfaz de usuario en proyectos con Arduino, Raspberry Pi y otros microcontroladores. Es compatible con el juego de comandos estándar del controlador HD44780 en el que se basan la mayoría de los displays de este tipo y permite visualizar hasta 2 líneas de texto de 16 caracteres cada una.



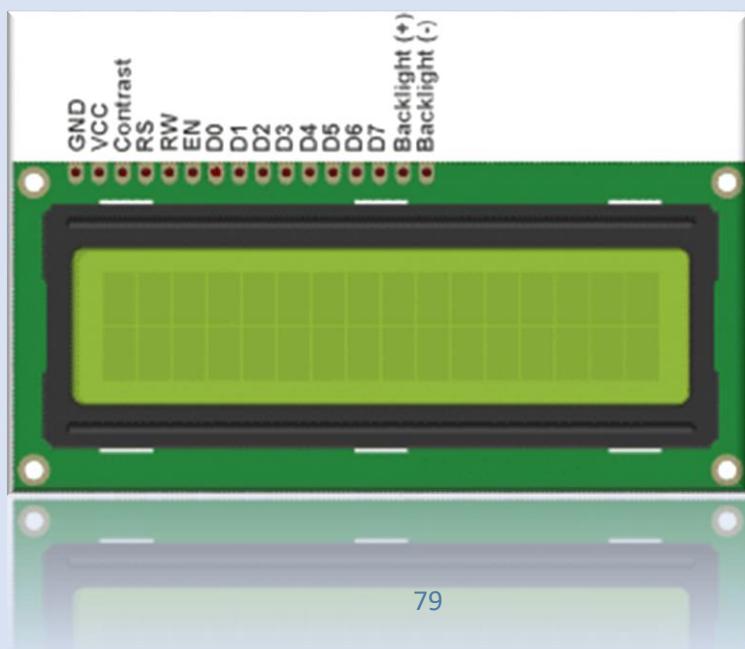
CONEXIÓN

Conectar la alimentación de 5 volts, incluyendo la iluminación LED y colocar un potenciómetro para el ajuste de contraste.

Conectar los 4 pines de datos a la pantalla (modo de 4 bits) y las señales de control RS y EN (R/W va a tierra para ahorrar un pin).

TIPOS

- Display oled
- Display lcd en matriz de puntos
- Conector I2c
- Display lcd de líneas



SENSOR ULTRASONICO

El sensor HC-SR04 es un sensor de distancia de bajo costo, su uso es muy frecuente en la robótica, utiliza transductores de ultrasonido para detectar objetos.



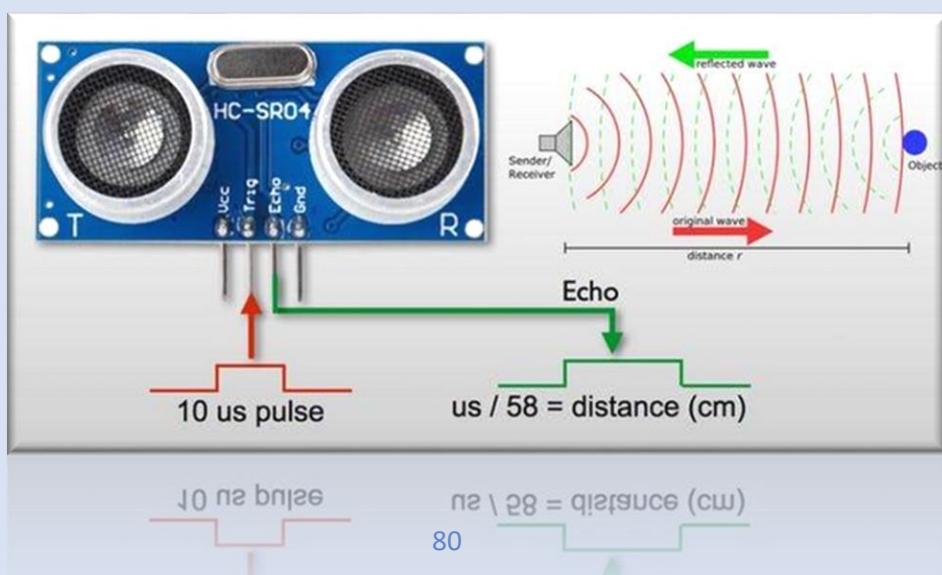
Su funcionamiento consiste en emitir un sonido ultrasónico por uno de sus transductores, y esperar que el sonido rebote de algún objeto presente, el eco es captador por el segundo transductor. La distancia es proporcional al tiempo que demora en llegar el eco.

