



# Control programado con Arduino

## Práctica 3: O pulsador momentáneo (Funcionamento estable)

Ata agora, os nosos circuítos só enviaban información cara ao exterior acendendo LEDs (canles de saída). Nesta práctica daremo un paso crucial: aprenderemos a recibir datos do exterior utilizando un pulsador (canle de entrada). Deseñaremos un programa intelixente que tome unha decisión lóxica e acenda un LED só mentres manteñamos premido o pulsador.

## Obxectivos de Aprendizaxe

[1pt]

Ao rematar esta sesión serás capaz de:

- Configurar pins dixitais como canles de entrada de datos empregando `pinMode` `INPUT`.
- Ler e monitorizar estados dixitais (valores lóxicos HIGH e LOW) coa orde `digitalRead`.
- Aplicar estruturas de control condicionais mediante a instrución `if else`.
- Entender a necesidade eléctrica de utilizar unha resistencia de protección (Pull-Down) para evitar o estado flotante (ruído eléctrico).

## Compoñentes Necesarios

[1pt]

Busca e coloca os seguintes compoñentes na túa mesa de traballo de TinkerCad:

- 1 Placa Arduino Uno.
- 1 Placa de prototipado pequena (Protoboard).
- 1 Diodo LED e 1 resistencia de  $220\ \Omega$  (para o LED).
- 1 Pulsador de catro patillas.
- 1 Resistencia de  $10\ k\Omega$  (código de cores: marrón, negro, laranxa; usarase como Pulldown).
- Cables de Amazon de conexión virtuais.



## Código Base: O condicional if/else

[1pt]

Escribe o seguinte programa no editor de código de TinkerCad. Observa con atención como o código decide de xeito continuo o estado do LED lendo a voltaxe do pulsador.

```
1 // Declaracion de variables asignando os pins de conexion
2 int led = 8;           // O LED esta conectado no pin dixital 8
3 int pulsador = 2;     // O pulsador esta conectado no pin dixital 2
4
5 void setup() {
6   pinMode(led, OUTPUT); // Configuramos o LED como SAIDA de corrente
7   pinMode(pulsador, INPUT); // Configuramos o pulsador como ENTRADA de datos
8 }
9
10 void loop() {
11   // Creamos unha variable local para gardar o estado do pulsador (HIGH ou LOW)
12   int estadoPulsador = digitalRead(pulsador);
13
14   // Condicional: Se o pulsador esta premido (recibe 5V / HIGH)
15   if (estadoPulsador == HIGH) {
16     digitalWrite(led, HIGH); // Acende o LED
17   }
18   // En caso contrario (se no esta premido / recibe 0V / LOW)
19   else {
20     digitalWrite(led, LOW); // Apaga o LED
21   }
22 }
```

Listing 1: Código para controlar un LED cun pulsador momentáneo.

## Como funciona o código?

[1pt]

- `pinMode` `INPUT` : Indícalle ao microcontrolador que o pin 2 no vai subministrar corrente, senón que vai "escoitar" de forma pasiva a voltaxe que lle chegue do circuíto exterior.
- `digitalRead` : É un sensor lóxico. Le a voltaxe no pin indicado. Se detecta 5V devolve-ranos o valor lóxico `HIGH` (alto), e si detecta 0V (GND) devolveranos `LOW` (baixo).
- **A comparación** : En programación, un só símbolo de igual ( `=` ) serve para gardar un valor dentro dunha variable. Para comparar se dúas cousas son matematicamente iguais, debemos usar obrigatoriamente o dobre igual ( `==` ).
- **if else (Se / Se non)**: Avalía unha condición lóxica entre parénteses. Se a condición se cumpre, execútanse as liñas encerradas entre as chaves do `if`. Se a condición é falsa, o programa ignora ese bloque e executa directamente o que hai dentro das chaves do `else`.

### O Reto da Práctica 3

Demostra a túa comprensión sobre o control do fluxo do código superando estes tres retos:

1. **Lóxica Inversa:** Modifica o programa en TinkerCad para que o LED estea sempre acendido por defecto, e se **apague** unicamente cando manteñas premido o pulsador.
2. **Semáforo Peonil de dúas posicións:** Engade un segundo LED (por exemplo, verde no pin 9 e vermello no pin 8). Fai que, por defecto, o LED vermello estea acendido e o verde apagado. Ao premer o pulsador, o vermello debe apagarse e o verde acenderse.
3. *Pregunta para reflexionar:* Se retiramos a resistencia de  $10\text{ k}\Omega$  de la protoboard e conectamos o pulsador directamente a GND e a 5V sen ela, que tipo de anomalía eléctrica ou física cres que podería ocorrerlle á placa de Arduino ao premer o botón?