



## TEMA 2 ENTRADAS Y SALIDAS DIGITALES

### Qué es una Entrada y una Salida Digital?¿

Entrada Digital: Arduino recibe señales digitales (valores de 0 o 1) desde sensores o dispositivos. Por ejemplo, un pulsador que se presiona o no.

Salida Digital: Arduino envía señales digitales para controlar dispositivos como LEDs, relés, motores, etc., activándolos (HIGH) o desactivándolos (LOW).

### 2. Configuración de Pines Digitales

#### 2.1 pinMode(pin, mode)

Configura un pin como entrada o salida.

pin: Número del pin digital que quieres usar.

mode: Modo del pin (INPUT, OUTPUT, INPUT\_PULLUP).

#### Ejemplo:

```
pinMode(2, INPUT);    // Configurar pin 2 como entrada
pinMode(13, OUTPUT);  // Configurar pin 13 como salida
pinMode(3, INPUT_PULLUP); // Entrada con resistencia pull-up interna
```

### 3. Lectura de Entradas Digitales

digitalRead(pin)

- Lee el estado de un pin configurado como entrada.
- Devuelve HIGH (5V) o LOW (0V).

Ejemplo: Leer el estado de un botón conectado al pin 2.

#### Ejemplo

int buttonState;

void setup() {

pinMode(2, INPUT); // Configurar el pin 2 como entrada

Serial.begin(9600); // Iniciar comunicación serial

}

void loop() {





```
buttonState = digitalRead(2); // Leer el estado del botón  
if (buttonState == HIGH) {  
    Serial.println("Botón presionado");  
} else {  
    Serial.println("Botón no presionado");  
}  
delay(500);  
}
```

## 4. Escritura en Salidas Digitales

digitalWrite(pin, value)

- Configura un pin de salida en HIGH o LOW.
- value: HIGH (enciende el dispositivo) o LOW (lo apaga).

Ejemplo: Encender y apagar un LED conectado al pin 13.

Ejemplo

```
void setup() {  
    pinMode(13, OUTPUT); // Configurar el pin 13 como salida  
}  
  
void loop() {  
    digitalWrite(13, HIGH); // Encender LED  
    delay(1000);           // Esperar 1 segundo  
    digitalWrite(13, LOW); // Apagar LED  
    delay(1000);           // Esperar 1 segundo  
}
```





## 5. Uso de Resistencia Pull-Up Interna

Arduino puede activar una resistencia **pull-up** interna para garantizar que un pin no quede "flotando" cuando no se aplica señal.

### Modo INPUT\_PULLUP

- Configura el pin como entrada con resistencia pull-up habilitada.
- Muy útil para botones o interruptores.

Ejemplo :

```
void setup() {  
  pinMode(2, INPUT_PULLUP); // Configurar pin 2 con resistencia pull-up  
}
```

