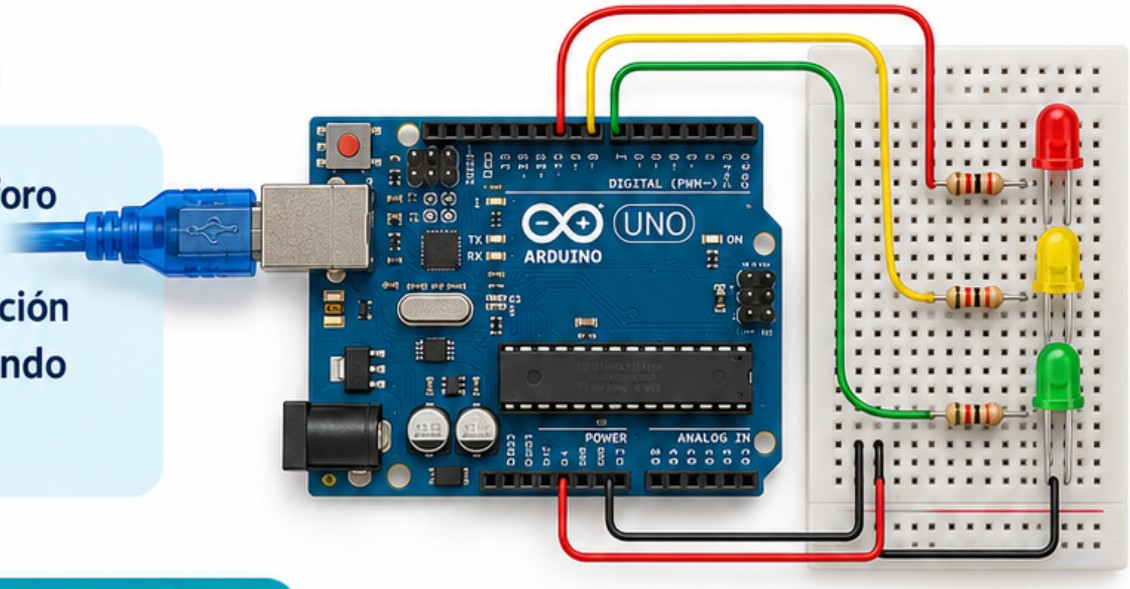
ROBÓTICA CON ARDUINO

Nuestro primer semáforo

Propósito

Construir un semáforo con tres LED y programar la duración de cada luz utilizando Arduino.

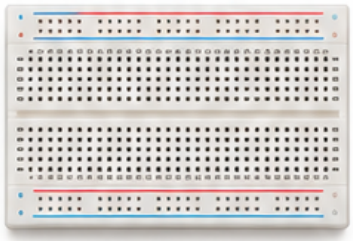


Materiales por equipo

1 Arduino Uno o Nano



1 protoboard



1 LED rojo



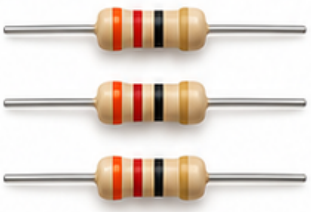
1 LED amarillo



1 LED verde



3 resistencias de 220 Ω o 330 Ω



7 cables Dupont aproximadamente



1 cable USB



Computadora con Arduino IDE



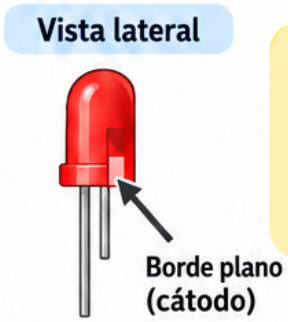
Nombre: _____ Grupo: _____ Fecha: _____

PÁGINA 2 · ANTES DE CONECTAR

Conoce los componentes y cómo funcionan antes de hacer tus conexiones.

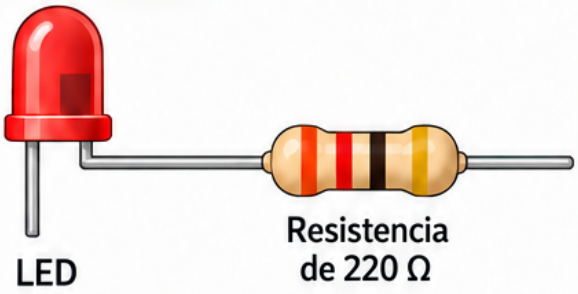
1 POLARIDAD DEL LED





El LED solo permite el paso de corriente en una dirección.

