

## Gracias por elegir el Beginner's Lab Kit

Bienvenido al Kit de Laboratorio para Principiantes, un paquete inicial completo diseñado específicamente para quienes se inician en el mundo de la electrónica y la programación. Este kit incluye una variedad de componentes esenciales como LEDs, resistencias, un zumbador, potenciómetros, fotorresistencias, termistores, pulsadores, tubos digitales y un módulo ultrasónico. Una de las características más destacadas de este kit es la inclusión de un multímetro, una herramienta invaluable que le permite medir la corriente, el voltaje y la resistencia dentro de sus circuitos. Esta adición es particularmente útil para profundizar su comprensión de cómo funciona cada componente.

La secuencia de cursos proporcionada con este kit está estructurada en torno a la sintaxis de programación de Arduino, asegurando una progresión lógica y educativa. Esta estructura le permite construir circuitos paso a paso mientras aprende a escribir los programas que los controlan. A lo largo del curso, se encontrará con desafíos de resolución de problemas que aumentarán su comprensión del material.

Para cualquier consulta o soporte, no dude en contactarnos en [service@sunfounder.com](mailto:service@sunfounder.com). ¡Sumérjase en su viaje de aprendizaje con el Kit de Laboratorio para Principiantes y comience a construir, programar y explorar el fascinante mundo de la electrónica!

# Contenido

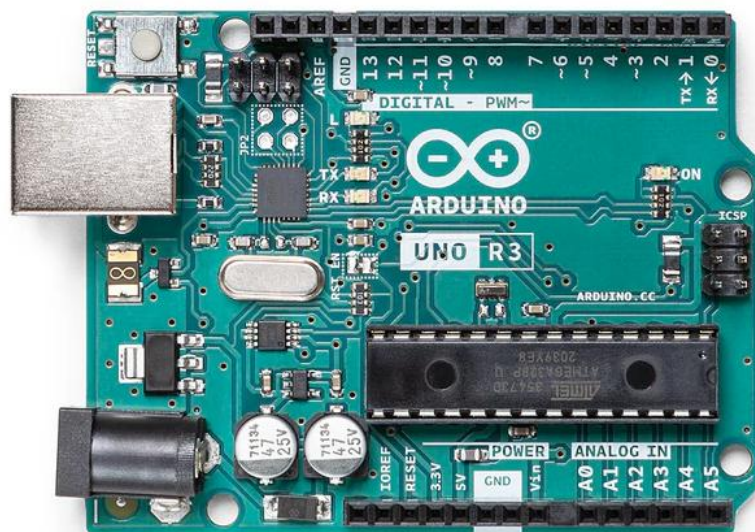
Qué incluye su kit .....	1
Lección 2: Tu primer circuito .....	8
Lección 3: Medir con el multímetro .....	9
Lección 4: Ley de Ohm .....	10
Lección 5: Circuito en serie vs. Circuito en paralelo .....	11
Lección 6: Parpadeo del LED .....	15
Lección 7: ¡Hagamos un semáforo! .....	16
Lección 8: Semáforo con botón para peatones .....	17
Lección 9: Lámpara de escritorio regulable .....	19
Lección 10: Lámpara de escritorio ON/OFF .....	20
Lección 11: Control de matrices de LED con potenciómetro .....	21
Lección 12: Los Colores del Arcoíris .....	22
Lección 13: El Espectro de la Vista .....	23
Lección 14: Colores Aleatorios .....	24
Lección 15: Colores Fríos o Cálidos .....	25
Lección 16: Alarma de Temperatura .....	26
Lección 17: Código Morse .....	27
Lección 18: Alarma de Luz .....	28
Lección 19: Sistema de Alarma para Estacionamiento en Reversa .....	29
Lección 20: El Temporizador Pomodoro .....	30
Lección 21: Sonido de Sirena .....	32
Lección 22: Toca "Estrellita, ¿dónde estás?" .....	34
Lección 23: Dados Cibernéticos .....	35
Lección 24: Luz Fluorescente con 74HC595 .....	37
Lección 25: Mostrar Número .....	38

## Qué incluye su kit

Dentro de nuestro kit, encontrará una variedad de componentes y piezas que utilizará a lo largo de este curso para construir circuitos. A continuación, se presenta una guía rápida de lo que se incluye.

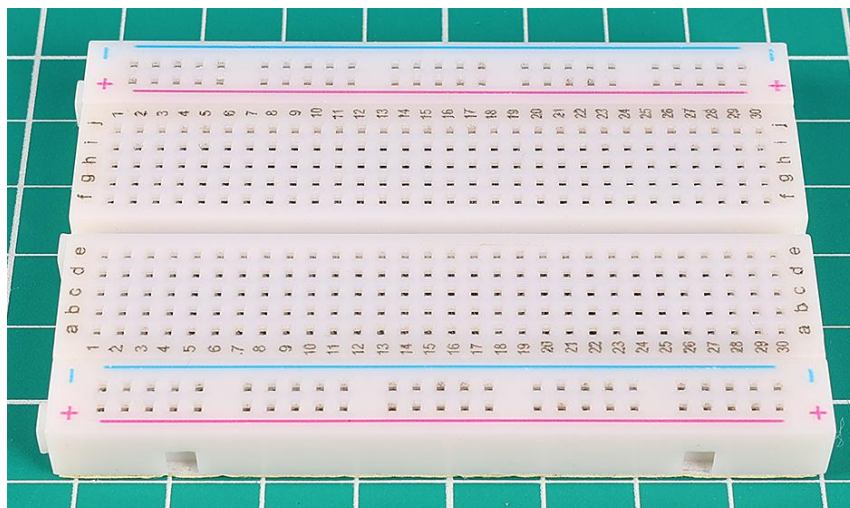
### 1 x Original Arduino Uno R3

Una placa con microcontrolador que actúa como el cerebro de sus circuitos. Tiene todo lo necesario para soportar el microcontrolador; simplemente conéctela a su computadora con un cable USB o aliméntela con un adaptador de CA a CC o una batería para comenzar.



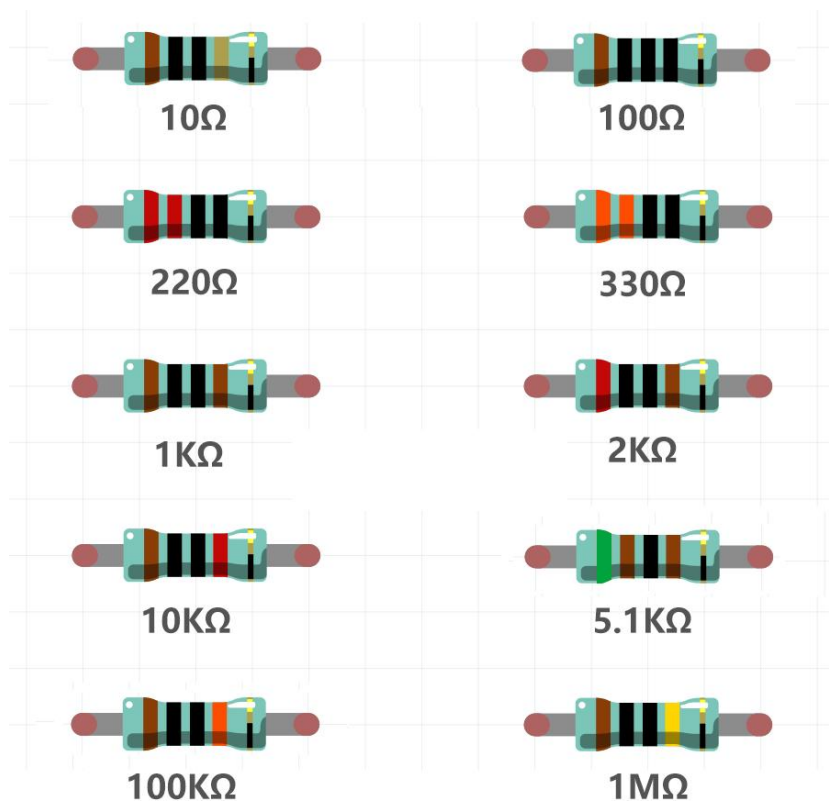
### 1 x 400-hole Breadboard

Una placa sin soldadura que le permite construir circuitos electrónicos fácilmente. Está compuesta por filas de orificios para conectar cables y componentes.



