



*ugr* | Universidad  
de **Granada**

## PDIH

PERIFÉRICOS Y DISPOSITIVOS DE INTERFAZ HUMANA.

### Práctica 3: Experimentación con Arduino.

---

**Autora:** Cristina María Crespo Arco

**Correo:** cmcrespo@correo.ugr.es

**Autor:** Andrés Piqueras Brück

**Correo:** andrespquieras@correo.ugr.es

**Profesor:** Pedro A. Castillo Valdivieso



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

---

Curso 2021 - 2022

# Índice

<b>1. Primer ejercicio.</b>	<b>3</b>
1.1. Descripción del ejercicio . . . . .	3
1.2. Componentes eléctricos. . . . .	3
1.3. Código. . . . .	3
1.4. Esquema de conexiones eléctricas. . . . .	4
1.4.1. Vista de Circuito. . . . .	4
1.4.2. Vista esquemática. . . . .	4
1.5. Funcionamiento del proyecto: En el simulador Tinkercad. . . . .	5
1.6. Funcionamiento del proyecto. . . . .	5
<b>2. Segundo ejercicio: Versión interruptor.</b>	<b>6</b>
2.1. Descripción del ejercicio . . . . .	6
2.2. Componentes eléctricos. . . . .	6
2.3. Código. . . . .	6
2.4. Esquema de conexiones eléctricas. . . . .	7
2.4.1. Vista de Circuito. . . . .	7
2.4.2. Vista esquemática. . . . .	7
2.5. Funcionamiento del proyecto: En el simulador Tinkercad. . . . .	8
<b>3. Segundo ejercicio: Versión pulsador.</b>	<b>9</b>
3.1. Descripción del ejercicio . . . . .	9
3.2. Componentes eléctricos. . . . .	9
3.3. Código. . . . .	9
3.4. Esquema de conexiones eléctricas. . . . .	10
3.4.1. Vista de Circuito. . . . .	10
3.4.2. Vista esquemática. . . . .	10
3.5. Funcionamiento del proyecto: En el simulador Tinkercad. . . . .	11
3.6. Funcionamiento del proyecto. . . . .	12
<b>4. Tercer ejercicio.</b>	<b>13</b>
4.1. Descripción del ejercicio . . . . .	13
4.2. Componentes eléctricos. . . . .	13
4.3. Código. . . . .	13
4.4. Esquema de conexiones eléctricas. . . . .	14
4.4.1. Vista de Circuito. . . . .	14
4.4.2. Vista esquemática. . . . .	14
4.5. Funcionamiento del proyecto: En el simulador Tinkercad. . . . .	15
4.6. Funcionamiento del proyecto. . . . .	15
<b>5. Cuarto ejercicio.</b>	<b>16</b>
5.1. Descripción del ejercicio . . . . .	16
5.2. Componentes eléctricos. . . . .	16
5.3. Código. . . . .	16
5.4. Esquema de conexiones eléctricas. . . . .	17
5.4.1. Vista de Circuito. . . . .	17
5.4.2. Vista esquemática. . . . .	17
5.5. Funcionamiento del proyecto: En el simulador Tinkercad. . . . .	18

5.6. Funcionamiento del proyecto. . . . .	18
---	----

# 1. Primer ejercicio.

## 1.1. Descripción del ejercicio

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.

## 1.2. Componentes eléctricos.

- 1 Arduino Uno R3
- 3 Resistencias 220 Ω
- 1 LED Verde
- 1 LED Amarillo
- 1 LED Rojo
- 7 cables macho-macho

## 1.3. Código.

```
// C++ code
void setup(){
    // put your setup code here, to run once:
    pinMode(11, OUTPUT);      //pin de salida de la LED roja
    pinMode(12, OUTPUT);      //pin de salida de la LED amarilla
    pinMode(13, OUTPUT);      //pin de salida de la LED verde
}

void loop(){
    // put your main code here, to run repeatedly:
    digitalWrite(13, LOW);    //Apagar LED verde
    digitalWrite(11, HIGH);   //Encender LED roja
    delay(1500);             // Esperar durante 1,5 segundos
    digitalWrite(11, LOW);    //Apagar LED roja
    digitalWrite(12, HIGH);   //Encender LED amarilla
    delay(1500);             // Esperar durante 1,5 segundos
    digitalWrite(12, LOW);    //Apagar LED amarilla
    digitalWrite(13, HIGH);   //Encender LED verde
    delay(1500);             // Esperar durante 1,5 segundos
}
```

## 1.4. Esquema de conexiones eléctricas.

### 1.4.1. Vista de Circuito.

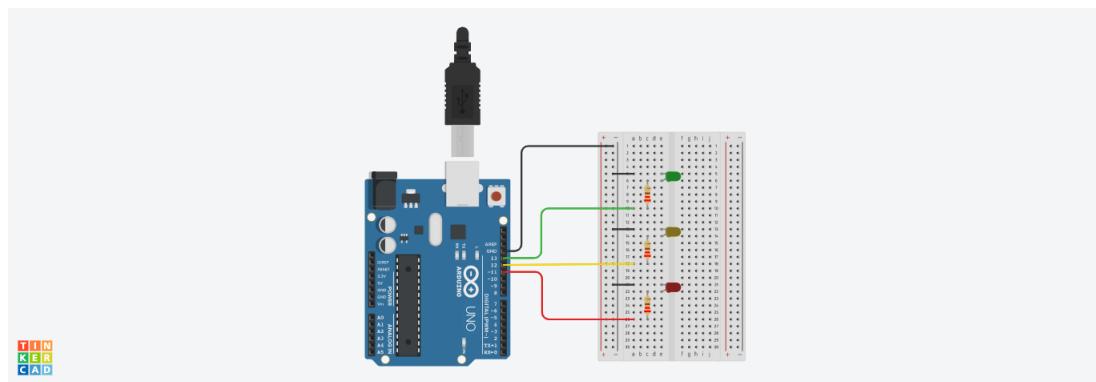


Figura 1: Circuito.