CONEXIÓN

El sensor de ultrasonidos tiene 4 pines, de los cuales 2 ya han conectado (5V y GND). Los otros 2 pines son ECHO y TRIG que deben ir conectados a 2 pines digitales, siendo estos pines los encargados de enviar la información a la placa Build&Code UNO

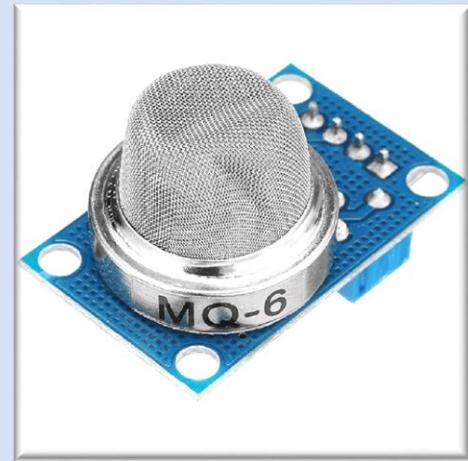
TIPOS

- Sensores fotoeléctricos.
- Sensores de proximidad inductivos.
- Sensores ultrasónicos.
- Sensores de proximidad capacitivos.
- Detección 3D reconfigurable Swift-E.



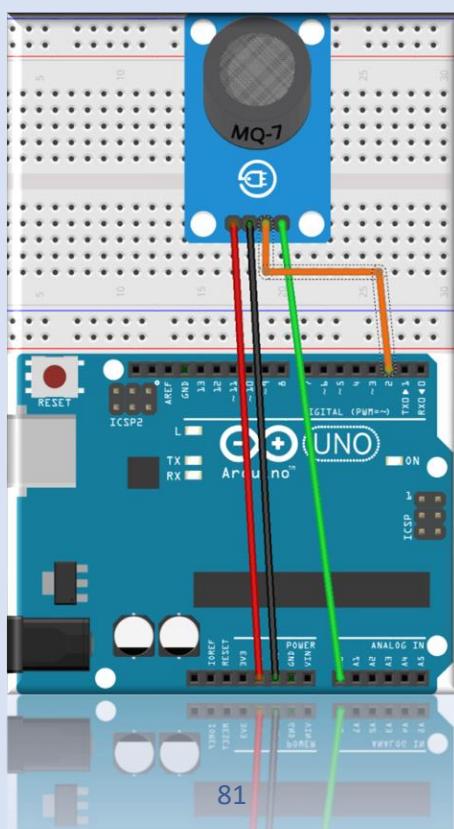
SENSOR DE GAS PROPANO

El sensor de gas MQ2 es analógico y se utiliza en la detección de fugas de gas de equipos en los mercados de consumo y la industria. En este tutorial se utilizará un Arduino UNO. También este sensor es adecuado para la detección de gas LP, i-butano, propano, metano, alcohol, hidrógeno. Incluso tiene una alta sensibilidad y un tiempo de respuesta rápido. Finalmente, la sensibilidad puede ser ajustada por un potenciómetro. Este pequeño sensor de gas detecta la presencia de gas combustible y humo en concentraciones de 300 a 10.000 ppm. Incorpora una sencilla interfaz de tensión analógica que únicamente requiere un pin de entrada analógica del microcontrolador.



CONEXIÓN

Con la conexión de cinco voltios en los pines el sensor se mantiene lo suficientemente caliente para que funcione correctamente. Solo tiene que conectar 5V a cualquiera de los pines (A o B) para que el sensor emita tensión. La sensibilidad del detector se ajusta con una carga resistiva entre los pines de salida y tierra.



BUZZER

Un 'zumbador' (en Inglés 'buzzer') es un transductor electroacústico que produce un sonido o zumbido continuo o intermitente de un mismo tono (generalmente agudo). Sirve como mecanismo de señalización o aviso y se utiliza en múltiples sistemas, como en automóviles o en electrodomésticos, incluidos los despertadores.