## 120 x Resistencias (10 de cada tipo, 30 de resistencias de 220Ω)

Una resistencia es un componente que impide el flujo de energía eléctrica, modificando así la tensión y la corriente en un circuito. El valor de una resistencia se mide en ohmios, simbolizados por la letra griega omega ( $\Omega$ ). Las franjas de colores en una resistencia indican su valor de resistencia y su tolerancia.



## 25 x LEDs (5 de cada color)

Esta selección de LED coloridos incluye cinco colores: rojo, verde, azul, amarillo y blanco, satisfaciendo diversas necesidades de iluminación y señalización. Adecuados para aplicaciones que van desde simples indicadores de estado hasta proyectos de iluminación decorativa complejos, estos LED ofrecen una amplia gama de colores para mejorar el atractivo visual de cualquier proyecto electrónico.



## 2 x RGB LEDs

Combina LEDs rojos, verdes y azules en una sola carcasa. Puede mostrar varios colores ajustando el voltaje de entrada, creando millones de combinaciones de colores.



## 1 x Fotorresistor

Un fotorresistor es un componente sensible a la luz que varía su resistencia según la intensidad de la luz a la que está expuesto, ideal para crear controles y sensores activados por luz en proyectos electrónicos.



## 1 x NTC Thermistor

Un termistor es una resistencia sensible a los cambios de temperatura. Los termistores NTC disminuyen su resistencia a medida que aumenta la temperatura, mientras que los termistores PTC aumentan su resistencia con el incremento de la temperatura.



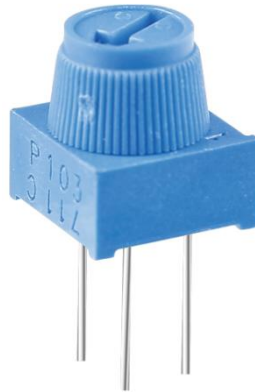
## 1 x Zumbador activo & 1 x Zumbador pasivo

Un zumbador, disponible en versiones activas y pasivas, es un dispositivo de señalización acústica que emite sonido cuando se le aplica corriente eléctrica. Se utiliza comúnmente en alarmas, temporizadores y sistemas de notificación.



## 1 x Potenciómetro

Un potenciómetro es una resistencia variable con tres pines. Dos de los pines se conectan a los extremos de la resistencia, mientras que el pin central se conecta a un cursor móvil que divide la resistencia en dos partes. Los potenciómetros, que se utilizan con frecuencia para ajustar el voltaje en circuitos, son similares a las perillas de volumen en radios.



## 10 x Botones pequeños

Un pulsador pequeño se utiliza para proporcionar una respuesta física al ser presionado y se usa comúnmente en dispositivos electrónicos para iniciar acciones o introducir comandos.





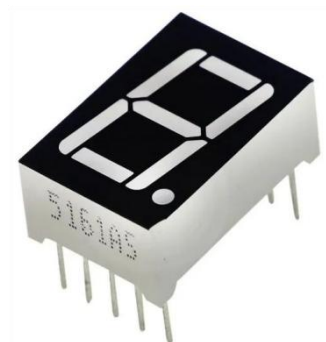
## 1 x 74HC595 Chip

El 74HC595 es un registro de desplazamiento que se utiliza para expandir los puertos de entrada/salida de los circuitos digitales al convertir la entrada serie en salida paralela, reduciendo así el número de pines de conexión necesarios. Este chip es ideal para controlar un gran número de dispositivos de salida, como pantallas de 7 segmentos, sin ocupar demasiados pines del microcontrolador.



## 1 x Pantalla de 7 segmentos

Una pantalla de 7 segmentos es un componente con forma de 8 que contiene 7 LEDs. Cada LED se denomina segmento y, al ser activado, cada segmento forma parte de un número que se desea mostrar.



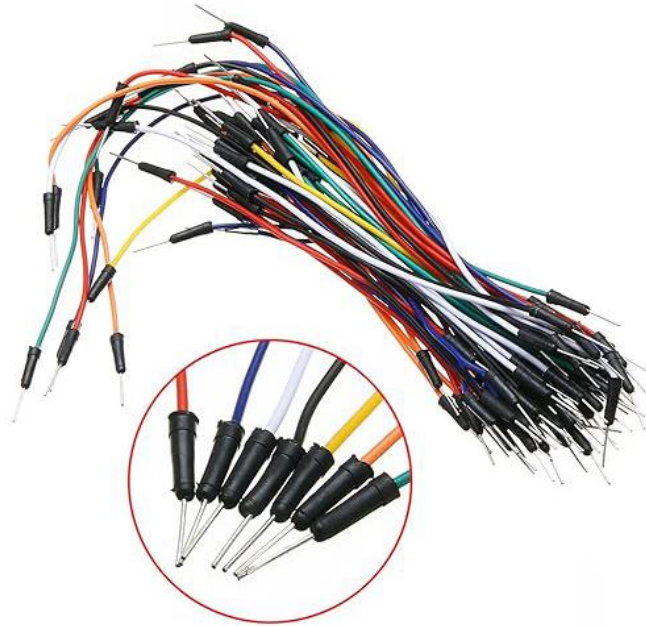
## 1 x Módulo ultrasónico

Este es un módulo ultrasónico que utiliza ondas ultrasónicas para medir distancias, detectando y midiendo con precisión la posición y la distancia de los objetos. Se usa ampliamente en robótica, sistemas de evitación de obstáculos y en el campo del control automático.



## 65 x Cables de puente

Conectan los componentes en la placa de pruebas entre sí y con la placa Arduino.



## 10 x Cables DuPont macho-hembra

Los cables DuPont macho-hembra están diseñados específicamente para conectar módulos con pines macho, como el módulo ultrasónico, a la placa de pruebas. Estos cables son esenciales para la interconexión de diferentes componentes en proyectos electrónicos.





## 1 x USB Cable

Conecta la placa Arduino a una computadora. Permite escribir, compilar y transferir programas a la placa Arduino. También alimenta la placa.



## 1 x Batería de 9V

Esta es una batería alcalina de 9V no recargable. Debe instalarla en el multímetro.



## 1 x Multímetro con cables rojo y negro

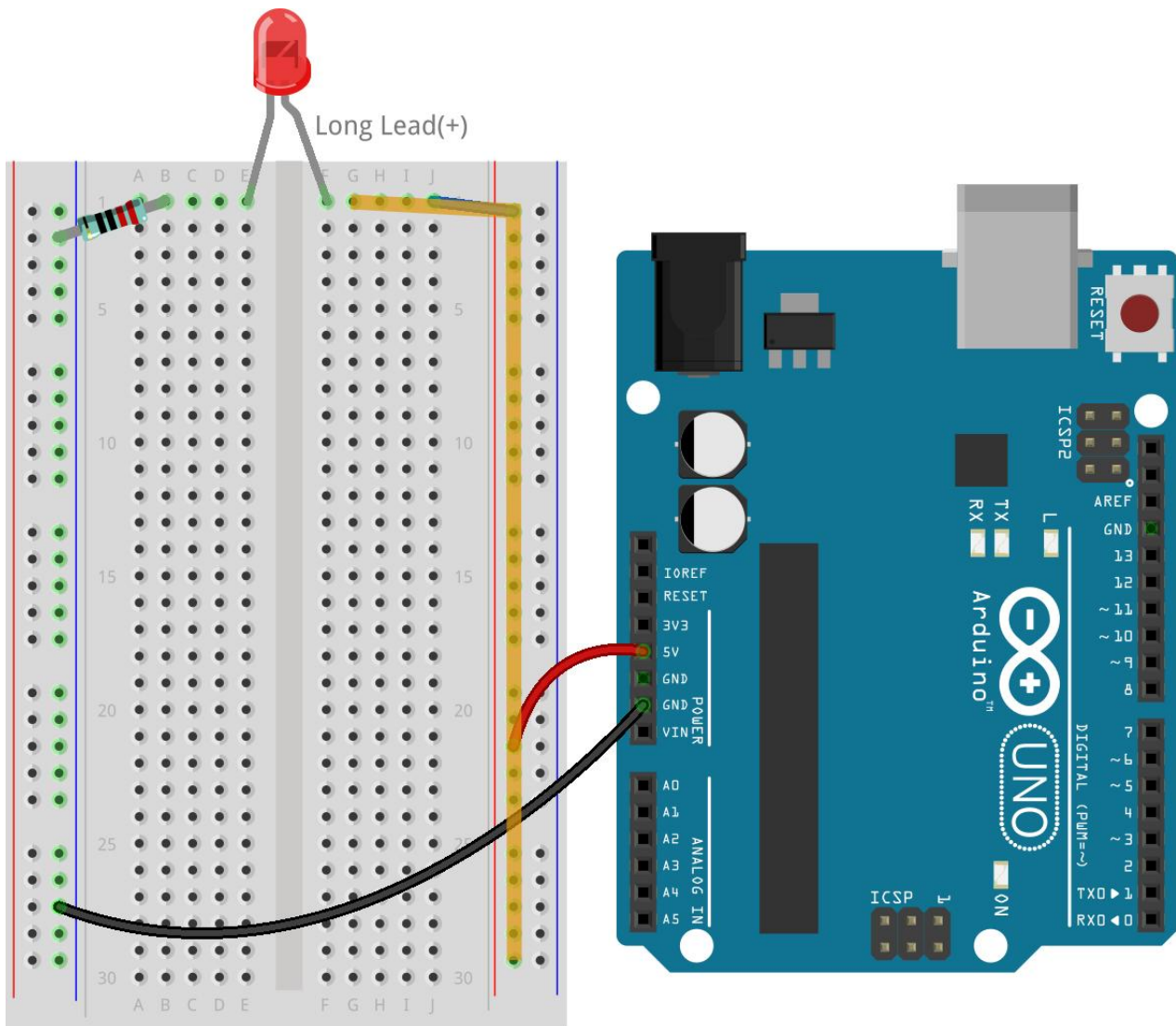
Este es un multímetro versátil capaz de medir voltaje, corriente y resistencia, así como realizar otras pruebas eléctricas, lo que lo convierte en una herramienta indispensable para trabajos de electrónica y electricidad.