### 1.4.2. Vista esquemática.

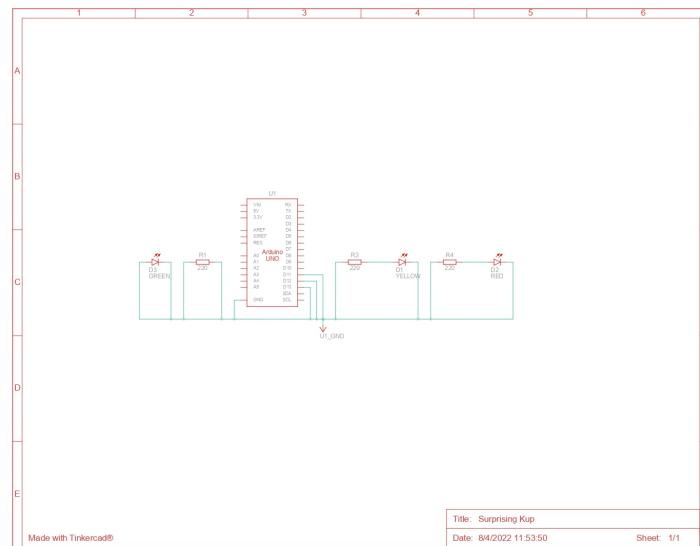


Figura 2: Esquema.

## 1.5. Funcionamiento del proyecto: En el simulador Tinkercad.

Enlace al proyecto en Tinkercad:

<https://www.tinkercad.com/things/3XWAtRAHLBh-led-semaforo/editel?sharecode=pLwf02e6YkntcCrNKS7YShTpB8K-nagr2CLsrJ0Ns7g>

Enlace al vídeo con una demostración del funcionamiento del proyecto en el simulador:

[https://github.com/cmcrespo00/PDIH/blob/main/P3/Ejercicio1/ejerV1\\_1.mp4](https://github.com/cmcrespo00/PDIH/blob/main/P3/Ejercicio1/ejerV1_1.mp4)

## 1.6. Funcionamiento del proyecto.

- ### ■ Circuito.

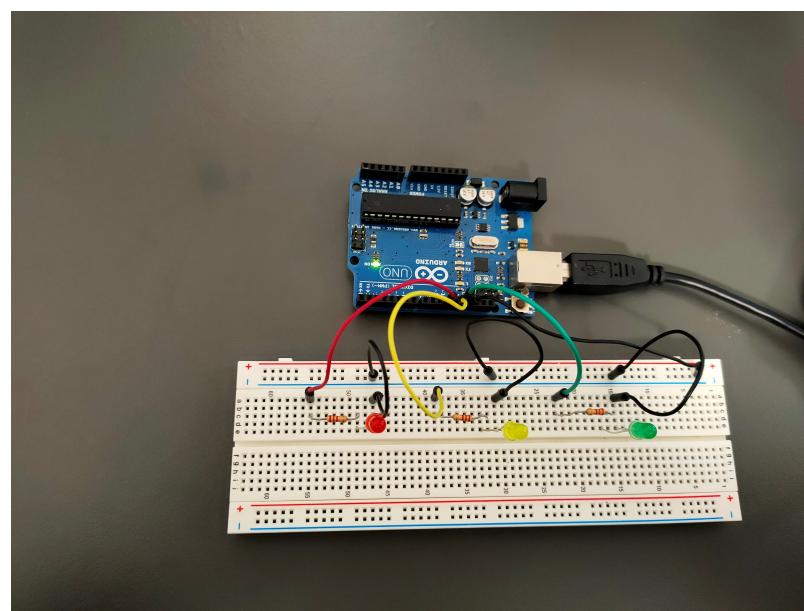


Figura 3: Arduino.

- Enlace al vídeo con una demostración del funcionamiento del proyecto:

[https://github.com/cmccrespo00/PDIH/blob/main/P3/Ejercicio1/ejerV1\\_2.mp4](https://github.com/cmccrespo00/PDIH/blob/main/P3/Ejercicio1/ejerV1_2.mp4)

## 2. Segundo ejercicio: Versión interruptor.

### 2.1. Descripción del ejercicio

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.

