



*ugr*

Universidad  
de **Granada**

**PDIH**

**Práctica 3**

**Autor**

**Pablo Fernández Gallardo**



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

Granada, 07/05/2024

<b>Ejercicio 1</b>	<b>3</b>
Materiales y componentes utilizados:	3
Imagen del circuito:	3
Código fuente del programa:	4
<b>Ejercicio 2</b>	<b>4</b>
Materiales y componentes utilizados:	4
Imagen del circuito:	5
Código fuente del programa:	6
<b>Ejercicio Opcional 1</b>	<b>7</b>
Materiales y componentes utilizados:	7
Imagen del circuito:	7
Código fuente del programa:	8
<b>Ejercicio Opcional 2</b>	<b>9</b>
Materiales y componentes utilizados:	9
Imagen del circuito:	10
Código fuente del programa:	11

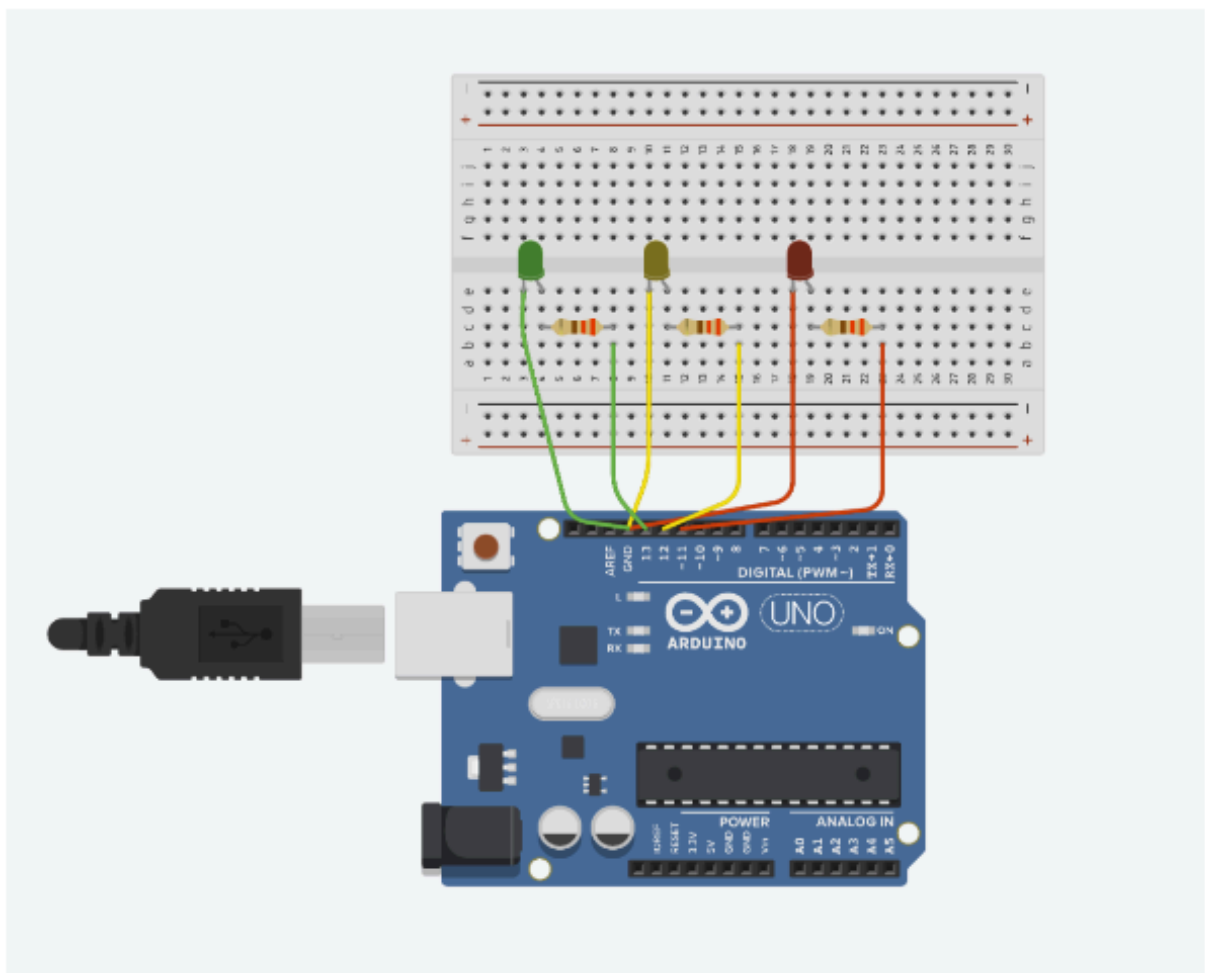
## Ejercicio 1

### Materiales y componentes utilizados:

- Placa Arduino
- 3 Resistencias de 220 ohmios
- 3 leds: rojo, amarillo y verde.

He unido los 3 leds con una de las patas a tierra de la placa arduino y a la otra una resistencia, siendo en total 3 utilizadas, para que el led no se funda. También he unido las resistencias a los pines verde -> 13; amarillo -> 12; rojo -> 11.

### Imagen del circuito:



### Código fuente del programa:

```
C/C++
void setup()
{
    pinMode(13, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(11, OUTPUT);
}

void loop()
{
    digitalWrite(13, HIGH);
    digitalWrite(12, LOW);
    digitalWrite(11, LOW);
    delay(1500); // Wait for 1500 millisecond(s)
    digitalWrite(13, LOW);
    digitalWrite(12, HIGH);
    digitalWrite(11, LOW);
    delay(1500); // Wait for 1500 millisecond(s)
    digitalWrite(13, LOW);
    digitalWrite(12, LOW);
    digitalWrite(11, HIGH);
    delay(1500); // Wait for 1500 millisecond(s)
}
```

Se inicializan los pines como salidas y se aplican esas salidas para encender el led correspondiente a su pin. También hay un delay para que funcione correctamente.

