

# Práctica 3. Experimentación con Arduino

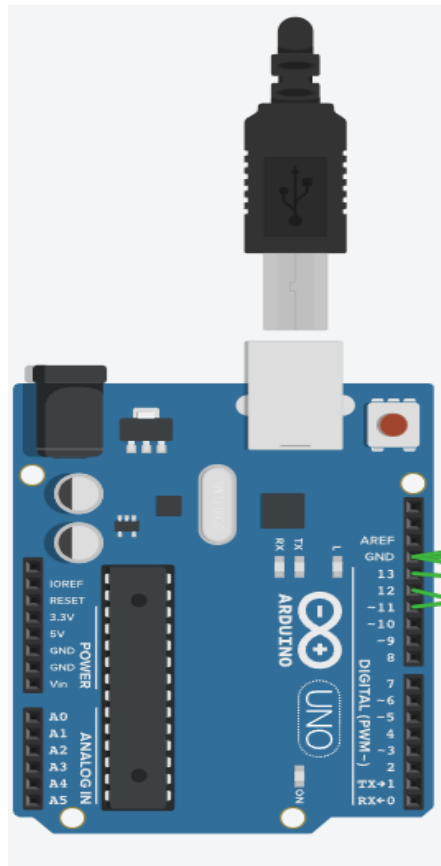
## Realizado por Rubén Martín Jáimez

Requisitos mínimos:

- Parpadeo tres LEDs: Para este ejercicio, simplemente amplié el ejercicio requerido en el seminario con un LED conectado a la salida 11 conectado a su resistencia.

Componentes usados:

Arduino Uno:



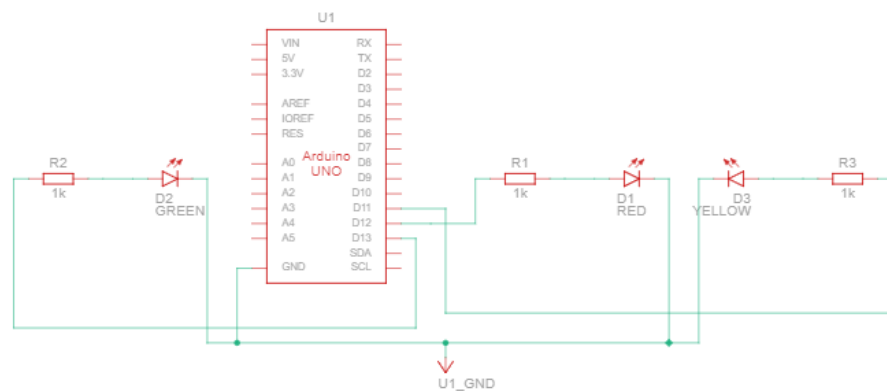
LEDs:



Resistencias:



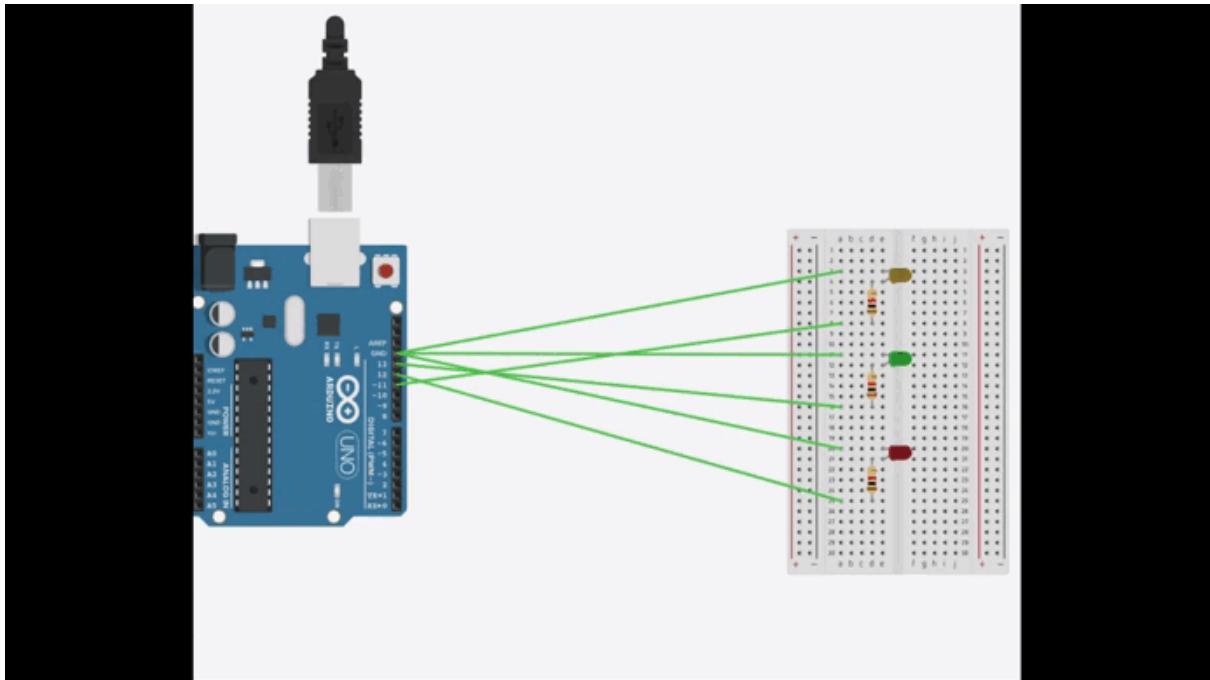
Esquema Fritzing:



Código fuente:

```
1 // C++ code
2 //
3
4 void setup()
5 {
6   pinMode(13, OUTPUT); //Configuración del pin 13 como salida
7   pinMode(12, OUTPUT); //Configuración del pin 12 como salida
8   pinMode(11, OUTPUT); //Configuración del pin 11 como salida
9 }
10
11 void loop()
12 {
13   digitalWrite(11,HIGH); // Enciende el led de la salida 11
14   delay(1500); // Espera 1500ms
15   digitalWrite(11,LOW); // Apaga el led de la salida 11
16   digitalWrite(13,HIGH); // Enciende el led de la salida 13
17   delay(1500); // Espera 1500ms
18   digitalWrite(13,LOW); // Apaga el led de la salida 13
19   digitalWrite(12,HIGH); // Enciende el led de la salida 12
20   delay(1500); // Espera 1500ms
21   digitalWrite(12,LOW); // Apaga el led de la salida 12
22 }
```

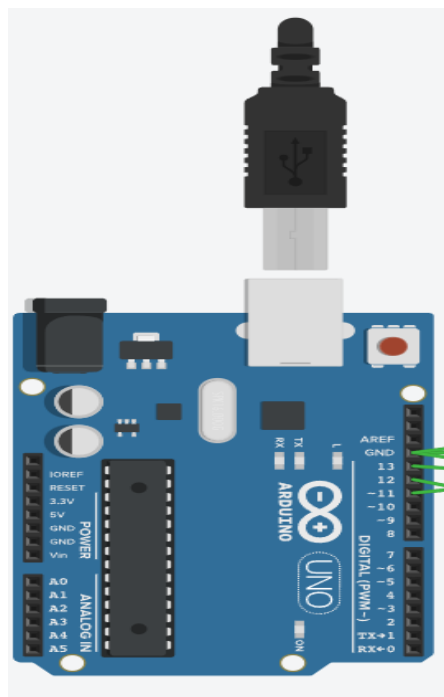
Video de comprobación:



- Pulsador LED rojo: En este ejercicio, he partido del ejercicio anterior y he añadido un botón a la entrada 7, conectado a la entrada de 5V y con su correspondiente resistencia. Para conseguir que el LED rojo solo se encienda si se pulsa el botón y los otros LEDs se apaguen, he añadido un if else, en el cuál si pulsas el botón se enciende el LED rojo y los restantes se apagan, sino es pulsado, los otros dos LEDs seguirán con el parpadeo, es decir, como en el ejercicio anterior.

Componentes usados:

Arduino UNO:



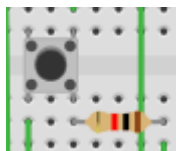
LEDs:



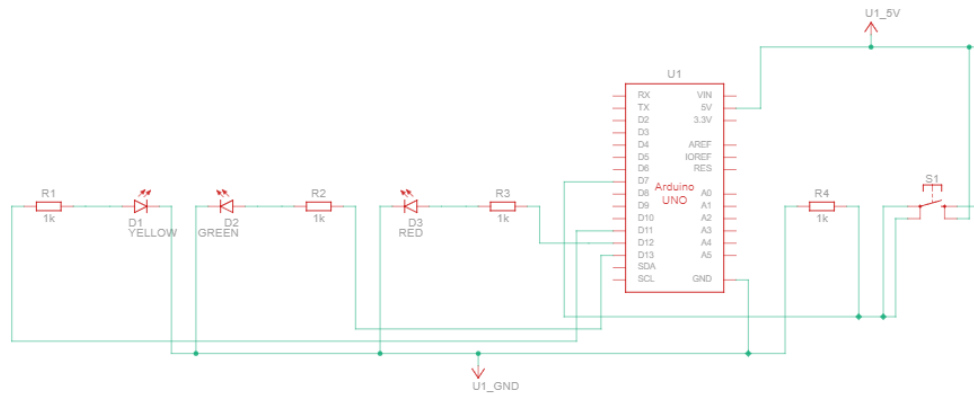
Resistencias:



Botón y su resistencia:



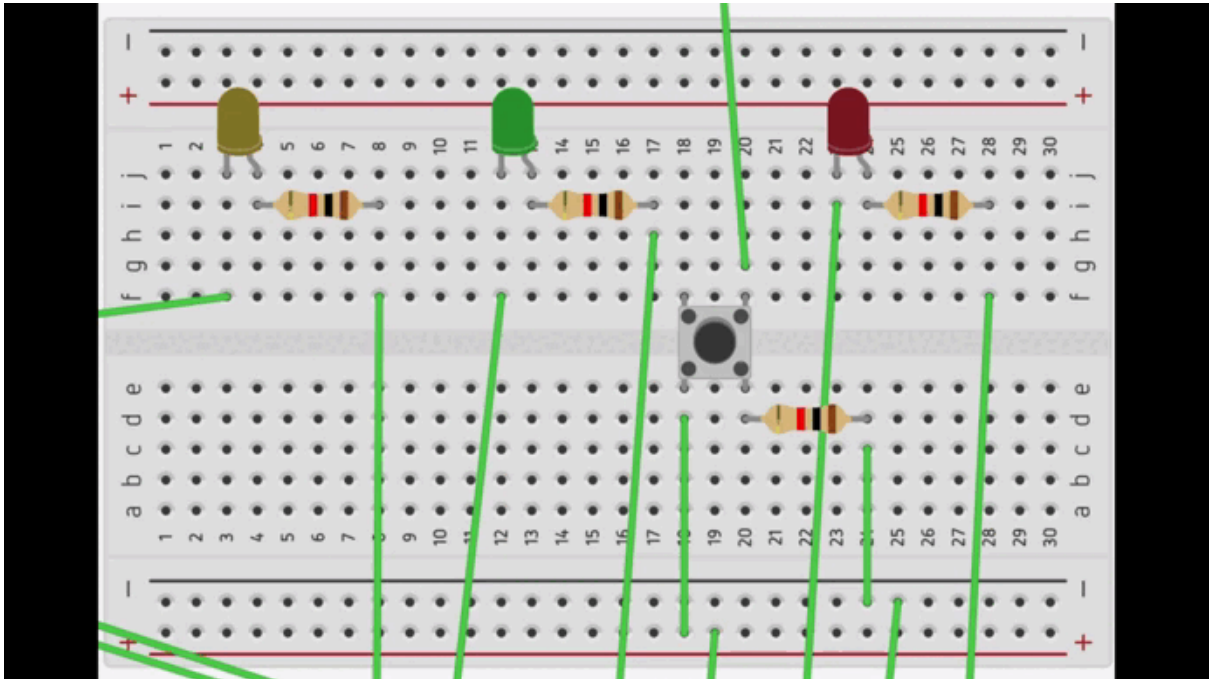
## Esquema Fritzing:



## Código Fuente:

```
1 // C++ code
2 //
3
4 int buttonState = 0; // Variable para ver el estado del botón, es decir, pulsado o no
5
6 void setup()
7 {
8   pinMode(13, OUTPUT); //Configuración del pin 13 como salida (LED verde)
9   pinMode(12, OUTPUT); //Configuración del pin 12 como salida (LED rojo)
10  pinMode(11, OUTPUT); //Configuración del pin 11 como salida (LED amarillo)
11  pinMode(7, INPUT);    //Configuración del pin 7 como entrada (botón)
12 }
13
14 void loop()
15 {
16   buttonState = digitalRead(7); //Lee el estado del pin 7, es decir, del botón
17
18   if (buttonState == HIGH){ //Si el botón está pulsado
19     digitalWrite(12,HIGH); //Enciende el LED de la salida 12
20     digitalWrite(13,LOW);  //Apaga el LED de la salida 13
21     digitalWrite(11,LOW);  //Apaga el LED de la salida 11
22   } else {
23     digitalWrite(12,LOW); // Apaga el LED de la salida 12
24     digitalWrite(11,HIGH); // Enciende el LED de la salida 11
25     delay(1500); // Espera 1500ms
26     digitalWrite(11,LOW); // Apaga el LED de la salida 11
27     digitalWrite(13,HIGH); // Enciende el LED de la salida 13
28     delay(1500); // Espera 1500ms
29     digitalWrite(13,LOW); // Apaga el LED de la salida 13
30   }
31 }
32 }
```

Vídeo de comprobación:

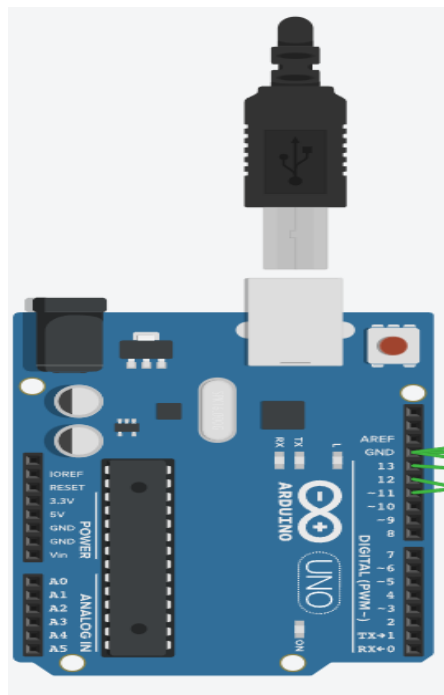


#### - Opcionales

He elegido el ejercicio de la detección de la cantidad de luz a partir del uso de un LED y de un fotosensor. Para ello, he declarado una variable que almacenará el valor de luz recogida, la cuál actuará como umbral, ya que a partir de un valor determinado (en este caso, 500) se iluminará el LED. Si no recoge más de 500, el LED se mantendrá apagado.

Componentes usados:

Arduino UNO:



LED:



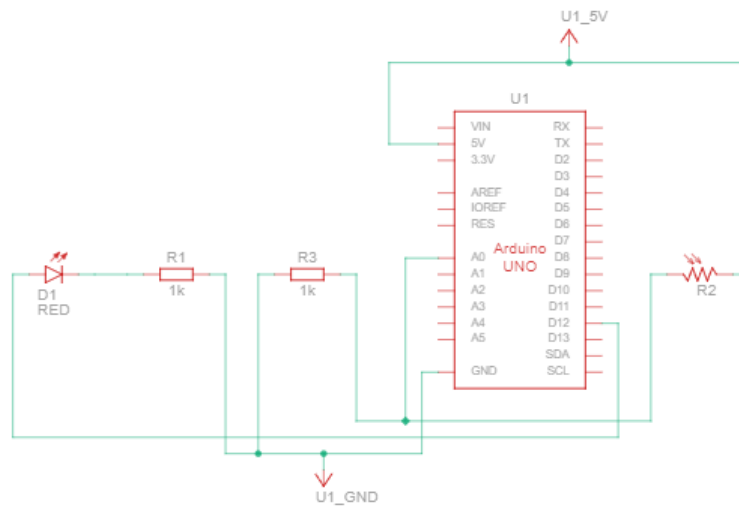
Resistencias:



Fotosensor:



Esquema Fritzing:



Código fuente:

```
1
2 void setup() {
3   pinMode(12, OUTPUT); //Configuración del pin 12 como salida (LED)
4   pinMode(A0, INPUT); // Configuración del pin analógico A0 como entrada (fotosensor)
5 }
6
7 void loop() {
8
9   int luminosidad = analogRead(A0); // Lee la señal analógica del fotosensor, es decir, la luz
10
11   if (luminosidad > 500){ // Si la luz captada es mayor que 500, entonces se enciende el LED
12     // sino, el LED permanece apagado
13     digitalWrite(12, HIGH);
14   }else {
15     digitalWrite(12, LOW);
16   }
17 }
18 }
```

Vídeo de comprobación:

