

Seminario Arduino y Práctica 3

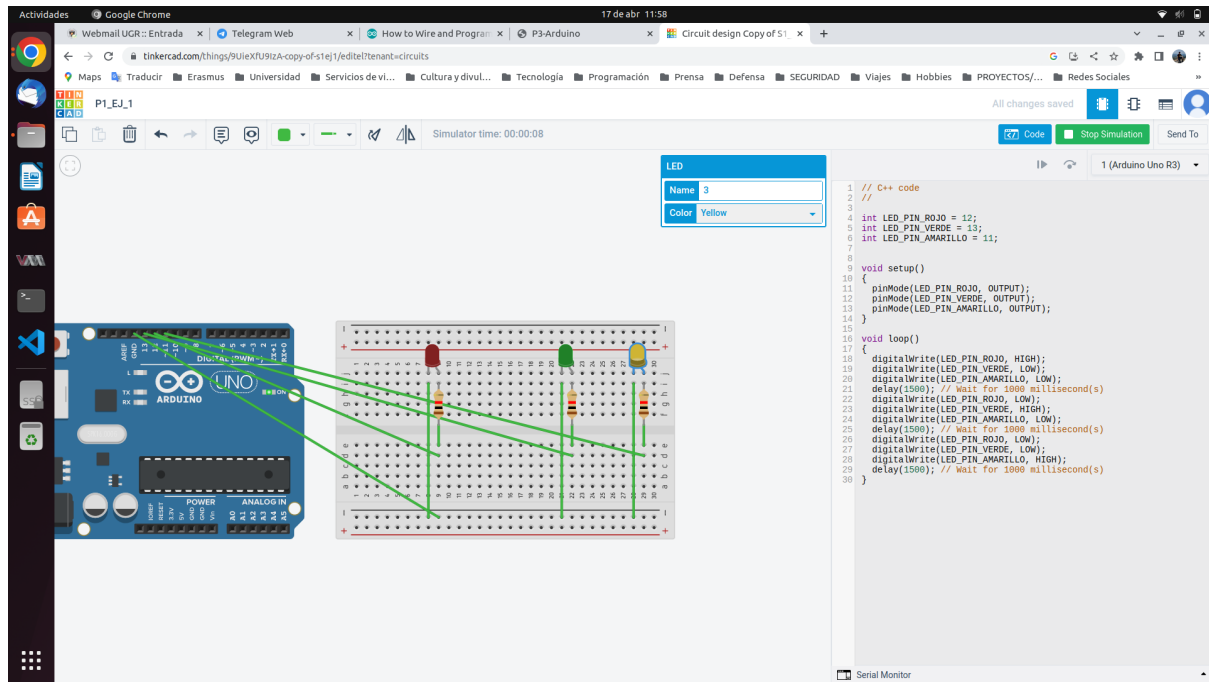
Ángel López Martos

1. Implementar el programa de parpadeo de LED, ampliándolo para que encienda y apague alternativamente tres LEDs (uno rojo, otro amarillo y otro verde), conectados a las salidas digitales 11, 12 y 13 del Arduino, a un intervalo de 1.5 segundos. Crear el esquema con Fritzing y cargar el programa en Arduino para comprobar que funciona correctamente.

```
int LED_PIN_ROJO = 12; //definimos los pines de control como globales
int LED_PIN_VERDE = 13;
int LED_PIN_AMARILLO = 11;

void setup()
{
  pinMode(LED_PIN_ROJO, OUTPUT); //Indicamos que los pines de los leds son de
  salida
  pinMode(LED_PIN_VERDE, OUTPUT);
  pinMode(LED_PIN_AMARILLO, OUTPUT);
}

void loop()
{
  digitalWrite(LED_PIN_ROJO, HIGH); //Encendemos
  digitalWrite(LED_PIN_VERDE, LOW); //Apagamos
  digitalWrite(LED_PIN_AMARILLO, LOW); //Apagamos
  delay(1500); //Esperamos
  digitalWrite(LED_PIN_ROJO, LOW); //Repetimos proceso
  digitalWrite(LED_PIN_VERDE, HIGH);
  digitalWrite(LED_PIN_AMARILLO, LOW);
  delay(1500);
  digitalWrite(LED_PIN_ROJO, LOW);
  digitalWrite(LED_PIN_VERDE, LOW);
  digitalWrite(LED_PIN_AMARILLO, HIGH);
  delay(1500);
}
```



