

Muestra de cómo se ve un microproyecto que **lleva de la mano** en los cuatro hilos (estándar doc 24).  
Compárala con la versión actual de una línea por paso.

**Lo que decía antes (1 línea):** «Conecta cada LED a un pin digital (13, 12, 11) con su resistencia.» — correcto, pero un niño de 3° no sabe *cómo*, ni *por qué*, ni *qué hacer si falla*. Abajo, el mismo paso, enseñado.

### 1 ¿Para qué hacemos esto?

En tu ciudad, el semáforo ordena el cruce y protege al peatón. Hoy das el **primer ladrillo**: lograr que tus tres luces —roja, amarilla y verde— enciendan desde el Arduino. Todavía no cambiarán solas (eso es el MP2): primero hay que saber **encenderlas y mandarlas desde el código**. Sin esto, no hay semáforo.

### 2 El concepto, en el momento que lo necesitas

**Predice primero:** conectas un LED a una pila, pero al revés (cambias sus dos patas).  
¿Crees que encenderá igual? Escribe tu respuesta antes de seguir.

**LED.** Un **LED** es una lucecita que **solo enciende en un sentido**. Tiene una **pata larga** (+, se llama ánodo) y una **pata corta** (–, cátodo). Es como una llave de agua que solo abre para un lado.

**Pin digital.** Un **pin digital** del Arduino es una salida con **dos estados**: encendido (**HIGH**) o apagado (**LOW**). Como el interruptor de la pared: sí o no. Por eso un pin digital sirve justo para **prender o apagar** un LED.

**Respuesta a la predicción:** al revés **no enciende** — pero no se daña. Si un LED no prende, lo primero es girarlo.

**¿Y qué tiene que ver con el ODS 11?** Las tres luces son el **idioma** con que la ciudad ordena el cruce: sin ese orden, hay choques. Estás construyendo orden.

### 3 Diseña antes de construir

**Boceto del poste.** Dibuja el poste con la luz **roja arriba**, la **amarilla en medio** y la **verde abajo**.

**Criterio de diseño (¿por qué ese orden?).** Es una norma en todo el mundo: el rojo va arriba para **verse mejor** (aunque haya sol o nieve) y para que quien **no distingue colores** sepa por la **posición** qué significa. El diseño no es adorno: cuida a las personas.

**Esquemático.** Dibuja el circuito con símbolos: el LED es un **triángulo con una barra**, la resistencia es un **zig-zag**. Traza: pin 13 → resistencia → LED rojo → GND, y repite para amarillo (12) y verde (11).

[INSERTAR FIGURA: boceto del poste + esquemático con los 3 LEDs, sus resistencias y la tierra común GND]

#### 4 Construye, paso a paso

1 **Recorta y dobla el poste S-1** por las líneas punteadas.

**Verifica:** ¿el poste se para solo sobre la mesa?

2 **Inserta los tres LEDs** en los círculos marcados. La **pata larga** debe quedar hacia el lado con el signo +.

**Error común:** si pones un LED al revés, **no encenderá** (no se daña). Solo gíralo.

**Verifica:** las tres patas largas miran al mismo lado.

3 **Pon una resistencia (220 Ω)** en la pata corta de cada LED.

**¿Por qué?** El LED aguanta **poca** corriente. La resistencia la **frena** para que no se quemé.

**Error común:** sin resistencia, el LED brilla fuerte un segundo y **se quema**.

4 **Conecta las patas largas a los pines:** rojo → 13, amarillo → 12, verde → 11. Las resistencias (patas cortas) van todas a **GND** (la tierra común).

**Verifica:** cuenta **3 cables** yendo a pines y **3** yendo a GND.

## 5 Programa: primero el plan, luego el código

**Algoritmo en palabras (esto se piensa antes de teclear):**

1. Al iniciar, prepara los pines 13, 12 y 11 como **salidas** (van a mandar).
2. Enciende los tres.

**Predice y corre:** mira el programa de abajo. ¿Qué crees que hará *antes* de subirlo al Arduino? Luego súbelo y comprueba.

```
void setup() {  
  pinMode(13, OUTPUT); // el pin 13 va a MANDAR corriente (salida)  
  pinMode(12, OUTPUT);  
  pinMode(11, OUTPUT);  
  digitalWrite(13, HIGH); // HIGH = manda corriente = enciende el ROJO  
  digitalWrite(12, HIGH); // enciende el AMARILLO  
  digitalWrite(11, HIGH); // enciende el VERDE  
}  
void loop() { // por ahora no repite nada  
}
```

En bloques (mBlock) es lo mismo: «al iniciar» → «fijar pin 13 en encendido» (y 12, 11).

**Explica lo nuevo:** pinMode(pin, OUTPUT) le dice al pin si **escucha** o **manda**; OUTPUT = manda. digitalWrite(pin, HIGH) manda corriente (enciende); LOW la corta (apaga).

**Traza (hazte el computador):** después de correr, ¿en qué estado queda cada pin?

Pin	Color	Estado	¿Luz?
13	Rojo	HIGH	encendida
12	Amarillo	HIGH	encendida
11	Verde	HIGH	encendida

**Depura como detective:** si el rojo y el amarillo encienden pero el **verde no**, fíjate: el código es **igual** para los tres. Entonces el problema casi seguro **no es el código**, sino el LED verde o su cable. Revisa su **dirección** y su **tierra**. (Aislar comparando: lo que funciona te dice dónde NO está el error.)

## 6 Integra y prueba

Sube el código con el circuito conectado.

**¿Funcionó?** Los tres LEDs deben quedar encendidos a la vez. **¿Ya es un semáforo?** Todavía no: un semáforo **cambia** y **espera**. Eso lo programarás en el **MP2 (el ciclo)**. Hoy lograste lo esencial: **mandar las luces desde el código**.

[INSERTAR FIGURA: foto de los tres LEDs encendidos en el poste de papel]

## 7 Cómo se unieron los hilos · IA · reflexión

**Convergencia:** hoy se juntaron **diseño** (el esquema y el orden de las luces), **construcción** (el circuito con sus resistencias) y **programación** (encender desde los pines). Ninguno solo logra el semáforo.

**IA (opcional):** pídele a la IA «explícame qué hace una resistencia como si tuviera 8 años» y compáralo con lo que entendiste. ¿Coincide? ¿Qué agregarías?

**Reflexiona:** ¿por qué crees que el rojo va **arriba** y no abajo?

## 8 Lo lograste · comprueba lo que sabes

★ **Electricista**

**Hito: «Enciendo y controlo 3 LEDs desde el código.»**

Ganas la insignia **Electricista** cuando los tres encienden desde tu programa.

**Preguntas (para pensar, no para memorizar):**

1. ¿Qué pasaría si quitas la resistencia de un LED? ¿Por qué?
2. ¿Por qué un LED tiene dirección y un foco común de la casa no?
3. Cambia el código para que el **verde** quede apagado: ¿qué línea tocas y cómo?