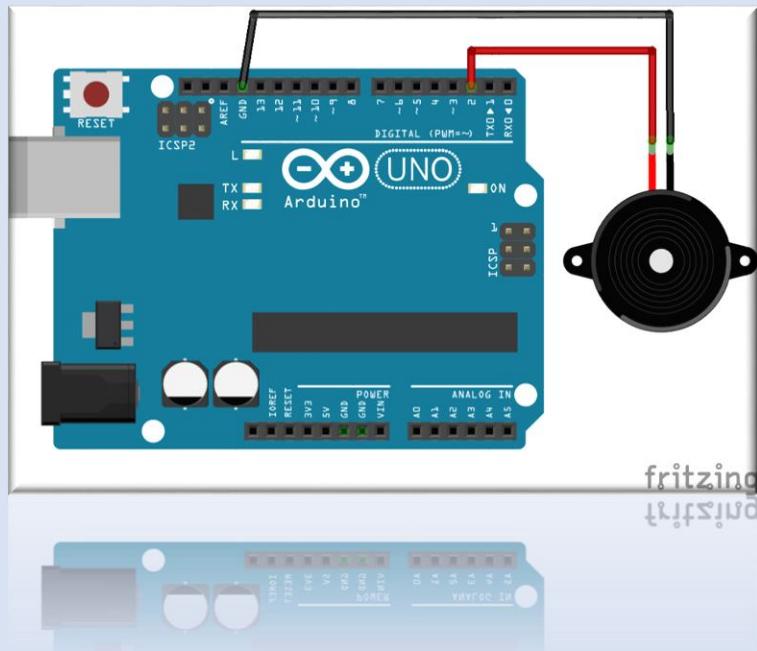


CONEXIÓN

La conexión de un zumbador o buzzer con Arduino no puede ser más simple. Sólo tienes que conectar el positivo a un pin digital PWM y el negativo a GND. En ocasiones, se pone una resistencia entre el pin digital y el zumbador o buzzer.

TIPOS

- Los buzzers activos generan una señal sonora tan pronto como son alimentados por el voltaje adecuado de corriente continua.
- Los buzzers pasivos son como bafles. Requieren de una "señal de sonido" de corriente alterna para sonar.

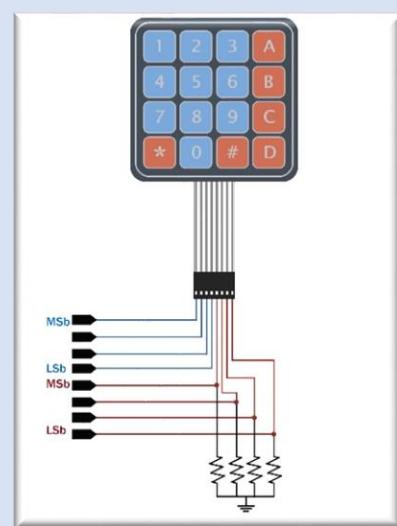


KEYPAD 4x4

El teclado matricial 4x4 está formado por una matriz de pulsadores dispuestos en filas (L1, L2, L3, L4) y columnas (C1, C2, C3, C4), con la intención de reducir el número de pines necesarios para su conexión.