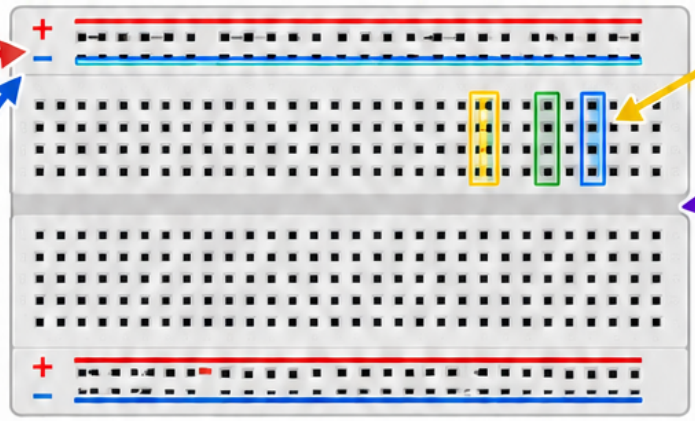
2 CADA LED NECESITA UNA RESISTENCIA



La resistencia limita la corriente y protege al LED.

⚠ **Nunca conectes un LED directamente a un pin o a 5 V.**

3 ¿CÓMO SE CONECTA EL PROTOBOARD?



Riel lateral rojo: alimentación (+)

Riel lateral azul: GND / negativo (–)

Grupos de cinco orificios conectados internamente.

Ranura central que separa ambos lados.

COMPRUEBA ANTES DE AVANZAR

- ☐ Identifico el ánodo y el cátodo.
- ☐ Sé para qué sirve la resistencia.
- ☐ Reconozco los grupos de cinco orificios.

¡Buen trabajo! Revisa que puedas explicarlo con tus propias palabras.

✂ RECORTA Y PEGA EN TU LIBRETA

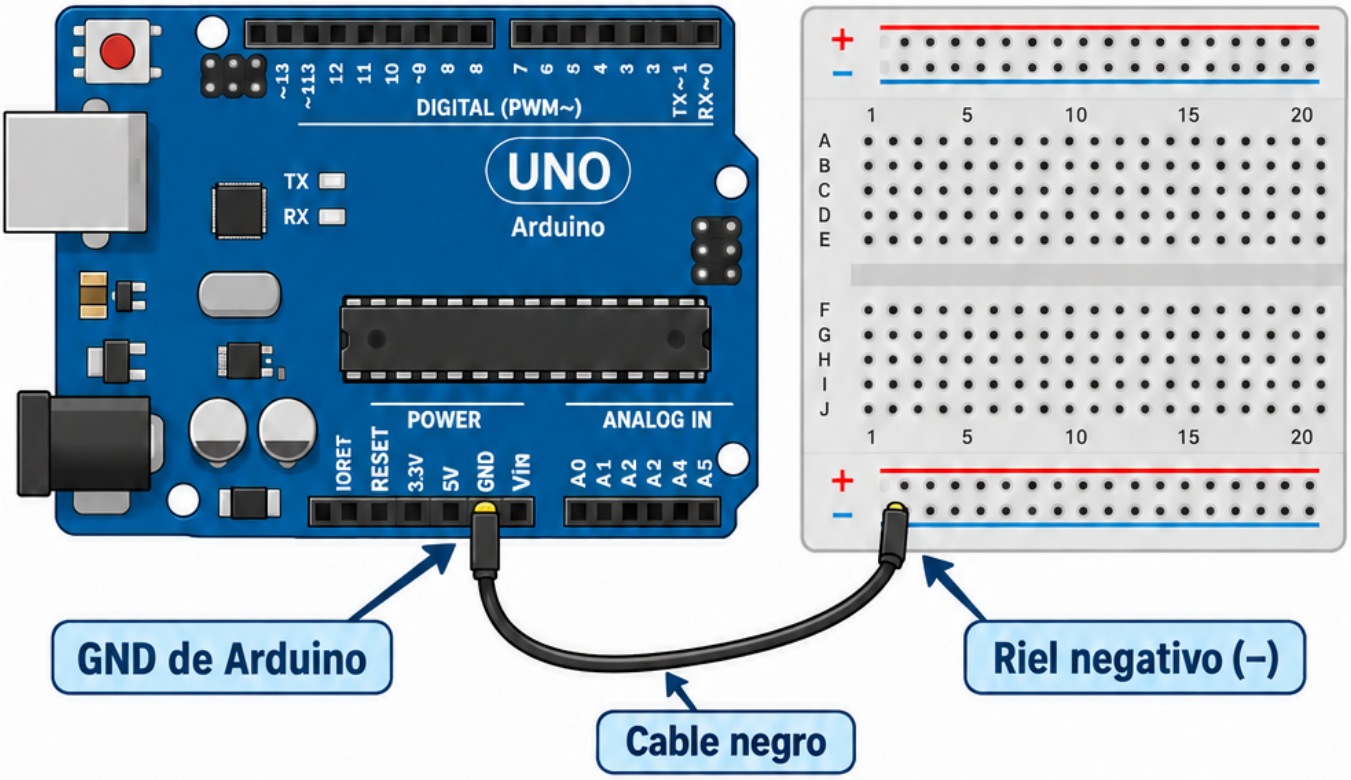
PÁGINA 3 • PREPARAMOS LA TIERRA (GND)