## Lección 2: Tu primer circuito

Después de completar la lección, responda las siguientes preguntas.

1. Retire el cable rojo de la placa de pruebas y experimente colocándolo en diferentes orificios de la placa. Observe cualquier cambio en el LED. Dibuje las posiciones de los orificios que permiten que el LED se encienda.



2. ¿Qué sucede si invierte los pines del LED? ¿Se encenderá? ¿Por qué sí o por qué no?

El LED no se enciende porque tiene conductividad unidireccional; la corriente debe fluir del ánodo al cátodo para que funcione.

## Lección 3: Medir con el multímetro

Responda esta pregunta después de completar la sección "Conozca más sobre el multímetro".

1. Ahora que tiene una comprensión detallada de cómo usar un multímetro, considere qué configuración del multímetro usaría para medir los siguientes valores eléctricos.

Objeto de medición	Configuración del multímetro
9V volts DC	<i>20V</i>
1K ohmios	<i>2k<math>\Omega</math></i>
40 miliamperios	<i>200mA</i>
110 volts AC	<i>200V~</i>

Rellene esta tabla durante la sección "Medición con un multímetro"

Tipo	Unidades	Resultados de la medición	Notas
Voltage	Volts	<i><math>\approx 5.13</math> volts</i>	
Current	Milliamps	<i><math>\approx 13.54</math> milliamps</i>	
Resistance	Ohms	<i><math>\approx 378.88</math> ohms</i>	

## Lección 4: Ley de Ohm

Rellene la siguiente tabla durante la sección “ Exploración de la Ley de Ohm con experimentos prácticos”.

1. Sustituya la resistencia de 220 ohmios con otras resistencias de diferentes valores como se indica a continuación. Registre los cambios en el brillo del LED con cada sustitución para observar cómo la resistencia afecta la corriente y, en consecuencia, la intensidad luminosa.

Resistencia	Observations
100Ω	<i>Más brillante</i>
1KΩ	<i>Brillante</i>
10KΩ	<i>Más tenue</i>
1MΩ	<i>Casi apagado</i>

Notará que solo con la resistencia de 100Ω el LED es más brillante que con la resistencia anterior de 220Ω. Con resistencias más altas, el brillo del LED disminuye hasta que se apaga por completo a 1MΩ. ¿Por qué ocurre esto?

Según la Ley de Ohm ( $I = V/R$ ), al aumentar la resistencia mientras se mantiene constante el voltaje, la corriente que pasa por el LED disminuye, lo que hace que el LED se atenúe. A 1MΩ, la corriente es demasiado baja para encender el LED.

1. Después de observar los efectos de cambiar la resistencia, mantenga la resistencia en 220 ohmios y cambie la fuente de voltaje del circuito de 5V a 3.3V. Registre cualquier cambio en el brillo del LED.

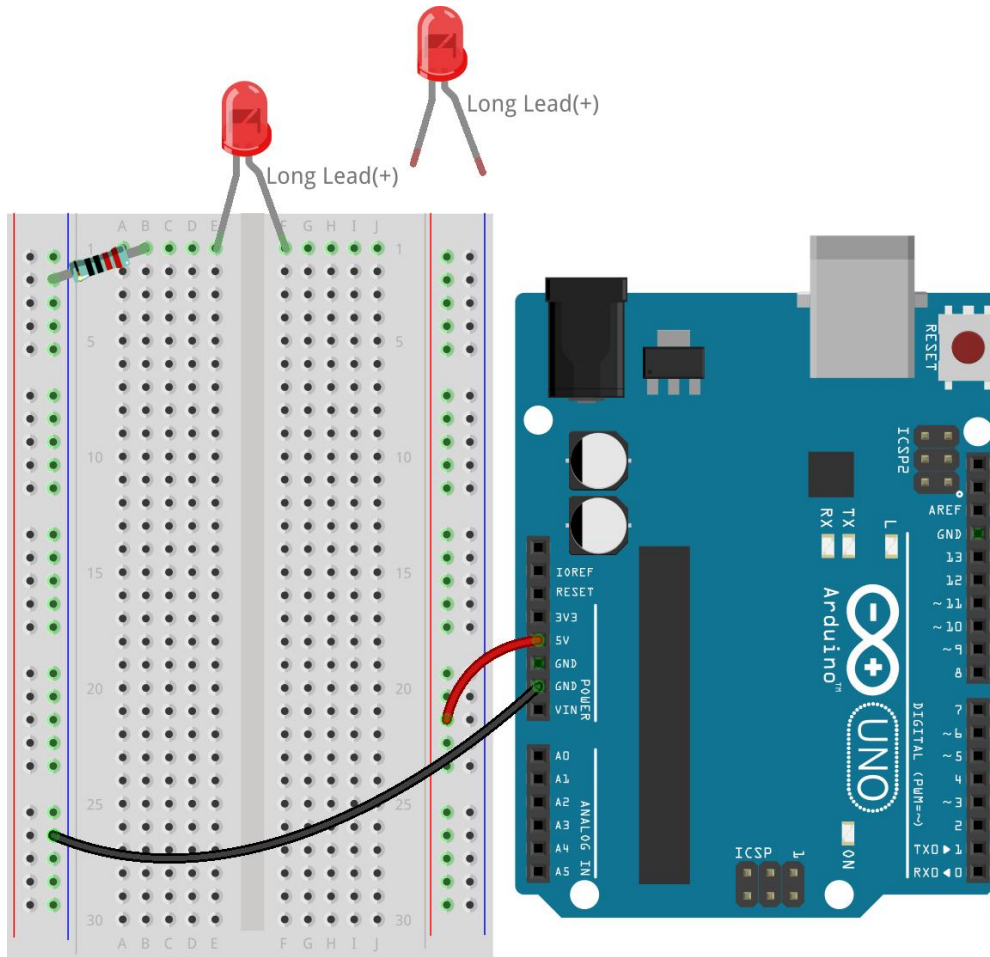
Encontrará que el LED es ligeramente más tenue a 3.3V que a 5V. ¿Por qué ocurre esto?

Según la Ley de Ohm, conociendo la resistencia y el nuevo voltaje, la corriente debe ser  $I = V/R$ . Con una disminución del voltaje mientras la resistencia se mantiene igual, la corriente disminuye, lo que atenúa el LED.