2. Partir del programa de parpadeo de LEDs anterior y ampliarlo con las modificaciones necesarias para que se encienda el LED rojo solo cuando se pulse un interruptor conectado a la entrada digital 7, y en ese momento se apaguen los LEDs amarillo y verde.

```
const int PIN_BOTON = 7; //definimos los pines de los leds y del boton como globales
const int PIN_LED = 13;
const int PIN_LED_VERDE = 12;
const int PIN_LED_AMARILLO = 11;
```

```
int ESTADO_BOTON = 0; //Establecemos una variable global para comparar el estado del boton
```

```
void setup() {
  pinMode(PIN_LED, OUTPUT); //Indicamos que pines son de entrada y salida
  pinMode(PIN_LED_VERDE, OUTPUT);
  pinMode(PIN_LED_AMARILLO, OUTPUT);
  pinMode(PIN_BOTON, INPUT);
}
```

```
void loop() {
```

```
  ESTADO_BOTON = digitalRead(PIN_BOTON); //leemos por el pin del boton y actualizamos su estado si hay o no hay pulsacion
```

```
  if(ESTADO_BOTON == LOW){ //Si no hay pulsación, repetimos el programa del ejercicio anterior
```

```

digitalWrite(PIN_LED, LOW);
digitalWrite(PIN_LED_VERDE, HIGH);
    digitalWrite(PIN_LED_AMARILLO, LOW);
delay(150);
digitalWrite(PIN_LED, LOW);
digitalWrite(PIN_LED_VERDE, LOW);
    digitalWrite(PIN_LED_AMARILLO, HIGH);
    delay(150);
digitalWrite(PIN_LED, HIGH);
digitalWrite(PIN_LED_VERDE, LOW);
    digitalWrite(PIN_LED_AMARILLO, LOW);
delay(150);
}

```

else if (ESTADO_BOTON == HIGH) { //Si hay pulsación, apagamos los leds verde y amarillo y encendemos rojo

```

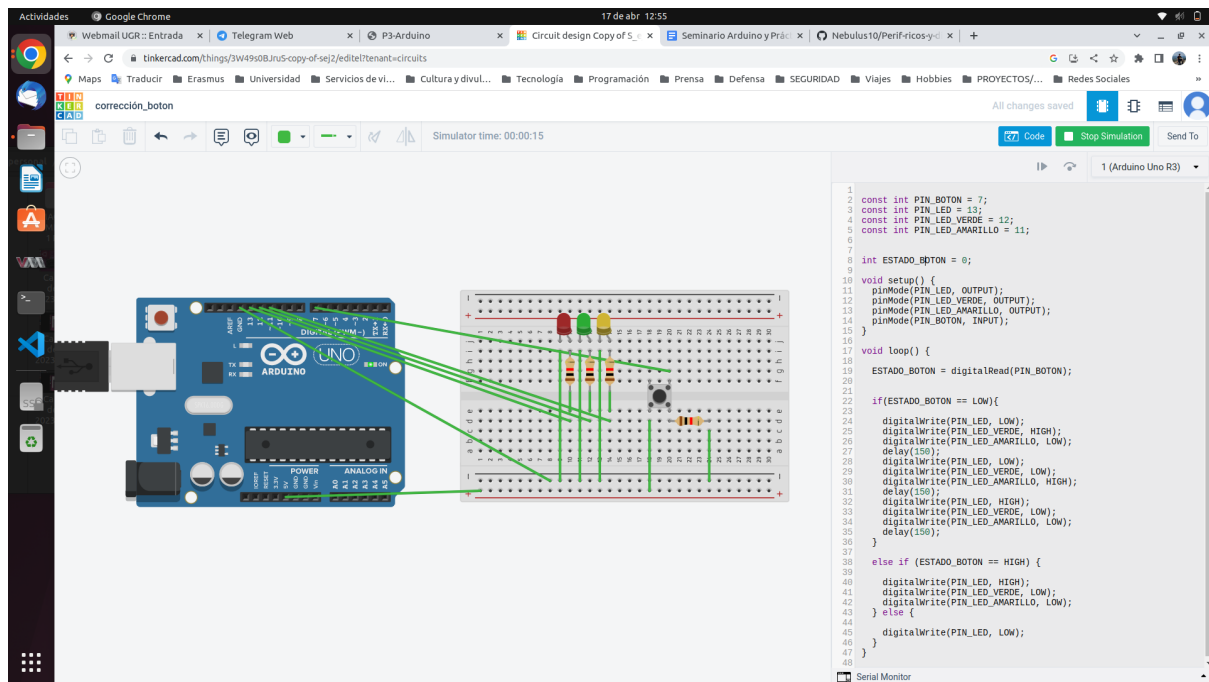
digitalWrite(PIN_LED, HIGH);
digitalWrite(PIN_LED_VERDE, LOW);
digitalWrite(PIN_LED_AMARILLO, LOW);
} else {