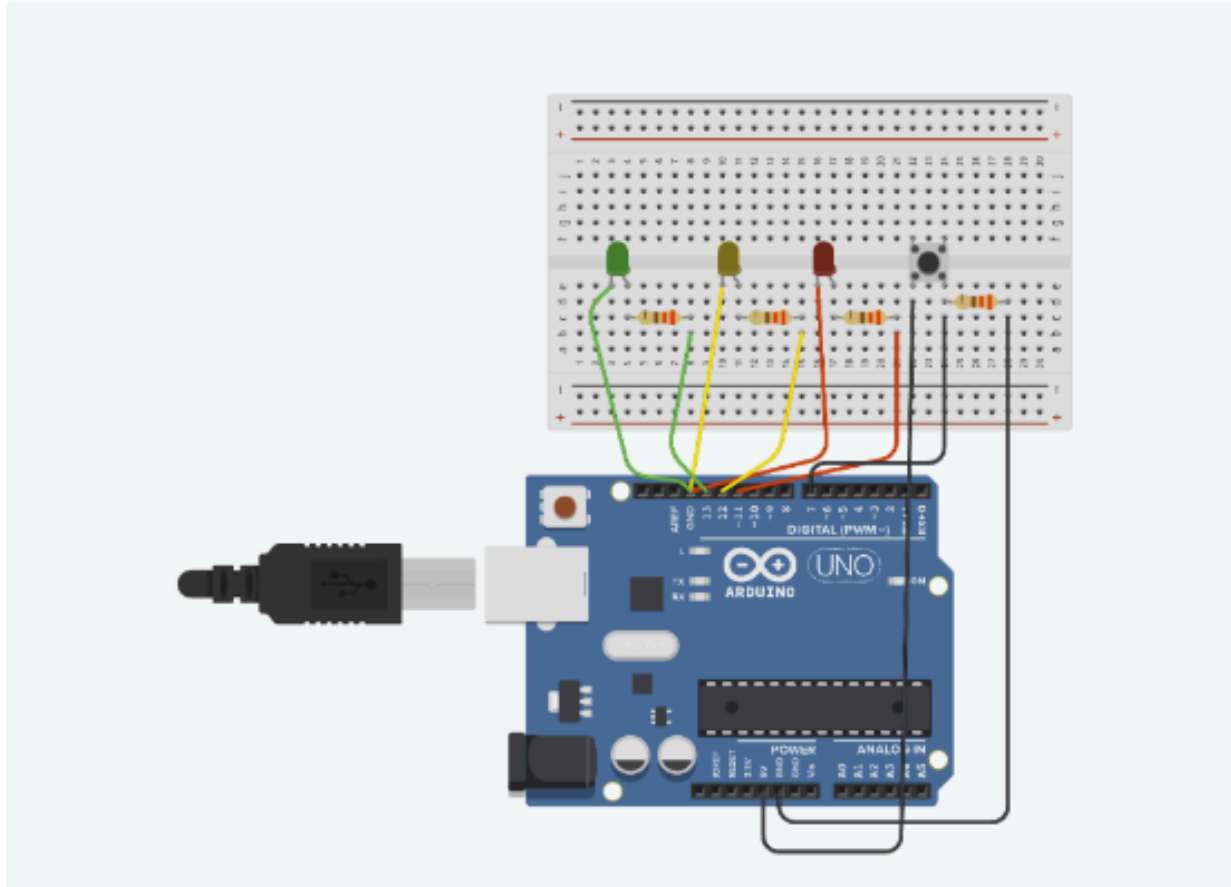
### **Ejercicio 2**

#### Materiales y componentes utilizados:

- 4 Resistencias de 220 ohmios
- 3 LEDs: rojo, amarillo y verde.

- 1 placa Arduino
- 1 interruptor

Imagen del circuito:



He añadido el pulsador conectándolo a la entrada digital 7 y a la corriente de 5V y también a tierra.

### Código fuente del programa:

```
C/C++
int state;
void setup()
{
    pinMode(7, INPUT);
    pinMode(13, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(11, OUTPUT);
}

void loop()
{
    state = digitalRead(7);

    if (state == HIGH) {
        digitalWrite(13, LOW);
        digitalWrite(12, LOW);
        digitalWrite(11, HIGH);
    } else {
        delay(1500); // Wait for 1500 millisecond(s)
        digitalWrite(13, HIGH);
        digitalWrite(12, LOW);
        digitalWrite(11, LOW);
        delay(1500); // Wait for 1500 millisecond(s)
        digitalWrite(13, LOW);
        digitalWrite(12, HIGH);
        digitalWrite(11, LOW);
    }
}
```