# Lección 5: Circuito en serie vs. Circuito en paralelo

Complete las siguientes preguntas durante la sección “Profundizando en los circuitos en serie”.

1. ¿Qué sucede si retira un LED? ¿Por qué ocurre esto?



En un circuito en serie, si retira un LED, el otro LED no se encenderá. Esto se debe a que, en un circuito en serie, la corriente debe pasar por todos los componentes en el trayecto. Al retirar un LED, se interrumpe el circuito, impidiendo que la corriente fluya a través del LED restante.

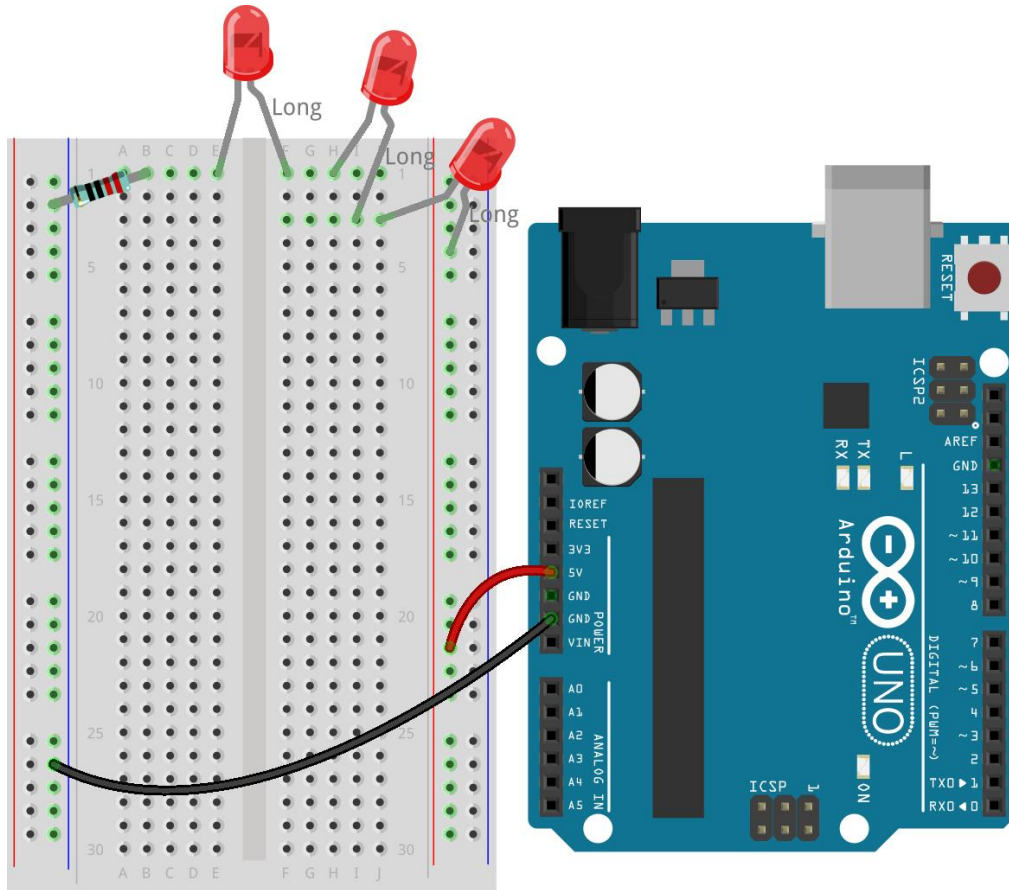
1. Mida el voltaje de cada componente en el circuito en serie.

Circuito	Voltaje de la resistencia	Voltaje del LED1	Voltaje del LED2	Voltaje total
2 LEDs	$\approx 1.13 \text{ volts}$	$\approx 1.92 \text{ volts}$	$\approx 1.92 \text{ volts}$	$\approx 4.97 \text{ volts}$

2. Mida la corriente de cada componente en el circuito en serie.

Circuito	Corriente del LED1	Corriente del LED2
2 LEDs	$\approx 4.43$ milliamps	$\approx 4.43$ milliamps

1. Si se añade otro LED a este circuito, resultando en tres LEDs, ¿cómo cambia el brillo de los LEDs? ¿Por qué? ¿Cómo cambian los voltajes a través de los tres LEDs?



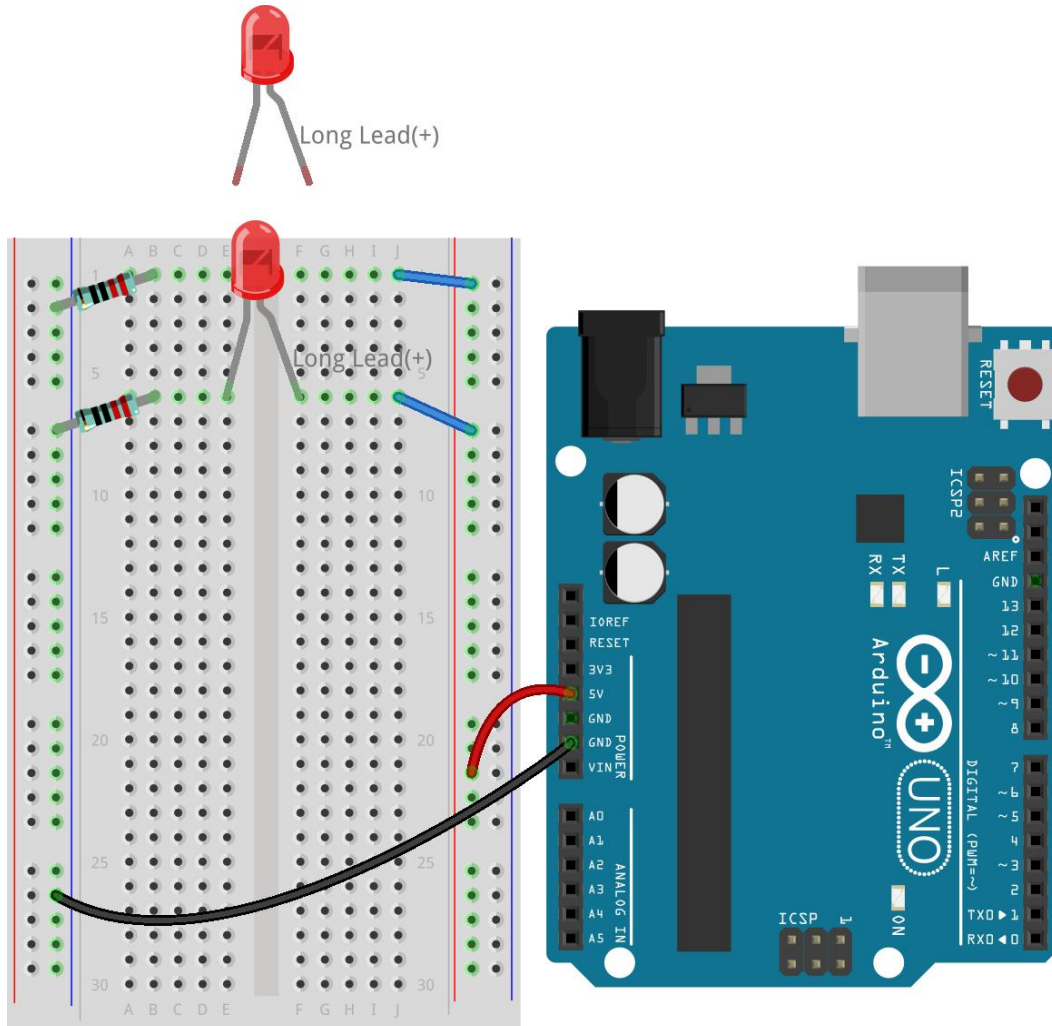
Agregar otro LED a un circuito en serie con dos LEDs generalmente resultará en una disminución del brillo de cada LED. Esto ocurre porque el voltaje total de la fuente de alimentación se divide entre más componentes, lo que da como resultado una caída de voltaje menor en cada LED en comparación con cuando solo había dos. En consecuencia, fluye menos corriente a través de cada LED, reduciendo su brillo.