```

```

digitalWrite(PIN_LED, LOW); //En cualquier otro caso apagamos el rojo
}
}

```



1. Secuencia de LEDs, encendiendo y apagando 4 LEDs secuencialmente, de forma similar a las lucecitas de "El coche fantástico".

```

void setup() {

  for (int i = 3; i <= 13; i++) { //Para los pines logicos del 3 al 13, declaramos que esos
  pines son de salida

    pinMode(i, OUTPUT);

  }

}

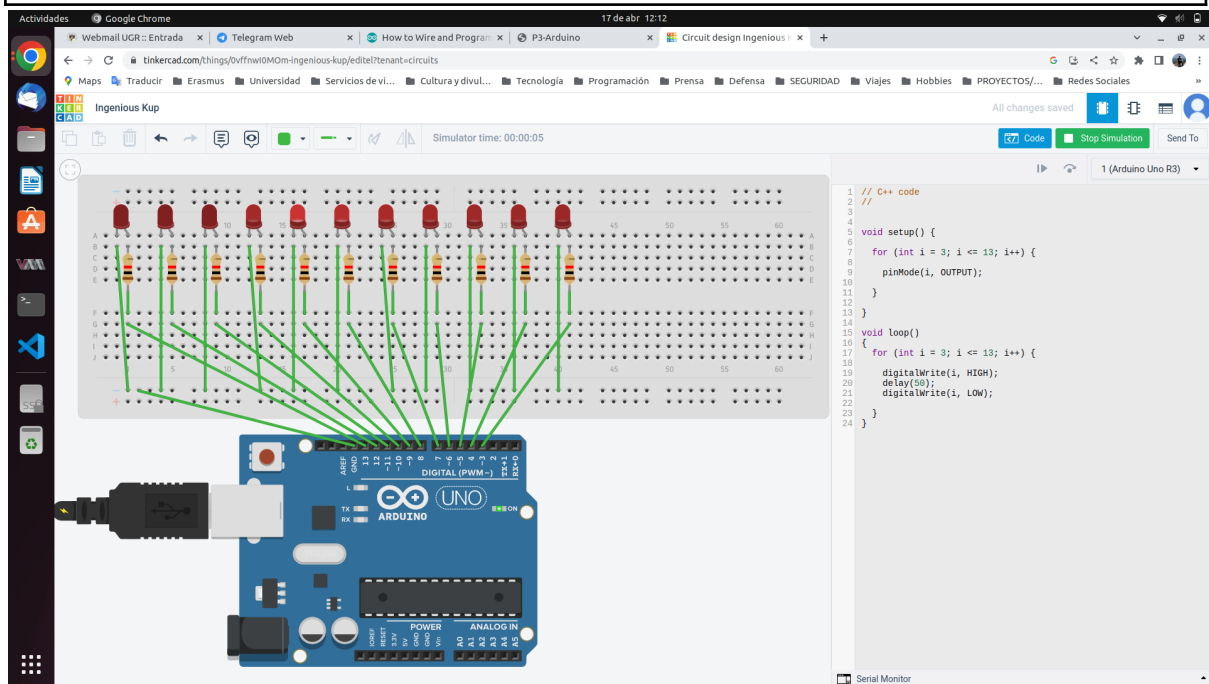
void loop()
{
  for (int i = 3; i <= 13; i++) { //Para cada pin, encendemos y apagamos el pin con un delay
  muy bajo para crear el efecto de cortina

    digitalWrite(i, HIGH);
    delay(50);
    digitalWrite(i, LOW);

  }

}

```



Disclaimer: He utilizado 10 LEDs en vez de 4 porque así daba un efecto más auténtico. El código es prácticamente el mismo, solo hay que cambiar el índice de los pines

2. Detector de la distancia a un objeto (usar el buzzer para hacer sonar un pitido en función de la distancia detectada por el sensor de ultrasonidos).

