# Manual Explicado desde Cero — Módulo 02: Semáforos, Botones (`INPUT_PULLUP`) y Sonidos

> [!NOTE]
>
> En este módulo aprenderás cómo darle **capacidad de decisión** a tu Arduino para que reaccione cuando presionas un botón y coordine varias luces y sonidos a la vez.

---

## 1. Variables con Nombres Claros: Poniéndole Etiquetas a los Pines

Cuando conectamos varios componentes, recordar de memoria qué número de pin corresponde a cada luz es incómodo. Por eso usamos **variables**:

```
int ledVerde    = 8;
int ledAmarillo = 9;
int ledRojo     = 10;
```

- La palabra `int` significa *número entero* (sin decimales).
- Piensa en una variable como **pegarle una etiqueta con nombre a un cajón**: en vez de recordar "el cajón número 8", simplemente escribes `ledVerde` en todo tu programa y el Arduino reemplaza esa palabra por el número `8` automáticamente.

---

## 2. El Secreto de `INPUT_PULLUP`: Cómo Conectar un Botón sin Enredarse

Un botón pulsador es simplemente un puentecito metálico: mientras no lo tocas, sus dos patitas están separadas; cuando lo presionas con el dedo, sus dos patitas se tocan.

En este curso usamos el modo inteligente interno del Arduino llamado `INPUT_PULLUP`:

| Estado de tu Dedo | Qué pasa en el Pin 2 del Arduino | Qué devuelve `digitalRead(2)` |
|---|---|---|
| **Botón SUELTO** (sin tocar) | La resistencia interna del Arduino mantiene el pin levantado a 5V. | `HIGH` |
| **Botón PRESIONADO** (dedo apretando) | El botón une el Pin 2 directamente con Tierra (`GND` = 0V). | `LOW` |

Por eso en el código siempre preguntamos: **"¿El botón está en `LOW`?"** para saber si alguien lo está presionando.

---

## 3. Cómo Toma Decisiones el Arduino: `if` (Si...) y `else` (Si no...)

Para que un proyecto sea inteligente debe poder decidir qué hacer según lo que ocurra afuera. En programación usamos la estructura `if / else`:

```
if (estadoBoton == LOW) {
  // Lo que está aquí adentro se ejecuta SOLO SI el botón está presionado
  digitalWrite(ledVerde, HIGH);
} else {
  // Lo que está aquí adentro se ejecuta SI NO está presionado (botón suelto)
  digitalWrite(ledVerde, LOW);
}
```

> [!IMPORTANT]

> **Detalle importante (`=` vs `==`):**

> * Un solo signo igual (`=`) sirve para **guardar** un valor: `int ledVerde = 8;`

> * Dos signos iguales juntos (`==`) sirven para **preguntar / comparar**: `if (estadoBoton == LOW)`

---

## 4. Cómo Funciona el Sonido con `tone()` en el Buzzer

Para emitir sonidos por el Buzzer conectado al Pin `11`, usamos la orden **`tone(pin, nota, duracion)`**:

```
tone(11, 1000, 250);
```

- **`11`:** Es el número del Pin Digital donde está conectado el positivo (`+`) del Buzzer.
- **`1000`:** Es el tono del sonido en Hercios (`Hz`). Un número como `500` suena más grave, y un número como `1500` o `2000` suena más agudo.
- **`250`:** Es cuántos milisegundos durará el pitido antes de apagarse solo (`250 ms` = un cuarto de segundo).