### 2.2. Componentes eléctricos.

- 1 Arduino Uno R3
- 4 Resistencias 220 Ω
- 1 LED Verde
- 1 LED Amarillo
- 1 LED Rojo
- 1 Interruptor deslizante
- 11 cables macho-macho

### 2.3. Código.

```
// C++ code
void setup(){
    // put your setup code here, to run once:
    pinMode(7, INPUT);           //pin de entrada del INTERRUPTOR
    pinMode(11, OUTPUT);         //pin de salida de la LED roja
    pinMode(12, OUTPUT);         //pin de salida de la LED amarilla
    pinMode(13, OUTPUT);         //pin de salida de la LED verde
}
void loop(){
    // put your main code here, to run repeatedly:
    //Comprueba si el interruptor está encendido
    if ((digitalRead(7) == true)) {
        digitalWrite(11, HIGH);      //Encender LED roja
        digitalWrite(12, LOW);       //Apagar LED amarilla
        digitalWrite(13, LOW);       //Apagar LED verde
    }
    //Comprueba si el interruptor está apagado
    else if ((digitalRead(7) == false)) {
        digitalWrite(11, LOW);      //Apagar LED roja
        digitalWrite(12, HIGH);      //Encender LED amarilla
        digitalWrite(13, HIGH);      //Encender LED verde
    }
}
```

## 2.4. Esquema de conexiones eléctricas.

### 2.4.1. Vista de Circuito.

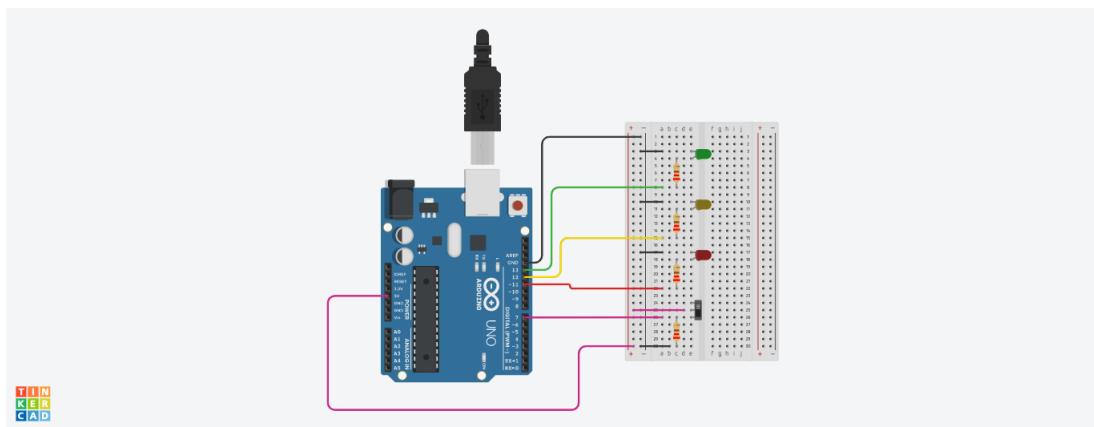


Figura 4: Circuito.

### 2.4.2. Vista esquemática.

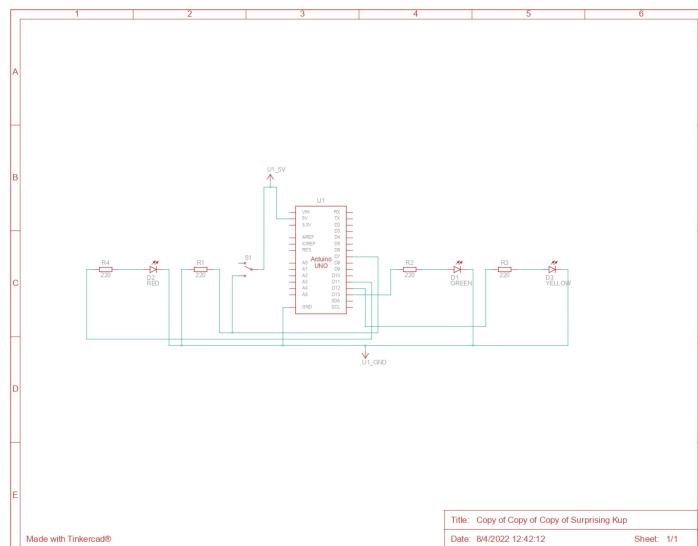


Figura 5: Esquema.

## 2.5. Funcionamiento del proyecto: En el simulador Tinkercad.

Enlace al proyecto en Tinkercad:

[https://www.tinkercad.com/things/9FHKK9Vtske-interruptor/editel?sharecode=6YUyp47uLKFe4RN136j1X8Fz\\_Q\\_11kqs4A1t7umCLvA](https://www.tinkercad.com/things/9FHKK9Vtske-interruptor/editel?sharecode=6YUyp47uLKFe4RN136j1X8Fz_Q_11kqs4A1t7umCLvA)

- ### ■ Circuito.