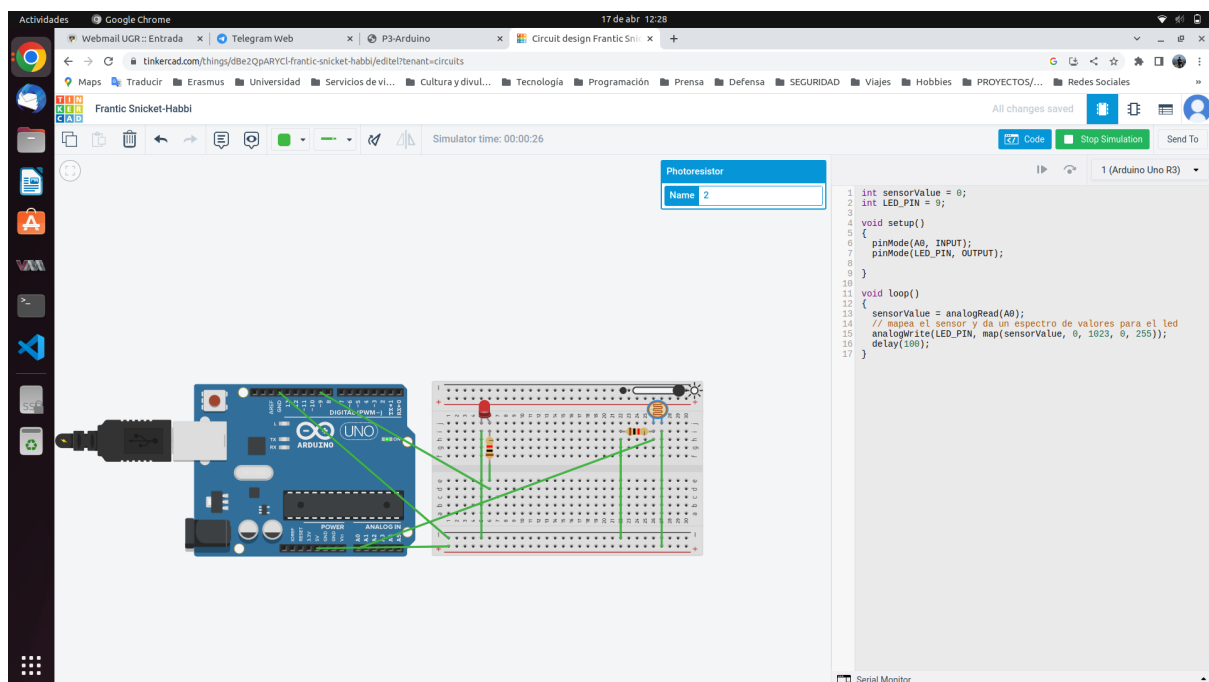
Este ejercicio se cubre con el trabajo personal de teoría que he realizado (Radar de proximidad con interfaz)

3. Detector de la cantidad de luz que haya en ese momento (usar un LED que se ilumine más o menos en función de la cantidad de luz detectada con el fotosensor).

```
int sensorValue = 0; //definimos los pines de los leds y del fotoresistor como globales
int LED_PIN = 9;

void setup()
{
  pinMode(A0, INPUT); //Indicamos que pines son de entrada y salida
  pinMode(LED_PIN, OUTPUT);
}

void loop()
{
  sensorValue = analogRead(A0); //Leemos del fotoresistor el valor que nos ofrece
  // mapea el sensor y da un espectro de valores para el led
  analogWrite(LED_PIN, map(sensorValue, 0, 1023, 0, 255));
  delay(100);
}
```



Versión del seminario del ejercicio 1 con solo dos leds:

```
int LED_PIN_ROJO = 12; //definimos los pines de los leds como globales
int LED_PIN_VERDE = 13;

void setup()
```

```

{
  pinMode(LED_PIN_ROJO, OUTPUT); //Indicamos que pines son de entrada y salida
  pinMode(LED_PIN_VERDE, OUTPUT);
}

void loop()
{
  digitalWrite(LED_PIN_ROJO, HIGH); //Encendemos el led
  digitalWrite(LED_PIN_VERDE, LOW); //Apagamos el led
  delay(1500); //Metemos una pausa en la iluminación
  digitalWrite(LED_PIN_ROJO, LOW);
  digitalWrite(LED_PIN_VERDE, HIGH);
  delay(1500);
}

```