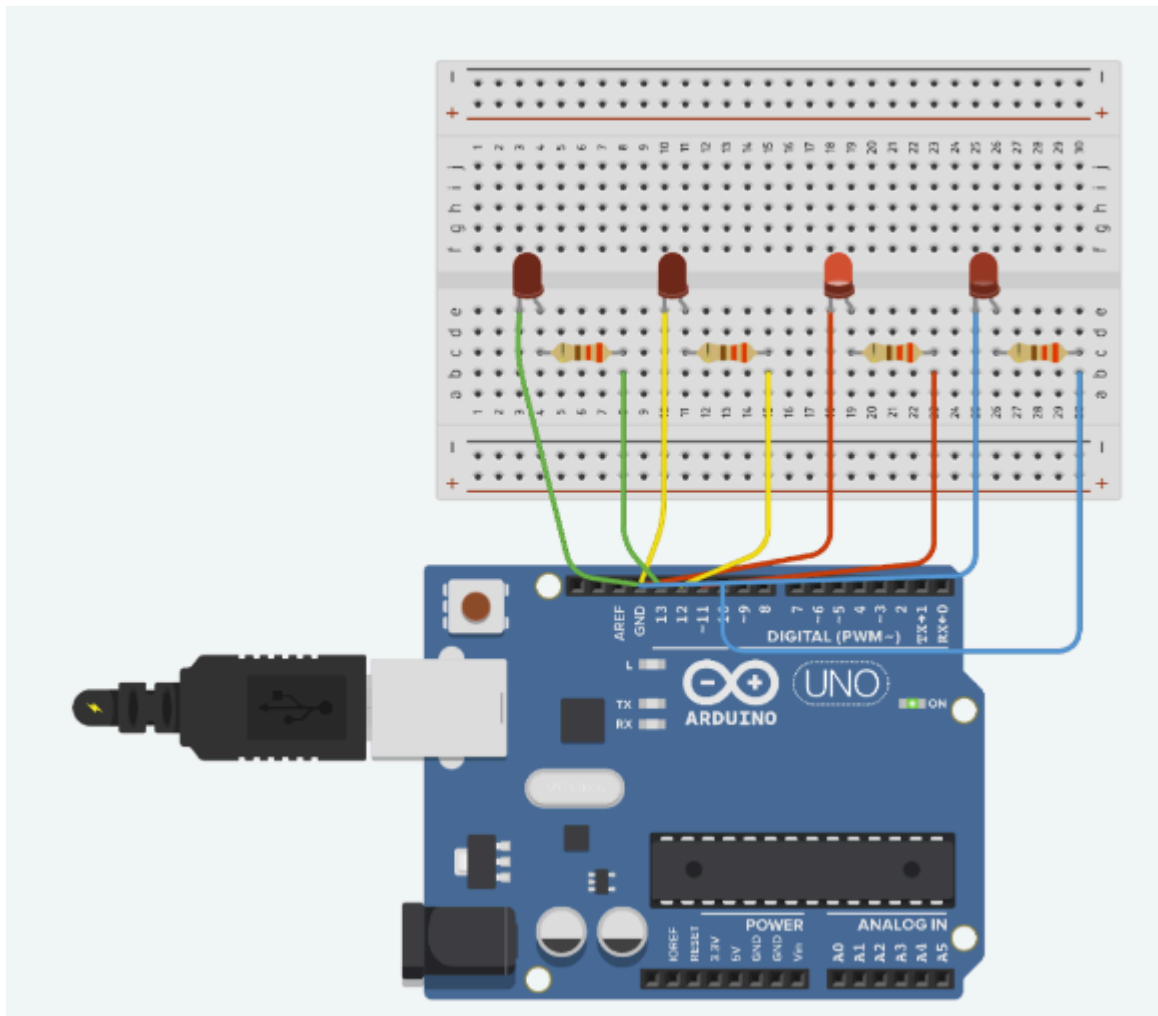
Inicializo la entrada 7 con un input, es decir, la del interruptor y en cada iteración del bucle compruebo el estado del interruptor para que funcione correctamente.

## Ejercicio Opcional 1

### Materiales y componentes utilizados:

- 4 Resistencias de 220 ohmios
- 4 LEDs: rojos
- 1 placa Arduino

### Imagen del circuito:



Inicializo como output el pin número 10 donde está el nuevo LED y realizo primero un recorrido normal cada 100ms, encendiendo el siguiente LED en la secuencia. A continuación, le digo que haga el mismo recorrido pero a la inversa añadiendo dos estados de los LEDs, ya que no quiero que cada LED tanto final como inicial repita su estado de encendido.

### Código fuente del programa:

```
C/C++
void setup()
{
    pinMode(13, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(11, OUTPUT);
    pinMode(10, OUTPUT);
}

void loop()
{
    digitalWrite(13, HIGH);
    digitalWrite(12, LOW);
    digitalWrite(11, LOW);
    digitalWrite(10, LOW);
    delay(150);
    digitalWrite(13, LOW);
    digitalWrite(12, HIGH);
    digitalWrite(11, LOW);
    digitalWrite(10, LOW);
    delay(150);
    digitalWrite(13, LOW);
    digitalWrite(12, LOW);
    digitalWrite(11, HIGH);
    digitalWrite(10, LOW);
    delay(150);
    digitalWrite(13, LOW);
    digitalWrite(12, LOW);
    digitalWrite(11, LOW);
    digitalWrite(10, HIGH);
    delay(150); // Recorrido inverso
    digitalWrite(13, LOW);
    digitalWrite(12, LOW);
    digitalWrite(11, HIGH);
    digitalWrite(10, LOW);
}
```