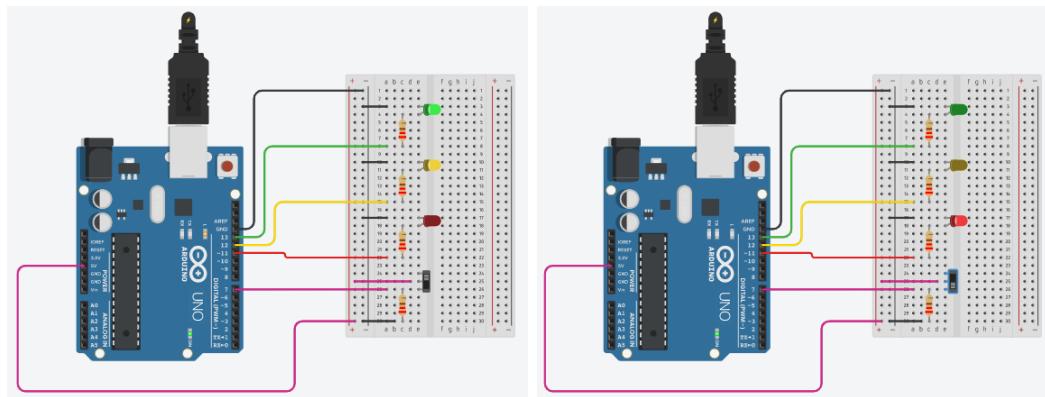


Figura 6: Interruptor apagado - Interruptor encendido.

- Enlace al vídeo con una demostración del funcionamiento del proyecto en el simulador:

[https://github.com/cmcrespo00/PDIH/blob/main/P3/Ejercicio2/ejerV2\\_1.mp4](https://github.com/cmcrespo00/PDIH/blob/main/P3/Ejercicio2/ejerV2_1.mp4)

### 3. Segundo ejercicio: Versión pulsador.

#### 3.1. Descripción del ejercicio

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.

#### 3.2. Componentes eléctricos.

- 1 Arduino Uno R3
- 4 Resistencias 220 Ω
- 1 LED Verde
- 1 LED Amarillo
- 1 LED Rojo
- 1 Pulsador
- 11 cables macho-macho

#### 3.3. Código.

```
// C++ code
void setup(){
    // put your setup code here, to run once:
    pinMode(7, INPUT);           //pin de entrada del INTERRUPTOR
    pinMode(11, OUTPUT);         //pin de salida de la LED roja
    pinMode(12, OUTPUT);         //pin de salida de la LED amarilla
    pinMode(13, OUTPUT);         //pin de salida de la LED verde
}
void loop(){
    // put your main code here, to run repeatedly:
    //Comprueba si el pulsador está pulsado
    if ((digitalRead(7) == true)) {
        digitalWrite(11, HIGH);      //Encender LED roja
        digitalWrite(12, LOW);       //Apagar LED amarilla
        digitalWrite(13, LOW);       //Apagar LED verde
    }
    //Comprueba si el pulsador no está pulsado
    else if ((digitalRead(7) == false)) {
        digitalWrite(11, LOW);      //Apagar LED roja
        digitalWrite(12, HIGH);     //Encender LED amarilla
        digitalWrite(13, HIGH);     //Encender LED verde
    }
}
```

### 3.4. Esquema de conexiones eléctricas.

#### 3.4.1. Vista de Circuito.

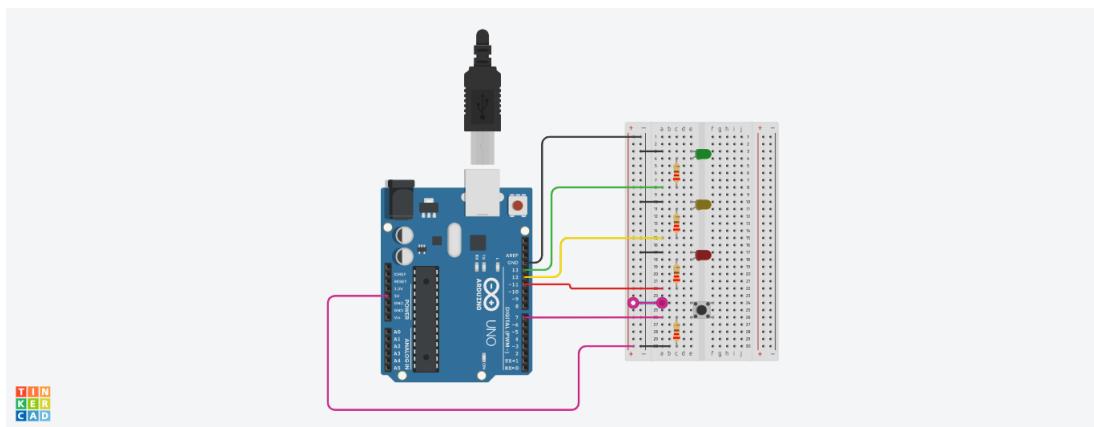


Figura 7: Circuito.

#### 3.4.2. Vista esquemática.

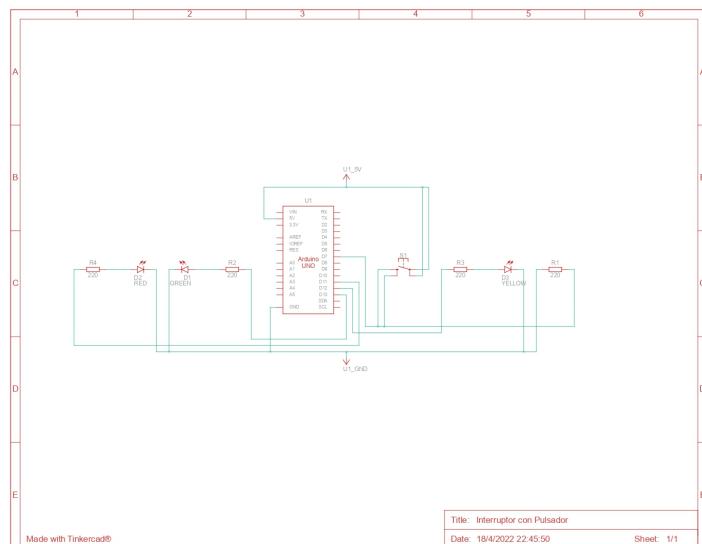


Figura 8: Esquema.