En cuanto a los voltajes a través de los tres LEDs, cada uno tendrá ahora una porción menor del voltaje total del circuito. Si el voltaje de la fuente de alimentación se mantiene igual, este voltaje se dividirá entre los tres, asumiendo que todos los LEDs tienen características eléctricas similares. Por lo tanto, el voltaje a través de cada LED en el circuito será aproximadamente un tercio del voltaje total suministrado por la fuente de alimentación.



Complete las siguientes preguntas durante la sección "Profundizando en los circuitos en paralelo".

1. En este circuito en paralelo, ¿qué sucede si se retira un LED? ¿Por qué ocurre esto?



En un circuito en paralelo, si se retira un LED, los otros LEDs en el circuito continuarán encendiéndose. Esto ocurre porque cada LED en un circuito en paralelo tiene su propio camino independiente hacia la fuente de alimentación. Retirar un LED no interrumpe el flujo de corriente hacia los demás LEDs, por lo que no se ven afectados y continúan funcionando con normalidad. Esta configuración permite que cada componente en un circuito en paralelo funcione de manera independiente de los demás.

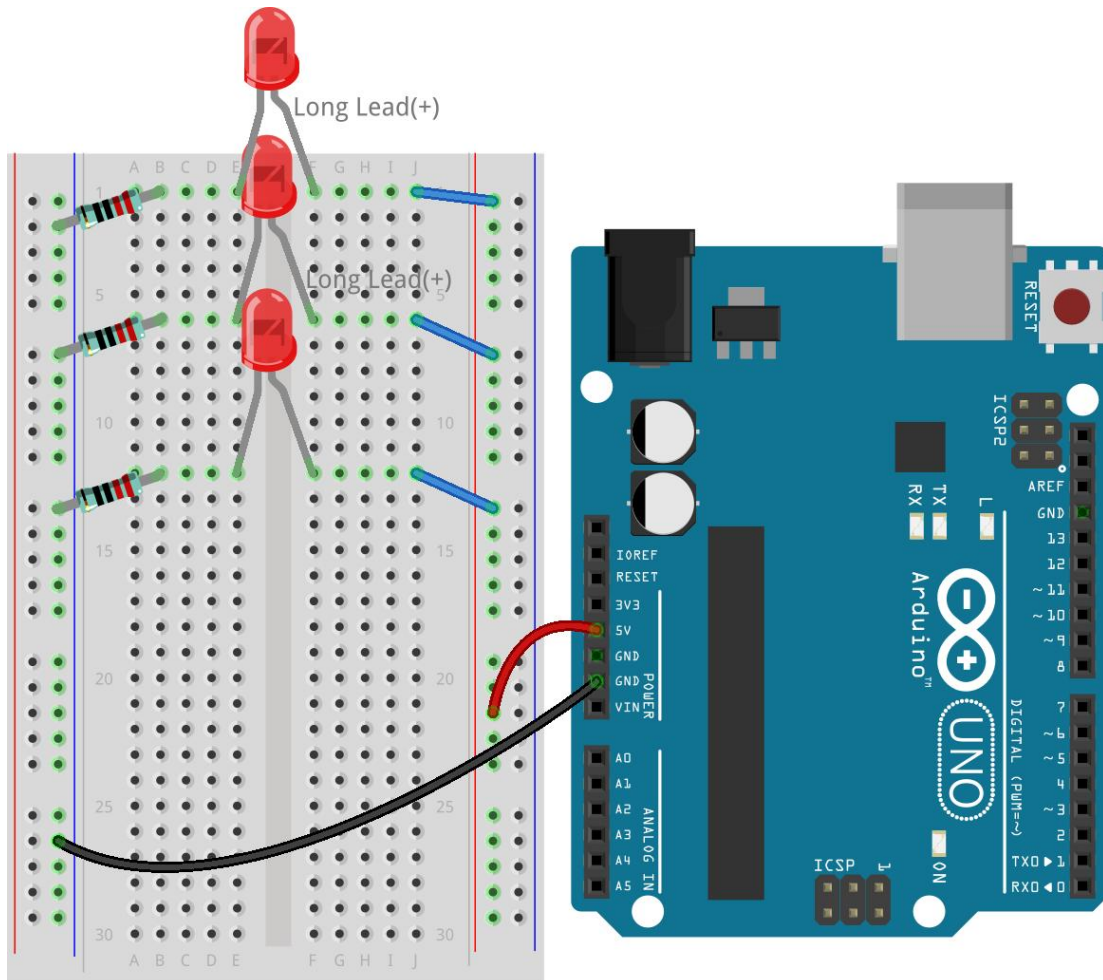
2. Registre el voltaje medido en la tabla.

Circuito	Voltaje del Camino 1	Voltaje del Camino 2
2 LEDs	<i>≈5.00 volts</i>	<i>≈5.00 volts</i>

3. Registre la corriente medida en la tabla.

Circuito	Corriente del LED1	Corriente del LED2	Corriente Total
2 LEDs	$\approx 12.6 \text{ milliamps}$	$\approx 12.6 \text{ milliamps}$	$\approx 25.3 \text{ milliamps}$

3. Si se añade otro LED a este circuito, ¿qué sucede con el brillo de los LEDs? ¿Por qué? Registre su respuesta en su manual.



Cuando se añade otro LED a un circuito en paralelo, el brillo de los LEDs existentes generalmente permanece sin cambios. Esto se debe a que cada LED en un circuito en paralelo tiene su propio camino directo hacia la fuente de alimentación, por lo que el voltaje a través de cada LED permanece constante independientemente de cuántos LEDs se añadan. Cada LED recibe el voltaje completo que necesita para operar a su brillo previsto. Por lo tanto, agregar más LEDs no afecta el brillo de los LEDs ya presentes, siempre y cuando la fuente de alimentación pueda soportar la demanda total de corriente del circuito. Asegúrese de registrar esto en su manual para futuras referencias.

## Lección 6: Parpadeo del LED

Complete la siguiente tabla durante la sección "Dando vida a los LEDs".

1. Registre el voltaje medido en la tabla para el Pin 3.

Estado	Voltaje del Pin 3
ALTO	<i>≈4.95 volts</i>
BAJO	<i>0.00 volts</i>

Después de completar la lección, responda la siguiente pregunta.

1. Cargue el código anterior y verá que el LED parpadea repetidamente a intervalos de 3 segundos. Si solo desea que se encienda y apague una vez, ¿qué debería hacer?

Puede mover los comandos que encienden y apagan el LED desde la función `loop()` a la función `setup()`. La función `setup()` se ejecuta solo una vez cuando comienza el programa, por lo que este cambio hará que el LED se encienda y apague una sola vez. A continuación, se muestra cómo puede ajustar su código:

```
void setup() {
  // Setup code here, to run once:
  pinMode(3, OUTPUT); // set pin 3 as output

  digitalWrite(3, HIGH); // Light up the LED on pin 3
  delay(3000);           // Wait for 3 seconds
  digitalWrite(3, LOW);  // Switch off the LED on pin 3
}

void loop() {
  // Main code here, to run repeatedly:
}
```

## Lección 7: ¡Hagamos un semáforo!

Complete la siguiente pregunta durante la sección "Escribir pseudocódigo para un semáforo".

---

Piense en lo que debe suceder para que su circuito funcione como un semáforo. En el espacio proporcionado en su registro, escriba el pseudocódigo que describa cómo funcionará su semáforo. Use un lenguaje sencillo.

Para simular un semáforo usando un Arduino, necesitaría una configuración con tres LEDs (rojo, amarillo y verde) y una secuencia que controle el encendido de manera que imite el funcionamiento de los semáforos reales. A continuación, se presenta un esquema simple de pseudocódigo que puede escribir en su registro para describir cómo podría funcionar este circuito de semáforo:

Setup:

- Define pins for the red, yellow, and green LEDs.
- Set all these pins as outputs.

Main Loop:

- Turn on the red LED for 5 seconds.
- Turn off the red LED.
- Turn on the yellow LED for 2 seconds.
- Turn off the yellow LED.
- Turn on the green LED for 5 seconds.
- Turn off the green LED.
- Repeat the cycle.

Complete la siguiente pregunta durante la sección "Escribir pseudocódigo para un semáforo".

---

Observe las intersecciones alrededor de su hogar. ¿Cuántos semáforos hay normalmente? ¿Cómo se coordinan entre sí?