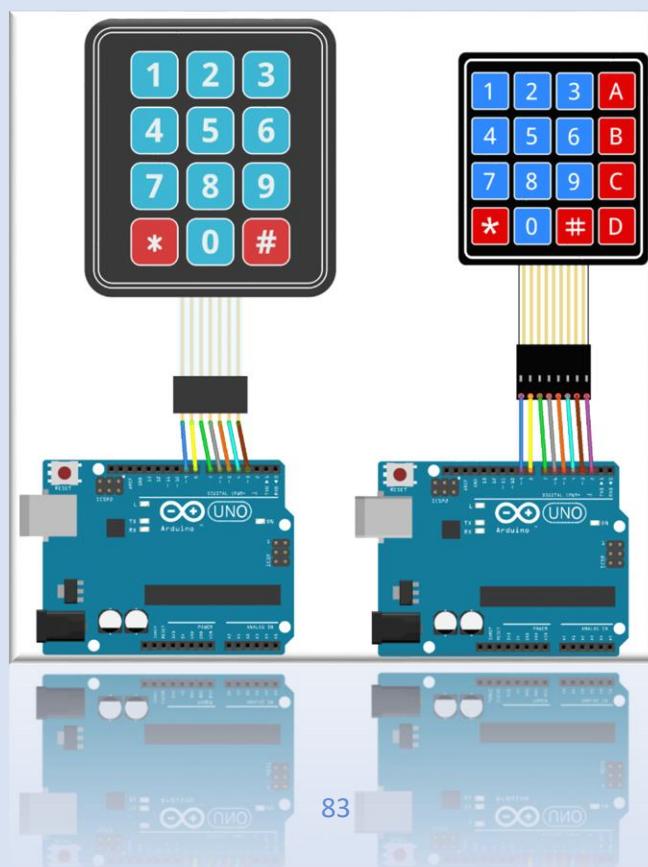
CONEXIÓN

El esquema de conexión es sencillo. Simplemente conectamos todos los pines a entradas digitales de Arduino. Por ejemplo, en el ejemplo de un teclado de 4x4 el esquema quedaría de la siguiente forma.



TIPOS

- Keypad 4x4 pin out.
- Keypad 4x4 tactil.
- Keypad 4x4 membrana.
- Keypad 4x4 botones.



SERVOMOTOR

Un servomotor es un motor eléctrico al que podemos controlar tanto la velocidad, como la posición del eje que gira (también llamada dirección del eje o giro del rotor).

Los servomotores no giran su eje 360º (aunque ahora hay algunos que si lo permiten) como los motores normales, solo giran 180º hacia la izquierda o hacia la derecha (ida y retorno).

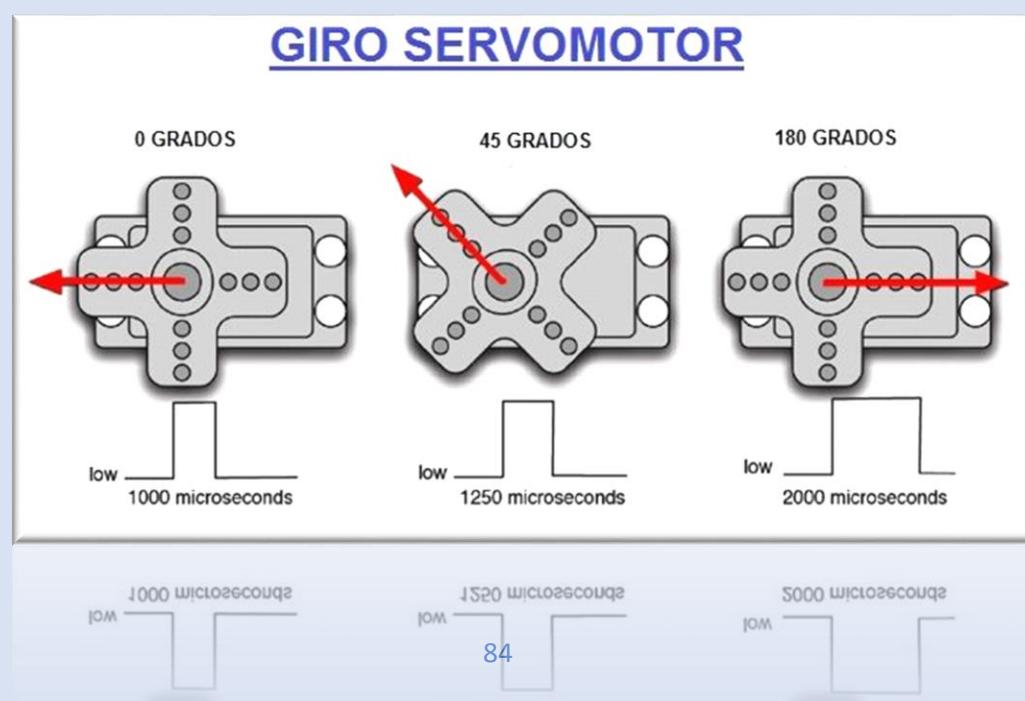


CONEXIÓN

El conector del servomotor se debe conectar directamente a la placa de conexiones, asegurando que el cable marrón del motor se conecte a la fila de pines GND (negro) de la placa de conexiones.