### 3.5. Funcionamiento del proyecto: En el simulador Tinkercad.

Enlace al proyecto en Tinkercad:

<https://www.tinkercad.com/things/2pPMAusq1Fu-interruptor-con-pulsador/editel?sharecode=QtdMwdQ01J2EGGbErEFm3aBVrP17Zu6JIKG3fMG7SFU>

- Circuito.

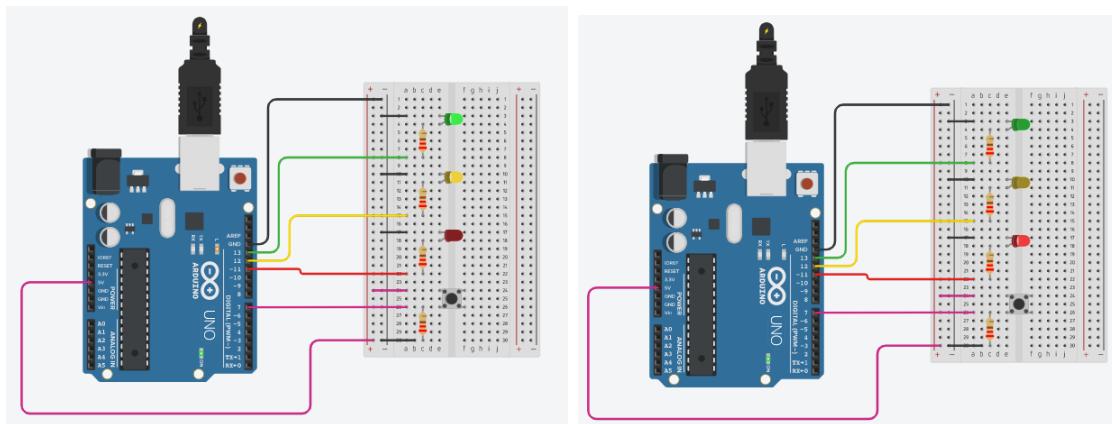


Figura 9: Interruptor apagado - Interruptor encendido.

- Enlace al vídeo con una demostración del funcionamiento del proyecto en el simulador:

[https://github.com/cmcrespo00/PDIH/blob/main/P3/Ejercicio2/ejerV2\\_11.mp4](https://github.com/cmcrespo00/PDIH/blob/main/P3/Ejercicio2/ejerV2_11.mp4)

### 3.6. Funcionamiento del proyecto.

- Circuito.

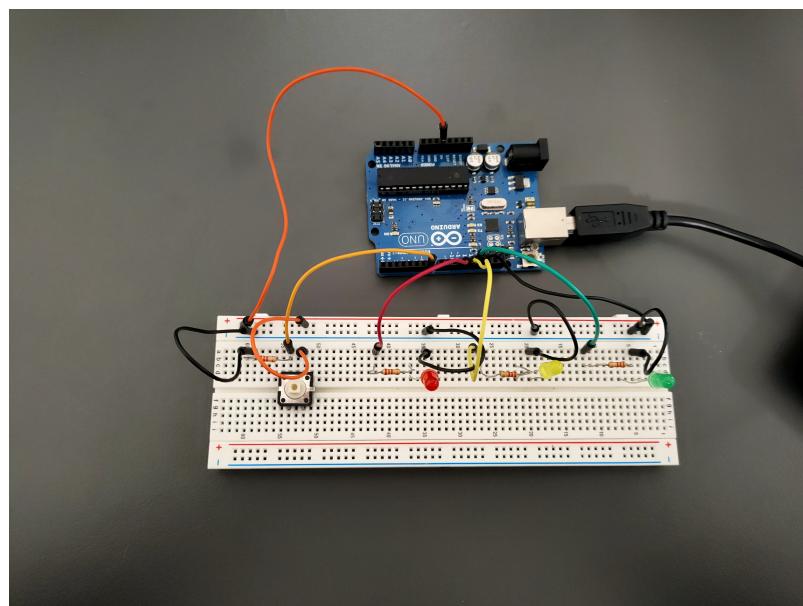


Figura 10: Arduino.