En las zonas urbanas, las intersecciones suelen tener semáforos para gestionar de manera eficiente el flujo de vehículos y peatones. El número de semáforos en una intersección puede variar mucho según su tamaño y complejidad. Una intersección simple de cuatro vías suele tener al menos cuatro semáforos, uno orientado en cada dirección del tráfico. Las intersecciones más complejas pueden tener semáforos adicionales para carriles de giro, cruces peatonales y otras necesidades de gestión del tráfico.

## Lección 8: Semáforo con botón para peatones

Después de completar la sección "**Construcción del circuito**", responda la siguiente pregunta.

1. Su semáforo es una combinación de circuitos en serie y en paralelo. Analice qué partes de su circuito están en serie y por qué. Luego, explique qué partes están en paralelo y por qué.

En el circuito, el botón y su resistencia pull-down de 10K están conectados en serie. Esta configuración asegura que, cuando se presiona el botón, cambie correctamente el estado del pin 8 al conectarlo directamente a tierra cuando no está presionado, evitando entradas flotantes.

Los tres LEDs conectados a los pines 3, 4 y 5 están en paralelo entre sí. Cada LED funciona de manera independiente porque están conectados a pines de control separados y comparten una fuente de alimentación común. Esta configuración permite que cada LED funcione sin afectar a los demás, lo cual es fundamental para un sistema de semáforo.

Rellene esta tabla durante la sección "**Creación de código**".

1. Complete la tabla con el voltaje medido en el pin 8 cuando el botón está presionado y cuando no lo está. Luego, complete los estados correspondientes de nivel alto y bajo.

Estado del botón	Voltaje del Pin 8	Estado del Pin 8
Soltado	0.00 volts	BAJO
Presionado	≈4.97 volts	ALTO

Complete la siguiente pregunta al finalizar esta lección.

1. Durante la prueba, puede notar que el LED verde solo parpadea mientras se mantiene presionado el botón para peatones, pero los peatones no pueden cruzar la calle manteniendo el botón presionado continuamente. ¿Cómo podría modificar el código para asegurarse de que, una vez presionado el botón para peatones, el LED verde se mantenga encendido el tiempo suficiente para un cruce seguro sin necesidad de mantener el botón presionado? Escriba la solución en pseudocódigo en su manual.

Para asegurarse de que el LED verde se encienda para los peatones sin necesidad de mantener el botón presionado y que el ciclo normal del semáforo continúe después,

puede ajustar su pseudocódigo para verificar la pulsación del botón y luego cambiar el estado de operación en función de esa pulsación. A continuación, se presenta una versión optimizada y más clara del pseudocódigo que refleja estos cambios:

**Setup:**

- Define pins for red, yellow, and green LEDs as output

- Define the button pin as input

**Main Loop:**

- Check if the button is pressed

- If button is pressed:

  - Turn off all LEDs

  - Turn on green LED for pedestrians

  - Delay 10 seconds

- Else:

  - Execute normal traffic light cycle:

    - Turn on green LED (for vehicles), turn off other LEDs

    - Delay 10 seconds

    - Turn on yellow LED, turn off other LEDs

    - Delay 3 seconds

    - Turn on red LED, turn off other LEDs

    - Delay 10 seconds



## Lección 9: Lámpara de escritorio regulable

Rellene esta tabla durante la sección "Construir el circuito".

1. Gire el potenciómetro en el sentido de las agujas del reloj desde la posición 1 hasta la 3, mida la resistencia en cada punto y registre los resultados en la tabla.

Punto de medición	Resistencia (kilohmios)
1	1.52
2	5.48
3	9.01

2. ¿Cómo cree que cambiaría el voltaje en A0 cuando el potenciómetro se gire en el sentido de las agujas del reloj y en sentido contrario?

Puede imaginar el potenciómetro como dos resistencias conectadas en serie dentro del circuito. De acuerdo con la medición de las resistencias, la resistencia entre A0 y GND aumenta a medida que se gira el potenciómetro en el sentido de las agujas del reloj. Dado que la corriente permanece constante en un circuito en serie, según la Ley de Ohm ( $\text{voltaje} = \text{corriente} \times \text{resistencia}$ ), un aumento en la resistencia genera un aumento en el voltaje en A0. Por lo tanto, girar el potenciómetro en el sentido de las agujas del reloj aumenta el voltaje en A0, mientras que girarlo en sentido contrario lo disminuye, ya que la resistencia disminuye.

Complete la siguiente pregunta al finalizar esta lección.

1. Si conecta el LED a un pin diferente, como el pin 8, y gira el potenciómetro, ¿cambiará aún el brillo del LED? ¿Por qué sí o por qué no?

Si conecta el LED al pin 8 de un Arduino UNO y gira el potenciómetro, el brillo del LED no cambiará. Esto se debe a que el pin 8 no es compatible con PWM (Modulación por Ancho de Pulso), que es necesario para ajustar los niveles de brillo utilizando la función `analogWrite()`. En un Arduino UNO, los pines que admiten PWM y que, por lo tanto, se pueden utilizar para controlar el brillo de un LED a través de `analogWrite()` son los pines 3, 5, 6, 9, 10 y 11.

# Lección 10: Lámpara de escritorio ON/OFF

Complete las siguientes preguntas al finalizar esta lección.

---

1. ¿Qué sucedería si configura el pin digital 7 solo como ENTRADA? ¿Por qué?

```
void setup() {  
  pinMode(9, OUTPUT); // Set pin 9 as output  
  pinMode(7, INPUT);  // Set pin 7 as input with an internal pull-up resistor  
  Serial.begin(9600); // Serial communication setup at 9600 baud  
}
```

Configurar el pin digital 7 en modo ENTRADA en su código de Arduino, en lugar de INPUT\_PULLUP, puede generar inestabilidad en la señal leída desde el pin. Cuando un pin se configura solo como ENTRADA y no está conectado a un voltaje alto o bajo a través de un circuito externo, se convierte en lo que se conoce como "flotante". Un pin flotante no está en un estado estable alto o bajo; su estado puede fluctuar debido al ruido eléctrico o a interferencias del entorno. Esta fluctuación puede provocar lecturas impredecibles cuando intenta leer el estado del pin a través de funciones de entrada digital, lo que genera datos erróneos o inconsistentes que recibe el microcontrolador.

2. Si el pin 7 se configura solo como ENTRADA, ¿qué ajustes se deben hacer en el circuito?

Si el pin 7 de su Arduino está configurado en modo ENTRADA y desea asegurar lecturas estables y predecibles, debe agregar una resistencia pull-up externa al circuito. Esto implica conectar una resistencia de 10kΩ entre el pin 7 y la fuente de alimentación de 5V del Arduino. La resistencia pull-up garantiza que el pin de entrada esté en un estado alto (nivel lógico 1) cuando no haya otra señal de entrada presente.

# Lección 11: Control de matrices de LED con potenciómetro

Responda las siguientes preguntas antes de continuar con "Creación de código".

---

1. Escriba su pseudocódigo para la matriz de LEDs.

El pseudocódigo sirve como un esbozo del programa, redactado en un lenguaje sencillo para facilitar la comprensión. Su tarea es crear pseudocódigo para una matriz de LEDs que reacciona a un potenciómetro. A medida que aumenta el valor del potenciómetro, se encenderán más LEDs.

A continuación se presenta un pseudocódigo simplificado que describe el control de una matriz de LEDs en función de la entrada de un potenciómetro:

```
Declare readValue variable.  
Setup.  
Declare 3 digital pin outputs.  
Main Loop.  
If the potentiometer's value is below 200, all LEDs should be off.  
If the value is between 200 and 600, the first LED should be on.  
If the value is between 600 and 1000, the first two LEDs should be on.  
If the value exceeds 1000, all LEDs should be on.  
Delay for a short time.
```

Después de completar la lección, responda la siguiente pregunta.