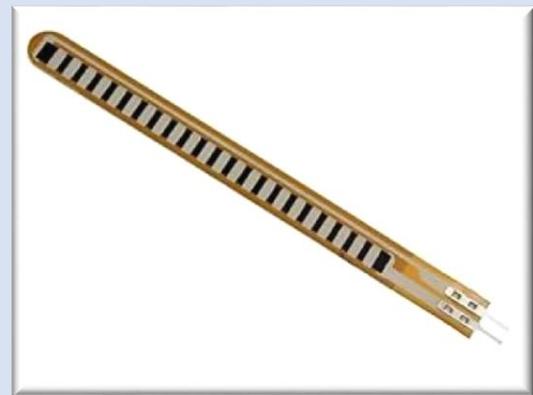
TIPOS

Pueden ser rotativos o lineales. Los servomotores rotativos pueden ser de corriente alterna, de corriente continua o motores paso a paso.



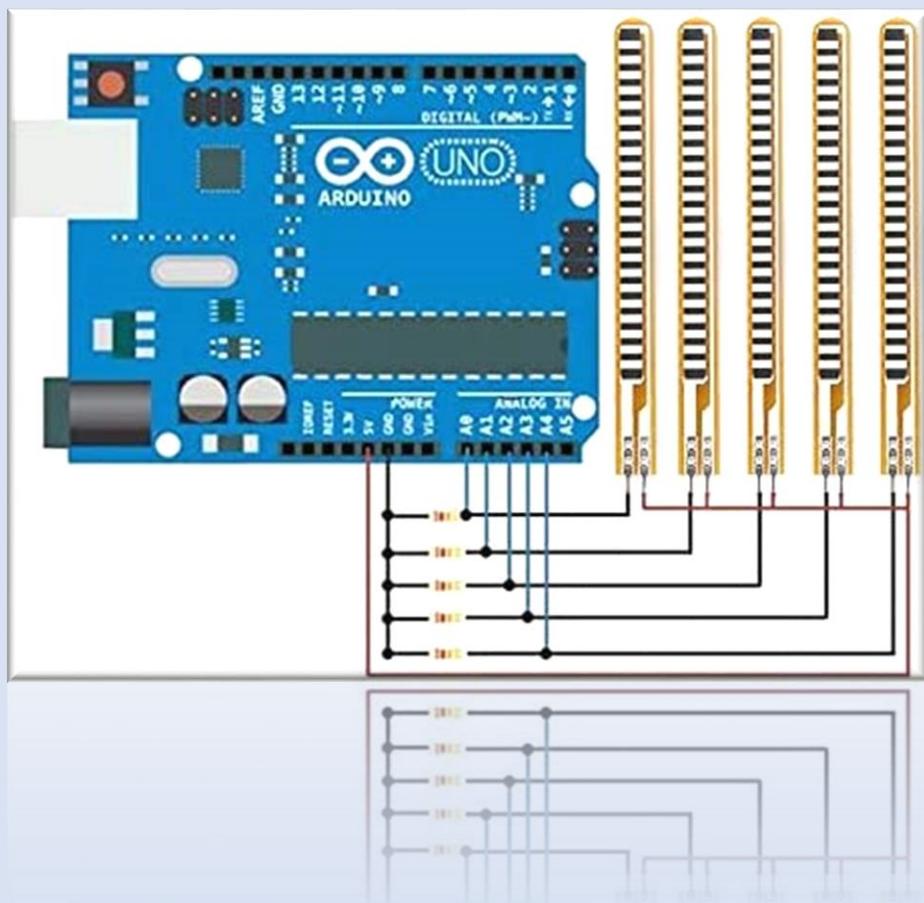
SENSOR FLEX

Los Sensores Flex son resistencias analogicas. Trabajan como divisores de tension analogica variable. Dentro de la flexion del sensor son elementos resistivos de carbono dentro de un sustrato flexible y delgado.