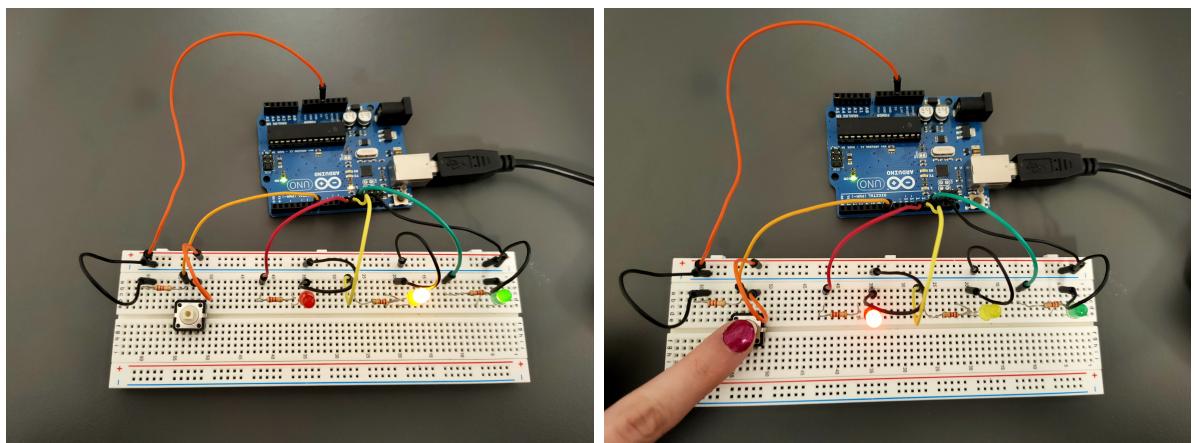


Figura 11: Interruptor apagado - Interruptor encendido.

- Enlace al vídeo con una demostración del funcionamiento del proyecto:

[https://github.com/cmcrespo00/PDIH/blob/main/P3/Ejercicio2/ejerV2\\_12.mp4](https://github.com/cmcrespo00/PDIH/blob/main/P3/Ejercicio2/ejerV2_12.mp4)

## 4. Tercer ejercicio.

### 4.1. Descripción del ejercicio

Secuencia de LEDs, encendiendo y apagando 4 LEDs secuencialmente, de forma similar a las luces de “El coche fantástico”.

### 4.2. Componentes eléctricos.

- 1 Arduino Uno R3
- 4 Resistencias 220 Ω
- 4 LED Rojo
- 9 cables macho-macho

### 4.3. Código.

```
// C++ code
void setup(){
    // put your setup code here, to run once:
    pinMode(10, OUTPUT);      //pin de salida de una LED
    pinMode(11, OUTPUT);      //pin de salida de una LED
    pinMode(12, OUTPUT);      //pin de salida de una LED
    pinMode(13, OUTPUT);      //pin de salida de una LED
}
void loop(){
    // put your main code here, to run repeatedly:
    digitalWrite(11, LOW);    //Apagar LED
    digitalWrite(10, HIGH);    //Encender LED
    delay(200);              // Esperar durante 0,2 segundos
    digitalWrite(10, LOW);    //Apagar LED
    digitalWrite(11, HIGH);    //Encender LED
    delay(200);              // Esperar durante 0,2 segundos
    digitalWrite(11, LOW);    //Apagar LED
    digitalWrite(12, HIGH);    //Encender LED
    delay(200);              // Esperar durante 0,2 segundos
    digitalWrite(12, LOW);    //Apagar LED
    digitalWrite(13, HIGH);    //Encender LED
    delay(200);              // Esperar durante 0,2 segundos
    digitalWrite(13, LOW);    //Apagar LED
    digitalWrite(12, HIGH);    //Encender LED
    delay(200);              // Esperar durante 0,2 segundos
    digitalWrite(12, LOW);    //Apagar LED
    digitalWrite(11, HIGH);    //Encender LED
    delay(200);              // Esperar durante 0,2 segundos
}
```

## 4.4. Esquema de conexiones eléctricas.

### 4.4.1. Vista de Circuito.

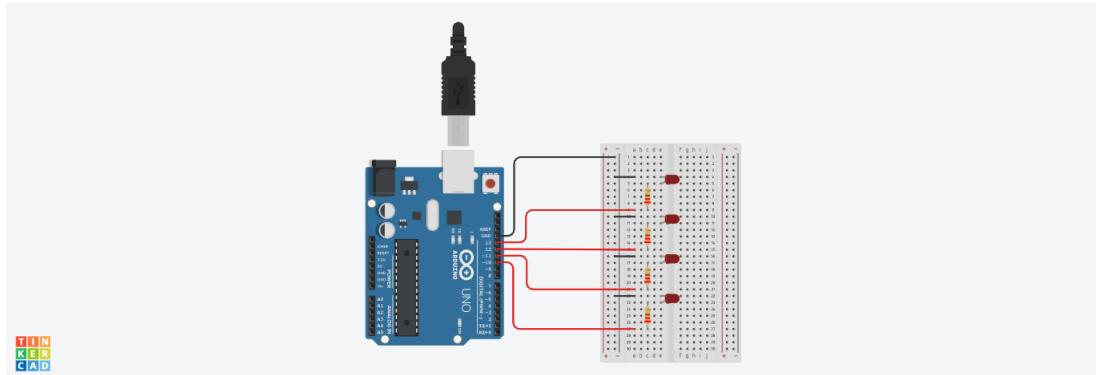


Figura 12: Circuito.