Antes de colocar los LED, todos compartirán un camino de regreso.

1 PASO 1 • DESCONECTA EL USB



Realiza las conexiones con Arduino sin energía.



¿QUÉ SIGNIFICA GND?

GND significa tierra. Es el punto de referencia de **0 V** y proporciona el camino de regreso de la corriente.

ANTES DE CONTINUAR

- ☐ El cable sale de un pin **GND**.
- ☐ El cable llega al riel azul (–).
- ☐ El Arduino continúa desconectado del USB.

GND DE ARDUINO → **CABLE NEGRO** → **RIEL NEGATIVO DEL PROTOBOARD**

No conectamos 5 V al riel rojo porque cada LED será controlado desde un pin digital.

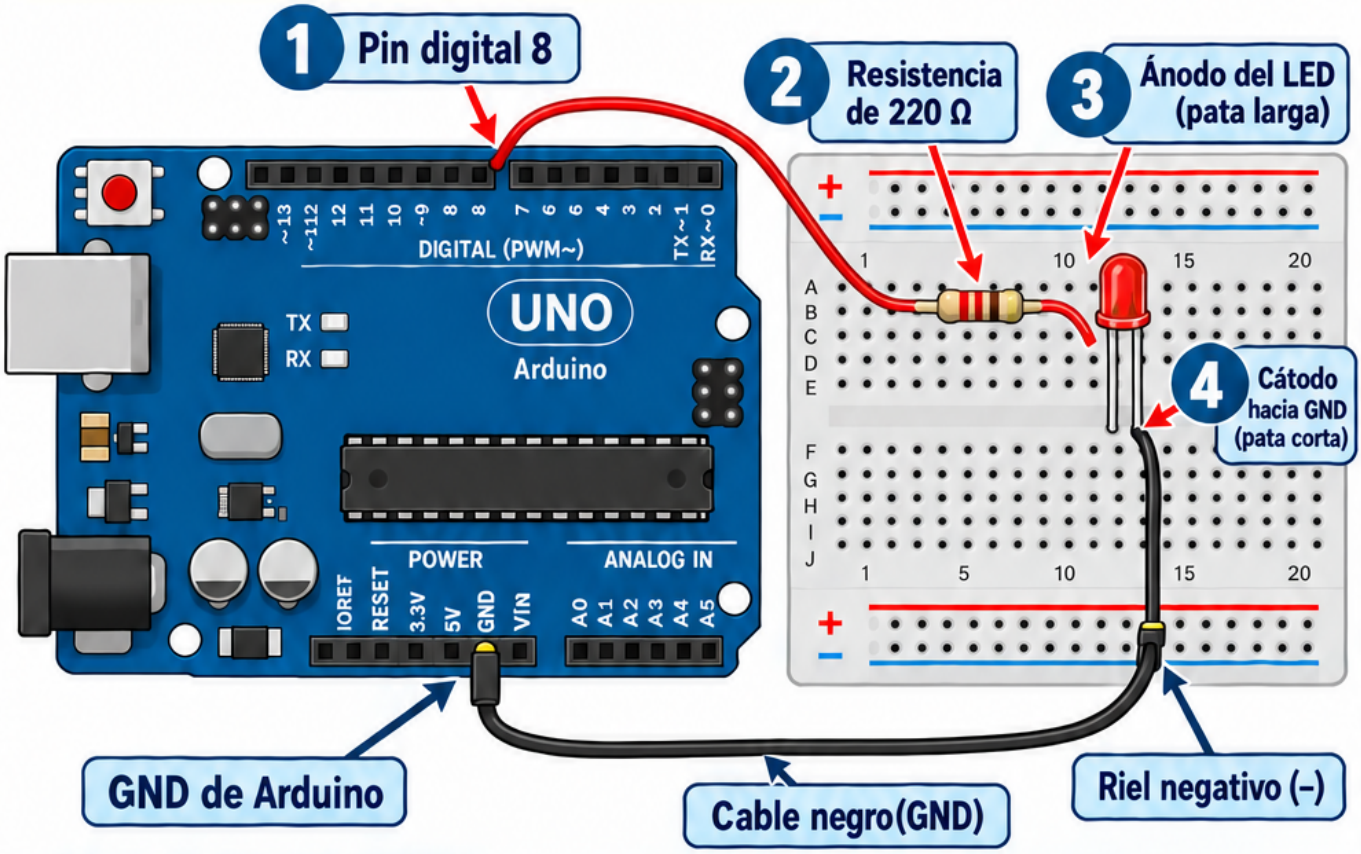
RECORTA Y PEGA EN TU LIBRETA

PÁGINA 4 • CONECTAMOS EL LED ROJO

Construiremos el circuito una conexión a la vez.

ANTES DE EMPEZAR

Mantén el Arduino desconectado del USB.



RUTA DEL CIRCUITO

PIN 8 → RESISTENCIA 220 Ω → ÁNODO DEL LED → CÁTODO → GND

IMPORTANTE

La resistencia puede colocarse antes o después del LED, siempre que permanezca **en serie**.

REVISA TU MONTAJE

- [ ] El Arduino está desconectado.
- [ ] Las patas del LED están en nodos diferentes.
- [ ] El ánodo se conecta al pin 8 mediante la resistencia.
- [ ] El cátodo llega al riel GND.

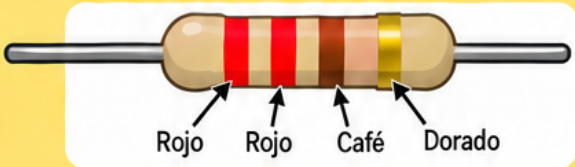
REGISTRO EN MI LIBRETA

Pin utilizado: _____

Color del LED: _____

Valor de la resistencia: _____

CÓDIGO DE COLORES DE LA RESISTENCIA (220 Ω)