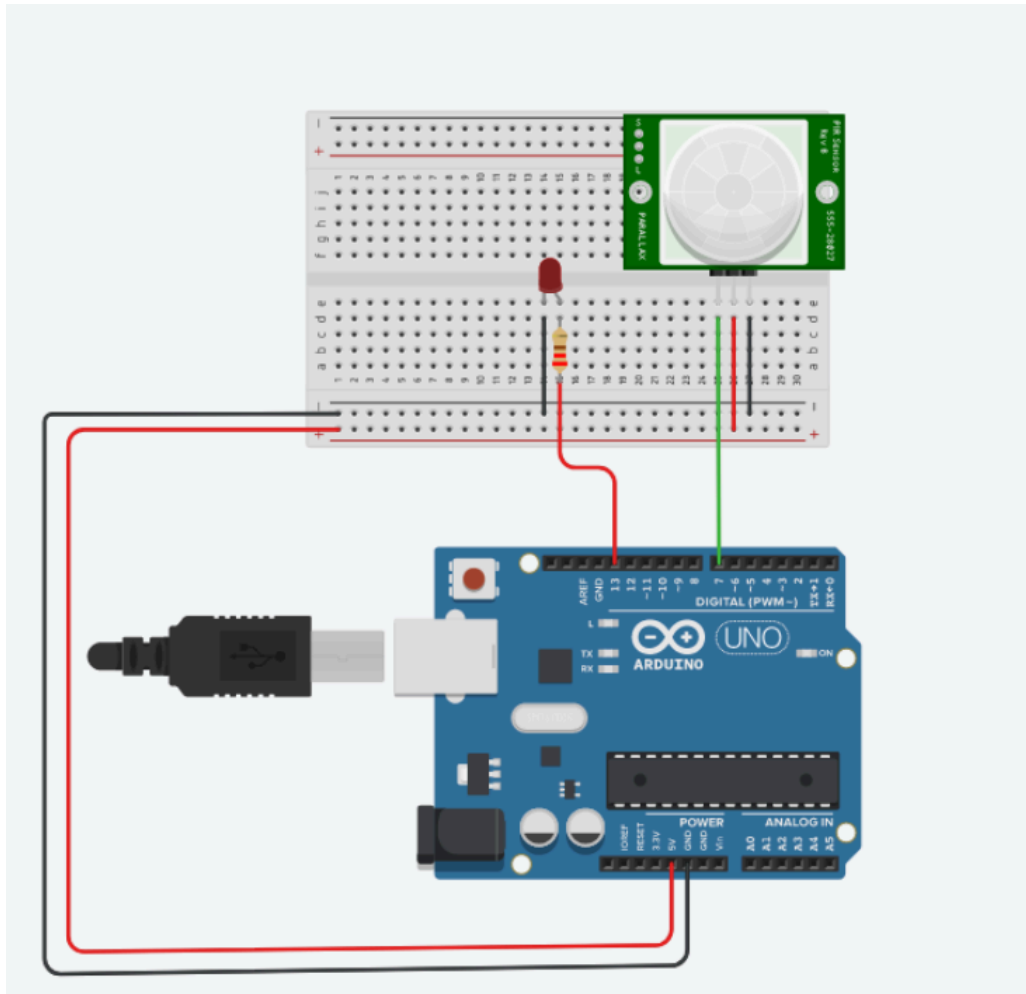
```
delay(150);  
digitalWrite(13, LOW);  
digitalWrite(12, HIGH);  
digitalWrite(11, LOW);  
digitalWrite(10, LOW);  
delay(150);  
}
```

## Ejercicio Opcional 2

### Materiales y componentes utilizados:

- 1 PIR de detección de presencia
- 1 resistencia de 220 ohmios
- 1 Placa Arduino
- 1 Placa de pruebas pequeña

Imagen del circuito:



### Código fuente del programa:

```
C/C++  
  
int ROJO = 13;  
int DETECTOR = 7;  
  
void setup()  
{  
    pinMode(ROJO, OUTPUT);  
    pinMode(DETECTOR, INPUT);  
}  
  
void loop()  
{  
    int estado = digitalRead(DETECTOR);  
    digitalWrite(ROJO, estado);  
}
```