### 4.4.2. Vista esquemática.

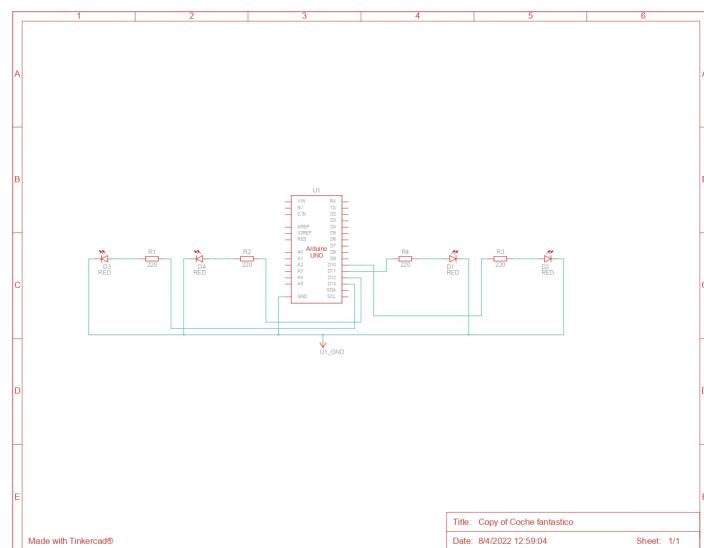


Figura 13: Esquema.

## 4.5. Funcionamiento del proyecto: En el simulador Tinkercad.

Enlace al proyecto en Tinkercad:

<https://www.tinkercad.com/things/7sFM81t13hl-coche-fantastico/editel?sharecode=1WBXki0tK04plnRnuB-RTrjcCDjAs2pG8HDVQgS3WRc>

Enlace al vídeo con una demostración del funcionamiento del proyecto en el simulador:

[https://github.com/cmcrespo00/PDIH/blob/main/P3/Ejercicio3/ejerV3\\_1.mp4](https://github.com/cmcrespo00/PDIH/blob/main/P3/Ejercicio3/ejerV3_1.mp4)

## 4.6. Funcionamiento del proyecto.

- Circuito.

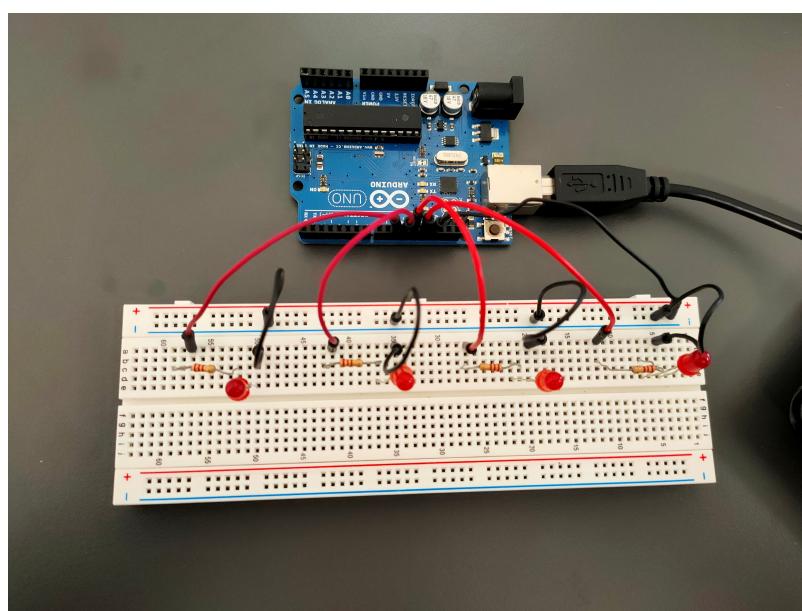


Figura 14: Arduino.

- Enlace al vídeo con una demostración del funcionamiento del proyecto:

[https://github.com/cmcrespo00/PDIH/blob/main/P3/Ejercicio3/ejerV3\\_2.mp4](https://github.com/cmcrespo00/PDIH/blob/main/P3/Ejercicio3/ejerV3_2.mp4)

## 5. Cuarto ejercicio.

### 5.1. Descripción del ejercicio

Alarma por detección de presencia.

### 5.2. Componentes eléctricos.

- 1 Arduino Uno R3
- 2 Resistencias 220 Ω
- 1 LED Rojo
- 1 Sensor PIR
- 7 cables macho-macho
- 3 cables macho-hembra

### 5.3. Código.

```
// C++ code
void setup(){
    // put your setup code here, to run once:
    pinMode(13, OUTPUT);           //pin de salida de la LED roja
    pinMode(7, INPUT);            //pin de entrada del SENSOR PIR
}

void loop(){
    // put your main code here, to run repeatedly:
    //Comprueba si hay movimiento cerca del sensor PIR
    if ((digitalRead(7) == HIGH)) {
        digitalWrite(13, HIGH);      //Encender LED roja
    } else {
        digitalWrite(13, LOW);       //Apagar LED roja
    }
}
```