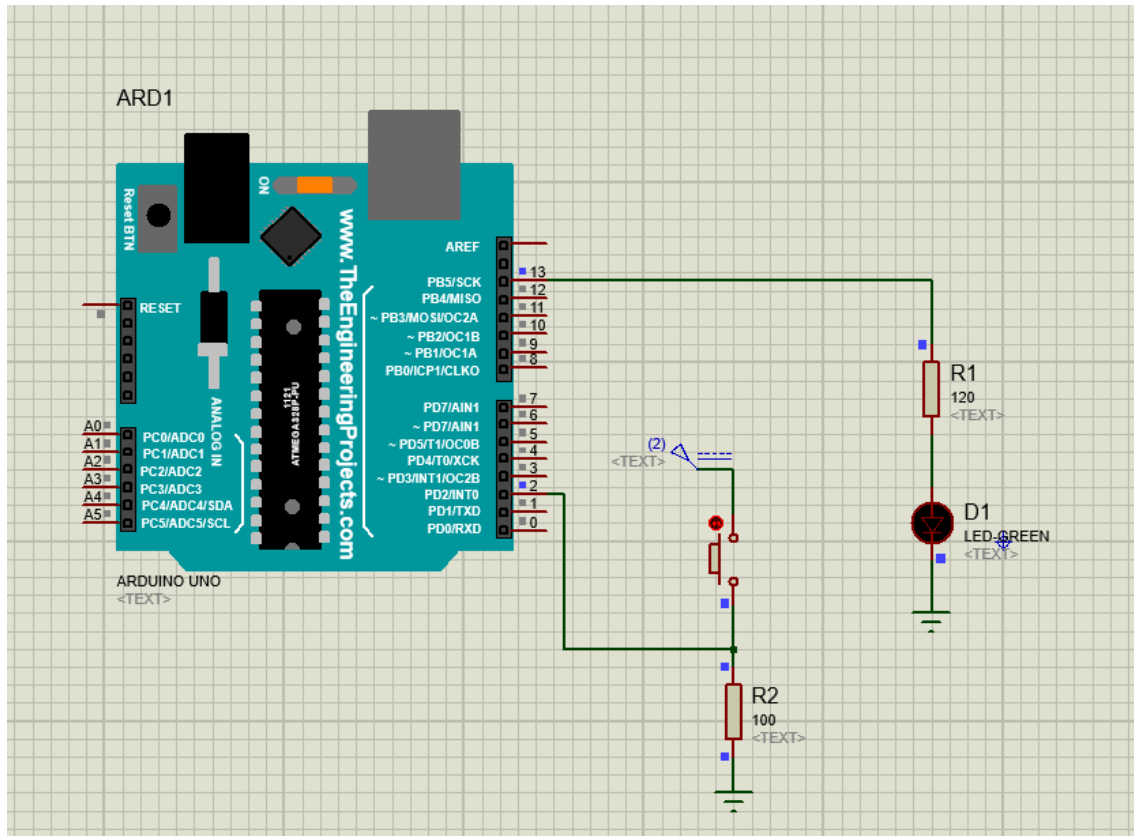
```
void loop() {  
  if (digitalRead(2) == LOW) { // El botón está presionado  
    Serial.println("Botón presionado");  
  } else {  
    Serial.println("Botón no presionado");  
  }  
  delay(500);  
}
```





## PROGRAMA EJEMPLOS

### 2.1.CONTROL DE LED CON PULSADOR



#### PROGRAMA

```
const int ledPin = 13; // LED conectado al pin 13
const int pulsador = 2; // Pulsador conectado al pin 2

void setup() {
  pinMode(ledPin, OUTPUT); // Configurar el pin del LED como salida
  pinMode(pulsador, INPUT_PULLUP); // Habilitar resistencia pull-up interna
}

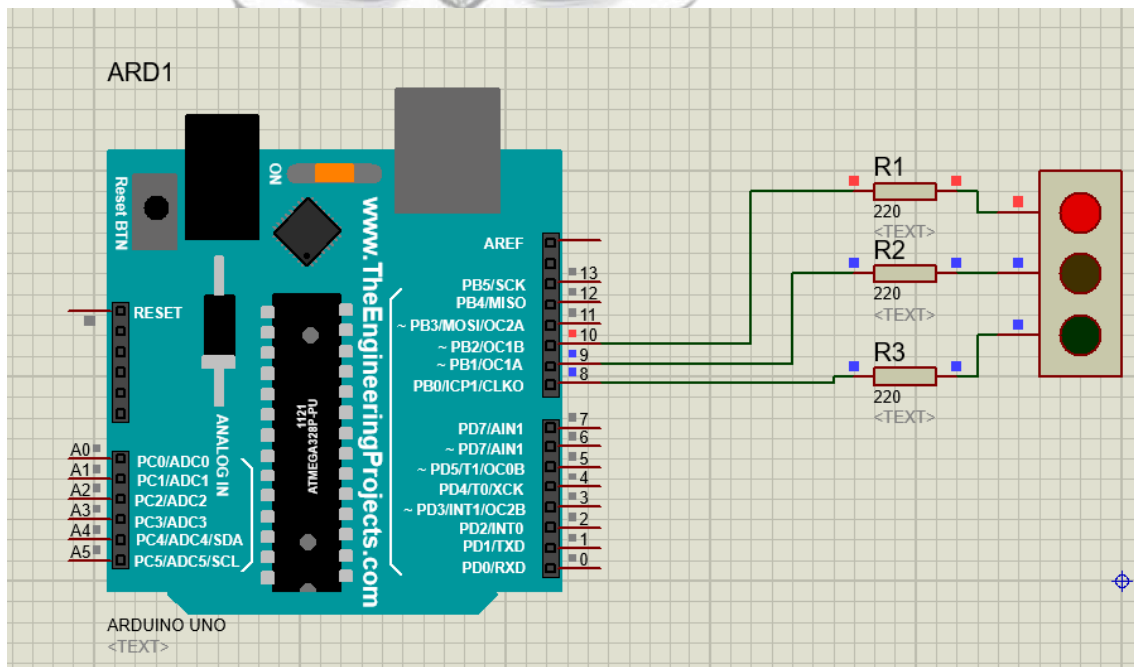
void loop() {
  // Leer el estado del pulsador (inverso debido a la resistencia pull-up)
  int buttonState = digitalRead(pulsador);
```





```
if (buttonState == HIGH) { // Pulsador presionado (con pull-up, el estado es LOW)
    digitalWrite(13, HIGH); // Encender LED
} else {
    digitalWrite(13, LOW); // Apagar LED
}
}
```