---

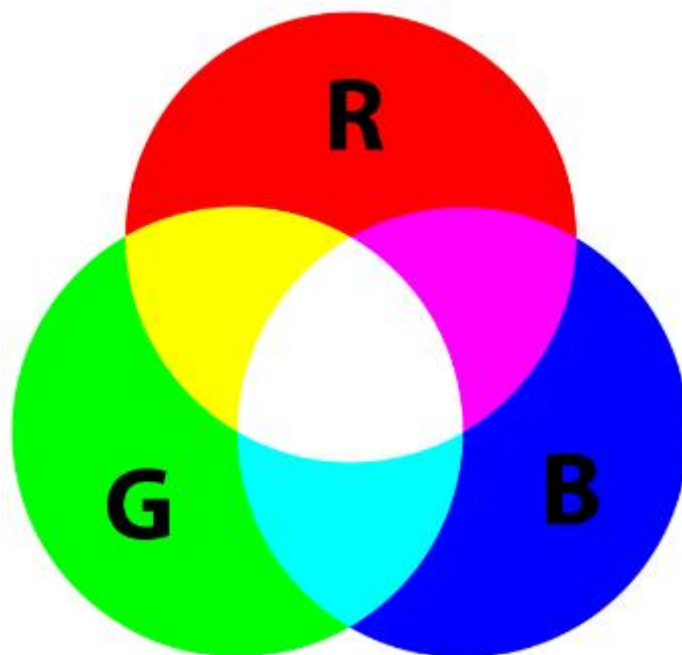
1. En el último código, determinamos la cantidad de LEDs que se encenderán según el valor del potenciómetro. ¿Cómo podemos modificar el código para que, al encender los LEDs, su brillo cambie de acuerdo con el potenciómetro?

Para modificar su código de manera que el brillo de los LEDs cambie según el valor del potenciómetro, puede usar la función `analogWrite()` en lugar de `digitalWrite()`. La función `analogWrite()` le permite controlar el brillo de los LEDs a través de PWM (Modulación por Ancho de Pulso).

## Lección 12: Los Colores del Arcoíris

Rellene esta tabla durante "Creación de Código".

1. Si deseas otros colores, ¿qué debes hacer? Consulta el diagrama a continuación y anota tus ideas en tu cuaderno.



Color	Pin Rojo	Pin Verde	Pin Azul
Rojo	<i>ALTO</i>	<i>BAJO</i>	<i>BAJO</i>
Verde	<i>BAJO</i>	<i>ALTO</i>	<i>BAJO</i>
Azul	<i>BAJO</i>	<i>BAJO</i>	<i>ALTO</i>
Amarillo	<i>ALTO</i>	<i>ALTO</i>	<i>BAJO</i>
Rosa	<i>ALTO</i>	<i>BAJO</i>	<i>ALTO</i>
Cian	<i>BAJO</i>	<i>ALTO</i>	<i>ALTO</i>
Blanco	<i>ALTO</i>	<i>ALTO</i>	<i>ALTO</i>

## Lección 13: El Espectro de la Vista

Rellena esta tabla durante "Creación de Código".

1. Ahora puedes ajustar los valores de los pines 9, 10 y 11 de forma independiente, y registrar los colores observados en tu cuaderno.

Pin Rojo	Pin Verde	Pin Azul	Color
0	128	128	<i>Azul oscuro</i>
128	0	255	<i>Púrpura</i>
128	128	255	<i>Azul claro</i>
255	128	0	<i>Naranja</i>

2. Elecciona algunos de tus colores favoritos y completa la tabla con sus valores RGB.

Color	Rojo	Verde	Azul

## Lección 14: Colores Aleatorios

Después de completar la lección, responde las siguientes preguntas.

1. Si cambias el código de `randomSeed(analogRead(A0))` a `randomSeed(0)`, ¿cómo cambiarán los colores del LED RGB y por qué?

Si cambias el código de `randomSeed(analogRead(A0))` a `randomSeed(0)`, la aleatoriedad de los colores del LED RGB se verá afectada. La función `randomSeed(seed)` se utiliza para inicializar el generador de números pseudoaleatorios en Arduino, lo que influye en la secuencia de números aleatorios generados por funciones como `random()`.

Utilizar `analogRead(A0)` como valor de semilla lee un valor algo aleatorio del pin analógico A0, que generalmente se ve afectado por el ruido ambiental y otros factores, lo que provoca que los valores de semilla sean diferentes cada vez que se inicia el programa. Esto significa que la secuencia de números aleatorios (y por lo tanto los colores) variará cada vez que se reinicie el Arduino.

En contraste, establecer la semilla con un valor fijo como `randomSeed(0)` inicializa el generador de números aleatorios con el mismo punto de partida cada vez que se ejecuta el programa. Esto resulta en la misma secuencia de números aleatorios y, por lo tanto, el LED RGB mostrará el mismo patrón de colores cada vez que reinicies o enciendas el Arduino. Esto elimina la aleatoriedad en los cambios de color del LED.

1. ¿Cuáles son algunas situaciones en las que se utiliza la aleatoriedad para resolver problemas en la vida cotidiana, además de seleccionar colores al azar para decoración y elegir números de lotería?

La aleatoriedad se utiliza en diversos contextos cotidianos para resolver problemas, incluyendo:

- **Juegos de mesa:** Lanzar los dados para determinar los movimientos, asegurando que cada partida sea diferente y justa.
- **Listas de reproducción musical:** Reproducir las canciones de forma aleatoria para mantener tu experiencia de escucha fresca e impredecible.
- **Opciones de comida:** Elegir un restaurante o una comida al azar puede facilitar la toma de decisiones y hacerla más divertida cuando no estás seguro de qué comer.
- **Distribución de asientos:** Asignar los asientos al azar en eventos para mezclar a los invitados y fomentar las interacciones sociales.
- **Noches de cine:** Utilizar un sorteo aleatorio para elegir una película cuando todos tienen diferentes preferencias.



## Lección 15: Colores Fríos o Cálidos

Completa esta tabla durante la "Creación de Código"

1. Abre Paint o cualquier herramienta de selección de color, encuentra lo que consideres los colores más cálidos y más fríos, y registra sus valores RGB en tu cuaderno.

Tipo de Color	Rojo	Verde	Azul
Color Cálido	246	52	8
Color Frío	100	150	255

Después de completar la lección, responde la siguiente pregunta.

1. Ten en cuenta que los "límites inferiores" de un rango pueden ser mayores o menores que los "límites superiores", por lo que se puede utilizar la función `map()` para invertir un rango de números, por ejemplo:

```
y = map(x, 1, 50, 50, 1);
```

La función también maneja adecuadamente los números negativos, por lo que este ejemplo también es válido y funciona correctamente.

```
y = map(x, 1, 50, 50, -100);
```

Para `y = map(x, 1, 50, 50, -100);`, si `x` es igual a 20, ¿cuál debería ser `y`? Consulta la siguiente fórmula para calcularlo.

$$\frac{\text{Value} - \text{From Low}}{\text{From High} - \text{From Low}} = \frac{\text{Y} - \text{To Low}}{\text{To High} - \text{To Low}}$$

$$\text{Y} = \frac{\text{Value} - \text{From Low}}{\text{From High} - \text{From Low}} \times (\text{To High} - \text{To Low}) + \text{To Low}$$

Para `x=20` usando la fórmula de mapeo `y = map(x, 1, 50, 50, -100);` el valor de `y` sería aproximadamente `-8.16`.

## Lección 16: Alarma de Temperatura

Complete esta tabla durante "Construyendo el Circuito"

1. Lee el valor de resistencia a diferentes temperaturas y regístralo en la tabla a continuación.

Entorno	Resistance (kilohm)
Temperatura actual	9.37
Temperatura alta	6.10
Temperatura baja	12.49

Después de completar la lección, responde la siguiente pregunta.

1. En el código se calculan las temperaturas en Kelvin y Celsius. Si también deseas conocer la temperatura en Fahrenheit, ¿qué deberías hacer?

Este es el método estándar para convertir Celsius a Fahrenheit y te proporcionará la temperatura en Fahrenheit basada en el valor de Celsius que ya tienes de tus cálculos.

$$F = C * 1.8 + 32$$

2. ¿Puedes pensar en otras situaciones o lugares donde un sistema de monitoreo de temperatura, como el que construimos hoy, podría ser útil?