#### 5.4. Esquema de conexiones eléctricas.

#### **5.4.1. Vista de Circuito.**

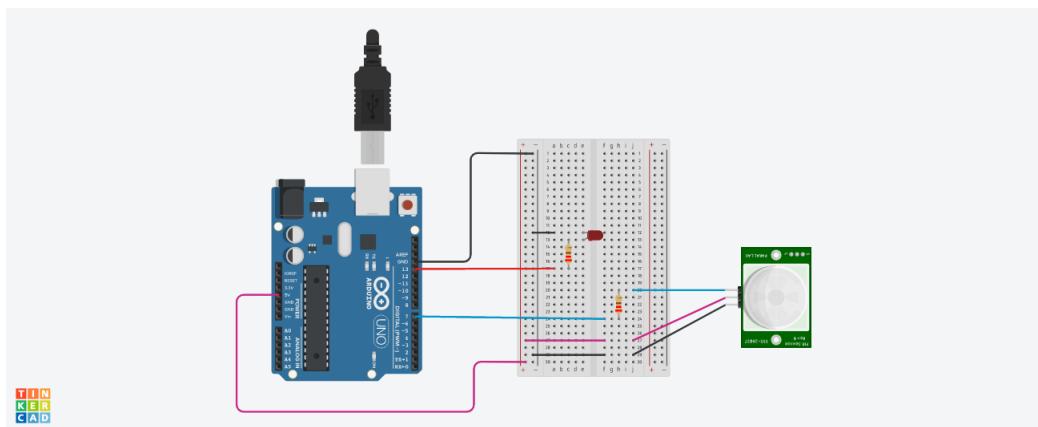


Figura 15: Circuito.

#### 5.4.2. Vista esquemática.

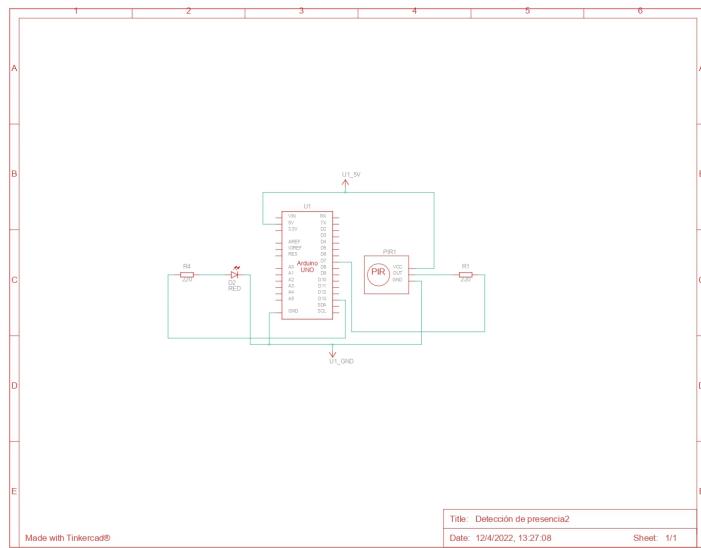


Figura 16: Esquema.

## 5.5. Funcionamiento del proyecto: En el simulador Tinkercad.

Enlace al proyecto en Tinkercad:

<https://www.tinkercad.com/things/0BfYXRWFB6p-deteccion-de-presencia/editel?sharecode=rJvfEQ8DL8dmhJyHp5KW9EdA01TfFvjVPImMQC3PTTY>

Enlace al vídeo con una demostración del funcionamiento del proyecto en el simulador:

[https://github.com/cmcrespo00/PDIH/blob/main/P3/Ejercicio4/ejerV4\\_1.mp4](https://github.com/cmcrespo00/PDIH/blob/main/P3/Ejercicio4/ejerV4_1.mp4)

## 5.6. Funcionamiento del proyecto.

- Circuito.

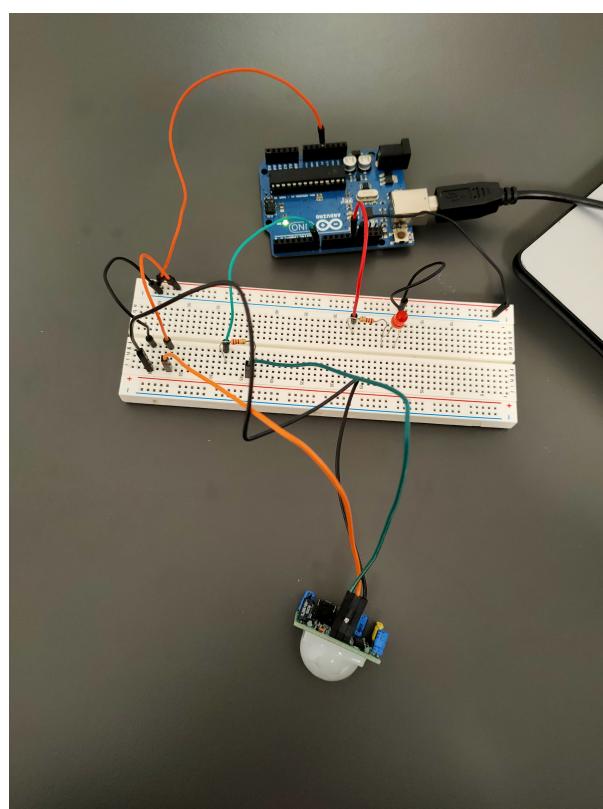


Figura 17: Arduino.

- Enlace al vídeo con una demostración del funcionamiento del proyecto:

[https://github.com/cmcrespo00/PDIH/blob/main/P3/Ejercicio4/ejerV4\\_2.mp4](https://github.com/cmcrespo00/PDIH/blob/main/P3/Ejercicio4/ejerV4_2.mp4)