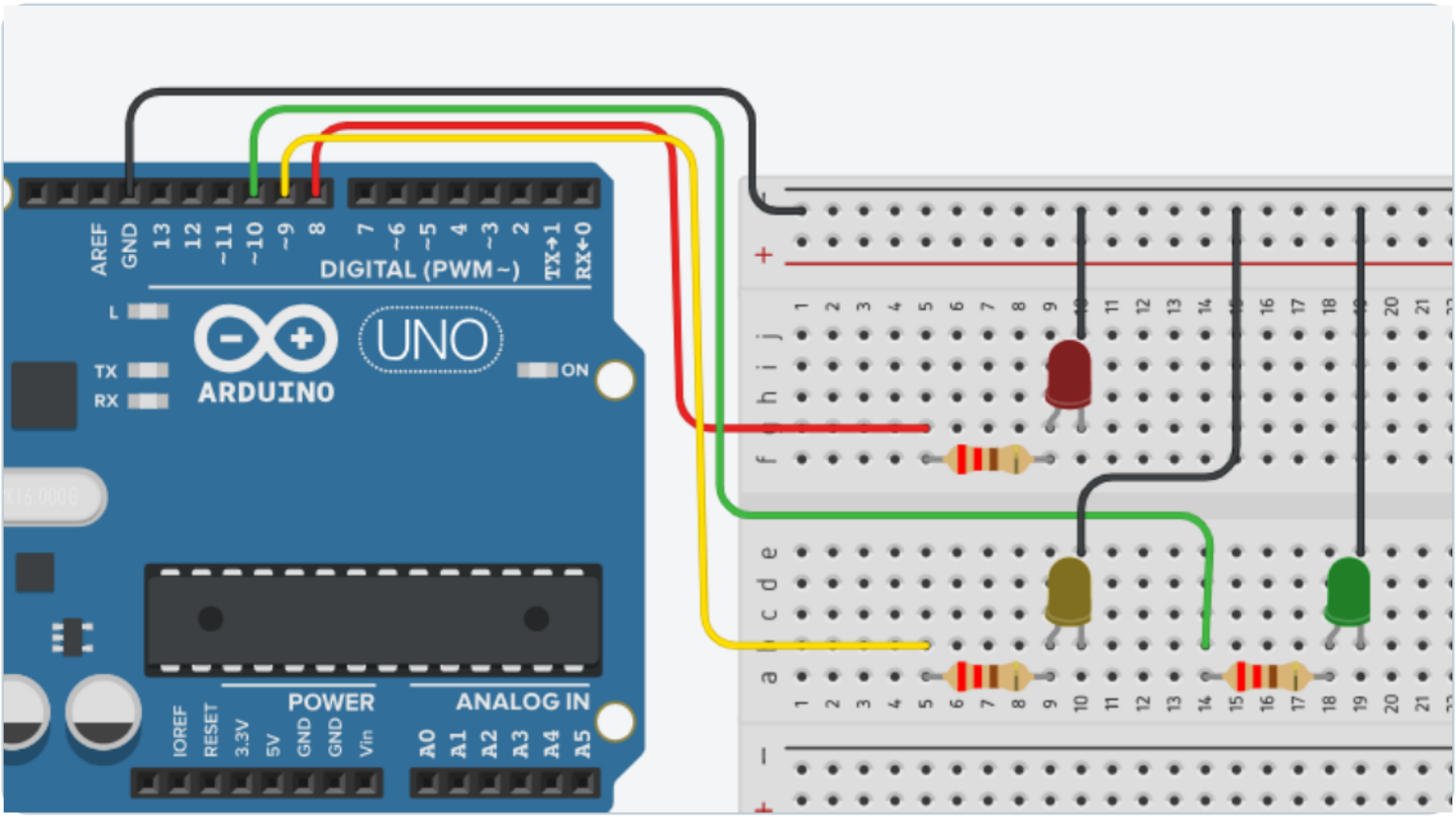


RECORTA Y PEGA EN TU LIBRETA

PÁGINA 5 · COMPLETAMOS EL SEMÁFORO

Cada luz tendrá su propio pin y su propia resistencia.

⚠ Realiza las conexiones con el Arduino desconectado del USB.



VERDE → PIN 10 · AMARILLO → PIN 9 · ROJO → PIN 8 · NEGRO → GND

MAPA DE CONEXIONES

LUZ	PIN	RESISTENCIA
Roja	8	220 Ω
Amarilla	9	220 Ω
Verde	10	220 Ω

REVISA TU MONTAJE

- ☐ Los LED respetan la polaridad.
- ☐ Cada LED tiene una resistencia.
- ☐ Los cables salen de 8, 9 y 10.
- ☐ Los tres cátodos llegan a GND.

✂ RECORTA Y PEGA EN TU LIBRETA

PÁGINA 6 · ESTRUCTURA DE UN PROGRAMA

Un programa de Arduino se llama sketch y tiene dos partes principales.

```
void  setup() {
  // Se ejecuta una sola vez
}

void  loop() {
  // Se repite continuamente
}
```

1 setup()

Se ejecuta **una sola vez** cuando Arduino se enciende o se reinicia.

Aquí prepararemos los pines con:

```
pinMode(pin, OUTPUT);
```

2 loop()

Se repite **continuamente** mientras Arduino tenga energía.

Aquí irá la secuencia del semáforo:



COMPRUEBA LO APRENDIDO

Escribe **“una vez”** o **“continuamente”**.

`setup()` se ejecuta: _____

`loop()` se ejecuta: _____

IDEA CLAVE: setup prepara · loop repite

✂ **RECORTA Y PEGA EN TU LIBRETA**

PÁGINA 7 · CONSTANTES Y PINES

Daremos un nombre fácil de recordar a cada pin del semáforo.

```
const int ledRojo = 8;

const int ledAmarillo = 9;

const int ledVerde = 10;
```

Ahora podemos escribir ledRojo en vez de recordar siempre el número 8.

¿QUÉ SIGNIFICA CADA PARTE?

`const`	El valor no cambiará durante el programa.
`int`	Guarda un número entero.
`ledRojo`	Es el nombre que asignamos al pin 8.

CONFIGURAMOS LOS PINES EN setup()

```
void setup() {
  pinMode(ledRojo,OUTPUT);
  pinMode(ledAmarillo,OUTPUT);
  pinMode(ledVerde,OUTPUT);
}
```

pinMode() configura el uso del pin · OUTPUT significa que enviará una señal.