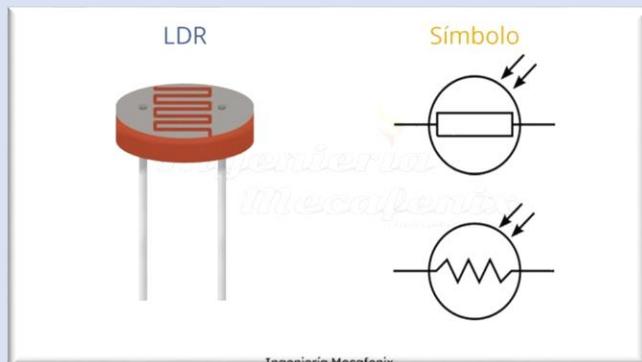
CONEXIÓN

Se puede conectar a la entrada analoga de un microcontrolador (con un resistencia pull-up) o una entrada digital con un condensador de 0.1uF.



FOTORRESISTENCIA o LDR

Una fotoresistencia o LDR por su sigla en inglés Light Dependent Resistor, son una clase de resistencia que va a variar de acuerdo con la luz que esté incidiendo en su superficie. En ese orden de ideas, a medida que la intensidad de la luz que incide en la superficie de la fotoresistencia, va a ser menor la resistencia, pero en cuanto menor sea la luz que incide en el material, mayor será la resistencia.

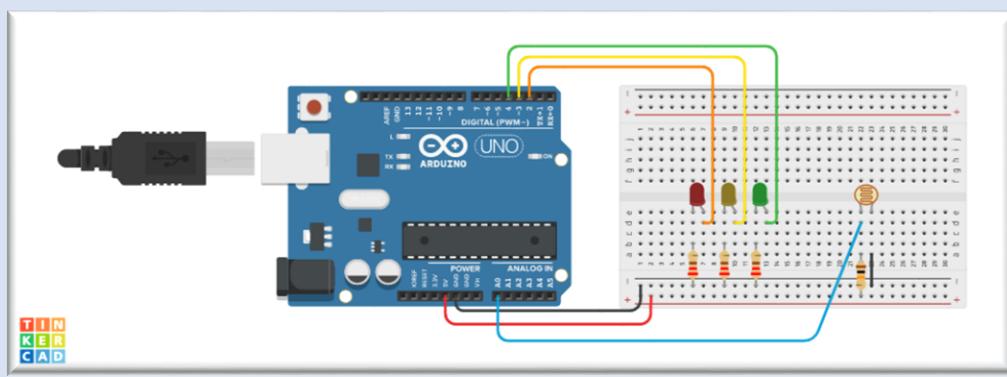


CONEXIÓN

- Si el divisor de tensión se alimenta con 5V el voltaje de la señal podrá tomar valores entre 0V y 5V, de igual manera es importante señalar que el comportamiento del sensor, es decir, si el voltaje aumenta al aumentar la luz y viceversa o si el voltaje disminuye al aumentar la luz y viceversa dependerá de como alimentemos el divisor de tensiones:
 - Si conectamos GND al pin de la resistencia de valor fijo y 5V al pin de la fotoresistencia, el voltaje aumentará al aumentar la intensidad del a luz y disminuirá en ausencia de luz.
 - Si conectamos 5V al pin de la resistencia de valor fijo y GND al pin de la fotoresistencia, el voltaje disminuirá al aumentar la intensidad de la luz y aumentará en ausencia de luz.

TIPOS

- Fotorresistencia o LDR lineales: Son aquellas que polarizan a la inversa de la fuente donde se conecte. Son tambien llamadas fotodiodos.
 - Fotorresistencia o LDR no lineales: La polaridad de este fotorresistor no depende de la fuente donde se conecte.



VOLTIMETRO

El voltímetro es un instrumento de medición de la diferencia de potencial o tensión entre dos puntos de un circuito eléctrico. Para tal fin ha de colocarse en paralelo, es decir, en derivación sobre los puntos entre los que se realiza la medida. Este aparato cumple la función de determinar el voltaje.



TIPOS

- Existen varios tipos de voltímetros:
- Voltímetro analógico: la tensión medida se debe leer manualmente.
- Voltímetro digital: el voltímetro proporciona directamente el valor de la tensión en una pantalla.