## 2.2 PARPADEO DE LEDS SEMÁFORO



### Materiales necesarios

1. Arduino Uno (o cualquier otra placa Arduino).
2. 3 LEDs (rojo, amarillo, verde).
3. 3 resistencias de 220 ohmios (una por cada LED).
4. Cables de conexión.
5. Protoboard (opcional, para conexiones organizadas).







## Esquema de conexión

1. Conecta el ánodo del LED rojo al pin digital 8.
2. Conecta el ánodo del LED amarillo al pin digital 9.
3. Conecta el ánodo del LED verde al pin digital 10.
4. Conecta los cátodos de cada LED a una resistencia de 220 ohmios, y luego a GND.

## PROGRAMA

```
// Pines de los LEDs

const int ledRojo = 8;  // LED rojo conectado al pin 8
const int ledAmarillo = 9; // LED amarillo conectado al pin 9
const int ledVerde = 10; // LED verde conectado al pin 10

void setup() {
  // Configurar los pines como salida
  pinMode(ledRojo, OUTPUT);
  pinMode(ledAmarillo, OUTPUT);
  pinMode(ledVerde, OUTPUT);
}

void loop() {
  // Encender LED rojo (stop)
  digitalWrite(ledRojo, HIGH);
  digitalWrite(ledAmarillo, LOW);
  digitalWrite(ledVerde, LOW);
  delay(5000); // Mantener encendido por 5 segundos
  // Encender LED amarillo (precaución)
  digitalWrite(ledRojo, LOW);
```

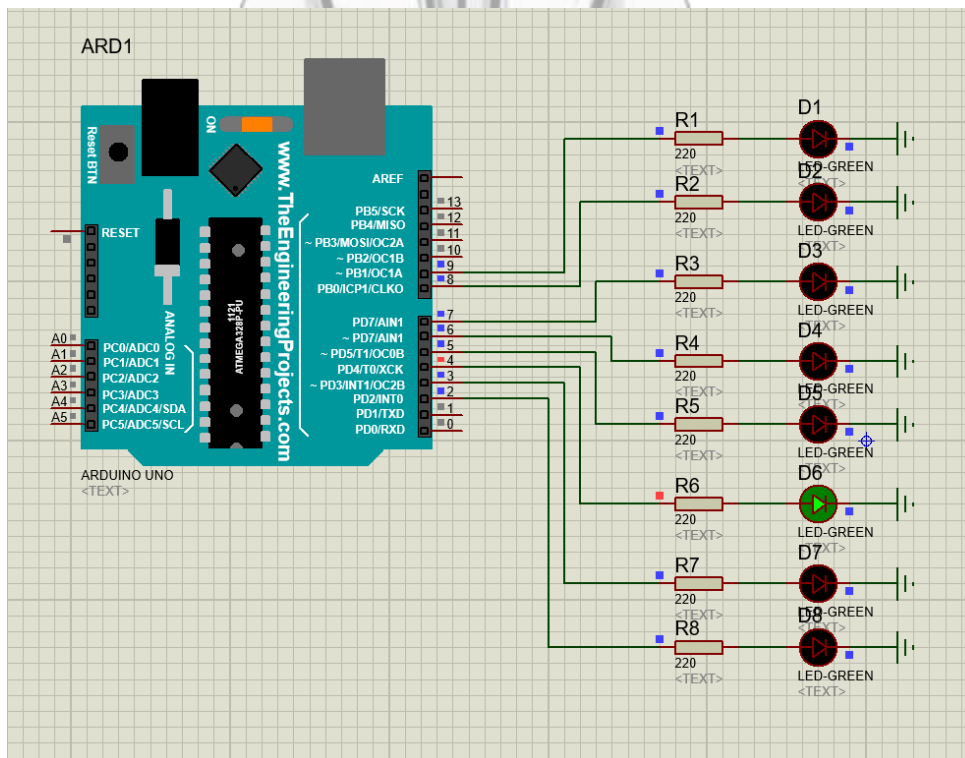




# TECH LAB ACADEMY ELECTRONIKA

```
digitalWrite(ledAmarillo, HIGH);  
  
digitalWrite(ledVerde, LOW);  
  
delay(2000); // Mantener encendido por 2 segundos  
  
// Encender LED verde (go)  
  
digitalWrite(ledRojo, LOW);  
  
digitalWrite(ledAmarillo, LOW);  
  
digitalWrite(ledVerde, HIGH);  
  
delay(5000); // Mantener encendido por 5 segundos  
  
// Regresar al LED amarillo (precaución)  
  
digitalWrite(ledRojo, LOW);  
  
digitalWrite(ledAmarillo, HIGH);  
  
digitalWrite(ledVerde, LOW);  
  
delay(2000); // Mantener encendido por 2 segundos  
  
}
```

## 2.3.LUCES SECUENCIALES 8 LEDS





# TECH LAB ACADEMY ELECTRONIKA

## Materiales necesarios:

1. Arduino Uno o similar.
2. 8 LEDs.
3. 8 Resistencias de  $220\Omega$ .
4. Protoboard.
5. Cables de conexión.

## Conexiones:

### 1. LEDs:

- Conecta el ánodo (pierna más larga) de cada LED a un pin digital del Arduino.
- Conecta el cátodo (pierna más corta) de cada LED a GND a través de una resistencia de  $220\Omega$ .

### 2. Pines de conexión:

- LED 1 → Pin digital 2.
- LED 2 → Pin digital 3.
- LED 3 → Pin digital 4.
- LED 4 → Pin digital 5.
- LED 5 → Pin digital 6.
- LED 6 → Pin digital 7.
- LED 7 → Pin digital 8.
- LED 8 → Pin digital 9.

## PROGRAMA

// Pines donde están conectados los LEDs

```
int leds[] = {2, 3, 4, 5, 6, 7, 8, 9};
```

```
int numLeds = 8; // Número de LEDs
```







# TECH LAB ACADEMY ELECTRONIKA

```
void setup() {  
    // Configurar los pines de los LEDs como salida  
    for (int i = 0; i < numLeds; i++) {  
        pinMode(leds[i], OUTPUT);  
    }  
}  
  
void loop() {  
    // Encender los LEDs uno por uno  
    for (int i = 0; i < numLeds; i++) {  
        digitalWrite(leds[i], HIGH); // Encender LED  
        delay(200); // Esperar 200 ms  
        digitalWrite(leds[i], LOW); // Apagar LED  
    }  
}
```