COMPLETA SEGÚN EL CIRCUITO

`ledRojo = ____;` `ledAmarillo = ____;` `ledVerde = ____;`

Pista: consulta el mapa de conexiones de la página anterior.

✂ **RECORTA Y PEGA EN TU LIBRETA**

PÁGINA 8 · ENCENDER, APAGAR Y ESPERAR

Arduino controla cada LED mediante instrucciones digitales y tiempos.

```
digitalWrite(ledVerde, HIGH);
```

Coloca el pin en estado alto y **enciende** el LED verde.

HIGH = ENCENDIDO

```
digitalWrite(ledVerde, LOW);
```

Coloca el pin en estado bajo y **apaga** el LED verde.

LOW = APAGADO

```
delay(3000);
```

Detiene temporalmente el programa durante **3000 ms**.

1000 milisegundos = 1 segundo · 3000 ms = 3 segundos

LEE LA SECUENCIA

```
digitalWrite(ledRojo, HIGH);
delay(5000);
```

El LED rojo se _____

durante _____ segundos.

CONVIERTE LOS TIEMPOS

2 segundos = _____ ms

7 segundos = _____ ms

4000 ms = _____ s

1000 ms = _____ s

✂ **RECORTA Y PEGA EN TU LIBRETA**

PÁGINA 9 · CÓDIGO COMPLETO DEL SEMÁFORO

Lee el código de arriba hacia abajo y después cárgalo en Arduino.

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

void setup() {
   pinMode(ledRojo, OUTPUT);
   pinMode(ledAmarillo, OUTPUT);
   pinMode(ledVerde, OUTPUT);
}

void loop() {
   // VERDE · avanzar
   digitalWrite(ledRojo, LOW);
   digitalWrite(ledAmarillo, LOW);
   digitalWrite(ledVerde, HIGH);
   delay(5000);

   // AMARILLO · precaución
   digitalWrite(ledVerde, LOW);
   digitalWrite(ledAmarillo, HIGH);

   delay(2000);

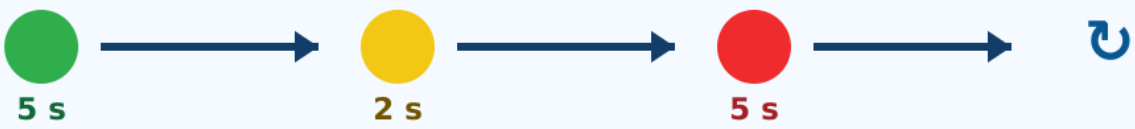
   // ROJO · detenerse

   digitalWrite(ledAmarillo, LOW);
   digitalWrite(ledRojo, HIGH);
   delay(5000);

}
```

Al terminar el rojo, loop() vuelve al principio y la secuencia se repite.

SECUENCIA DEL SEMÁFORO



✂ **RECORTA Y PEGA EN TU LIBRETA**

PÁGINA 10 · COMPLETA EL PROGRAMA

Utiliza lo aprendido para completar los espacios sin consultar la página 9.

PROGRAMA RELLENABLE

```
const int ledRojo = ____;
const int ledAmarillo = ____;
const int ledVerde = ____;

void setup() {
   pinMode(ledRojo, _________);
   pinMode(ledAmarillo, _________);
   pinMode(ledVerde, _________);
}

void loop() {
   // VERDE
   digitalWrite(ledRojo, _______);
   digitalWrite(ledAmarillo, _______);
   digitalWrite(ledVerde, _______);
   delay(_________);

   // AMARILLO
   digitalWrite(ledVerde, _______);
   digitalWrite(ledAmarillo, _______);

   delay(_________);

   // ROJO

   digitalWrite(ledAmarillo, _______);
   digitalWrite(ledRojo, _______);
   delay(_________);
}
```

BANCO DE PALABRAS

OUTPUT	HIGH
LOW	2000
5000	8, 9, 10

TIEMPOS DEL RETO

- Verde: 5 s
- Amarillo: 2 s
- Rojo: 5 s

Recuerda: 1 segundo = 1000 ms

DEMUESTRA QUE SABES

1. ¿Qué función se ejecuta una sola vez?

   _____

2. ¿Qué función se repite?

   _____

3. ¿Qué significa HIGH?

   _____

4. ¿Qué significa LOW?

   _____

5. ¿Por qué usamos resistencias?

PRUEBA FINAL: Carga el programa, observa la secuencia y modifica un tiempo.

Cambio realizado: _____

✂ **RECORTA Y PEGA EN TU LIBRETA**