Los sistemas de monitoreo de temperatura son ampliamente aplicables en situaciones cotidianas y en diversos entornos. Aquí hay algunos ejemplos simplificados:

- **Confort en el hogar:** Ajusta automáticamente la calefacción o el aire acondicionado de tu hogar en función de lecturas de temperatura en tiempo real para mantener tu espacio habitable cómodo.
- **Jardinería:** Controla las temperaturas en invernaderos para asegurar que las plantas crezcan en condiciones óptimas. Agrega sistemas automatizados para ajustar las temperaturas según sea necesario.
- **Seguridad alimentaria:** - Mantén un registro de las temperaturas del refrigerador y el congelador para garantizar que los alimentos permanezcan seguros para el consumo, especialmente en restaurantes o durante el transporte de alimentos.
- **Atención médica:** Monitorea y registra las temperaturas en áreas de almacenamiento para medicamentos y vacunas sensibles a la temperatura, asegurando que se mantengan efectivos.

## Lección 17: Código Morse

Responde a la siguiente pregunta durante "Construyendo el Circuito"

---

1. ¿Qué sucederá si conectas el cátodo de un zumbador activo directamente a GND y el ánodo a 5V? ¿Por qué?

Si conectas el cátodo de un zumbador activo directamente a GND y el ánodo a 5V, el zumbador emitirá un sonido continuo. Esto ocurre porque el oscilador interno del zumbador se activa con la alimentación de 5V, lo que provoca que genere sonido hasta que el circuito se desconecte.

Después de completar la lección, responde la siguiente pregunta.

---

1. Utilizando la tabla de código Morse proporcionada, escribe un código para enviar el mensaje "Hola".

En código Morse, "Hola" se codificaría de la siguiente manera según los caracteres:

- **H:** ....
- **E:** .
- **L:** .-..
- **A:** .-.
- **O:** ---

Resumiendo, el código Morse para "Hola" es:

.... .-.. .-.. ---

En la comunicación práctica, generalmente hay una pausa más larga entre las palabras para diferenciarlas claramente, pero dado que "Hola" es una sola palabra, el código es continuo, con espacios que solo separan las letras individuales.

## Lección 18: Alarma de Luz

Completa esta tabla durante "Construyendo el Circuito"

1. Lee el valor de resistencia bajo la luz ambiental actual y regístralo en la tabla a continuación.

Ambiente	Resistencia (kiloohm)
Luz Normal	$\approx 5.48$
Luz Brillante	$\approx 0.16$
Oscuridad	$\approx 1954$

Después de completar la lección, responde a la siguiente pregunta:

1. 1.Ladrones astutos podrían optar por robar durante la noche, y si una pintura desaparece, el fotorresistor podría no detectar ningún cambio en la luz, lo que impediría activar la alarma. ¿Qué se puede hacer para mejorar esta deficiencia?

Una solución es instalar una fuente de luz frente a la pintura. Esto no solo garantiza que la pintura esté claramente visible e iluminada, sino que también significa que cualquier alteración o remoción de la pintura activará inmediatamente la alarma del fotorresistor al modificar el nivel de luz detectado.

## Lección 19: Sistema de Alarma para Estacionamiento en Reversa

Después de completar la lección, responde la siguiente pregunta:

---

1. Si deseas que la distancia detectada por este dispositivo sea más precisa hasta los decimales, ¿cómo deberías modificar el código?

Para hacer que la medición de distancia sea más precisa hasta los decimales, puedes modificar el cálculo en la función `measureDistance` para utilizar aritmética de punto flotante en lugar de aritmética entera. Esto permitirá que el resultado incluya valores decimales, proporcionando mediciones más precisas.

```
float distance = duration * 0.034 / 2.0;
```

La variable de `distance` en la función `measureDistance` se cambia a tipo float para admitir valores decimales.

El cálculo de la `distance` utiliza aritmética de punto flotante al dividir entre 2.0 en lugar de 2.

## Lección 20: El Temporizador Pomodoro

Responde la siguiente pregunta durante "Creación de Código - millis()"

---

1. Si se cambia `delay(100)` por `delay(1000)`, ¿qué sucederá con el programa? ¿Por qué?

```
if (currentMillis - previousMillis >= interval) {  
  previousMillis = currentMillis; // Save the last time the buzzer beeped  
  digitalWrite(buzzerPin, HIGH); // Make a voice  
  delay(100);  
  digitalWrite(buzzerPin, LOW); // silence  
}
```

In the original code, the buzzer beeps for about 100 milliseconds every 1000 milliseconds (1 second, as set by the `interval` variable), followed by a silence of 900 milliseconds. After the modification, the buzzer will beep for 1000 milliseconds every 1000 milliseconds, and then almost immediately beep again, as the next interval starts almost right away. Thus, changing the delay from 100 to 1000 milliseconds turns the buzzer from emitting brief beeps to continuous sound, which becomes more annoying and unsuitable for the original intent.

En el código original, el zumbador emite un beep durante aproximadamente 100 milisegundos cada 1000 milisegundos (1 segundo, como lo establece la variable de intervalo), seguido de un silencio de 900 milisegundos. Después de la modificación, el zumbador sonará durante 1000 milisegundos cada 1000 milisegundos y, casi de inmediato, volverá a sonar, ya que el siguiente `interval` comienza casi al instante. Por lo tanto, cambiar el retraso de 100 a 1000 milisegundos transforma el zumbador de emitir beeps breves a un sonido continuo, lo que resulta más molesto y poco adecuado para la intención original.

Cambiar `delay(100)` a `delay(1000)` en tu código hará que el zumbador suene durante un segundo completo en lugar de emitir solo un beep corto, ya que aumenta el tiempo de pausa mientras el zumbador está encendido. Esto resulta en ruidos más prolongados del zumbador y ciclos de programa menos frecuentes, lo que puede hacer que el programa sea menos receptivo a otras tareas durante estos intervalos.

Una vez completada la lección, responde a la siguiente pregunta.

---

Piensa en otros lugares de tu vida donde puedas "escuchar" el tiempo. ¡Enumera algunos



ejemplos y anótalos en tu cuaderno!

Escuchar el tiempo es un concepto intrigante, y hay varias situaciones cotidianas donde podemos experimentar esto. Aquí hay algunos ejemplos que podrías considerar anotar:

- **Relojes y cronómetros:** El tic-tac de los relojes analógicos o los pitidos específicos de los relojes digitales que señalan cada segundo o minuto que pasa.
- **Temporizadores de cocina:** El tic-tac y la alarma final de un temporizador mecánico o digital de cocina que cuenta el tiempo de cocción o horneado.
- **Campanas escolares:** El sonido de las campanas en las escuelas que marcan el inicio y el final de las clases o recreos.
- **Anuncios de transporte público:** Los pitidos o campanillas que preceden a los anuncios en las estaciones de tren o en los autobuses, señalando la inminente salida o llegada.
- **Horno microondas:** El sonido de pitido cuando el temporizador termina, señalando que el proceso de calentamiento ha finalizado.
- **Dispositivos de seguimiento de fitness o relojes deportivos:** Los pitidos o alarmas que indican la finalización de un tiempo determinado durante los entrenamientos o intervalos.

## Lección 21: Sonido de Sirena

Responde la siguiente pregunta durante **"Construir el Circuito"**.

---

1. ¿Qué sucederá si conectas el cátodo de un zumbador pasivo directamente a GND y el ánodo a 5V? ¿Por qué?

Si conectas el cátodo de un zumbador pasivo directamente a GND y el ánodo a 5V, a diferencia de un zumbador activo, el zumbador pasivo no emitirá ningún sonido por sí mismo, ya que no tiene un oscilador incorporado.

Un zumbador pasivo requiere una señal externa para generar sonido. Normalmente, necesitas alimentarlo con una onda cuadrada (voltaje oscilante) a la frecuencia deseada para crear sonidos audibles.

Responde a las siguientes preguntas durante **"Creación de Código - Hacer Sonar el Zumbador Pasivo"**.

---

1. Si cambias el código y los pines del circuito a 7 u 8, que no son pines PWM, ¿el zumbador seguirá sonando? Puedes hacer la prueba y luego escribir tu respuesta en el manual.

Aunque el pin 8 no es un pin PWM, la función `tone()` puede generar una onda cuadrada precisa en él, lo que permite activar un zumbador pasivo para producir sonido. Esta flexibilidad te permite utilizar cualquier pin digital para la salida de sonido sin estar limitado a los pines capaces de PWM. Cuando llamas a la función `tone(pin, frecuencia)`, Arduino configura un temporizador para alternar el estado del pin (de ALTO a BAJO y de nuevo a ALTO) a la frecuencia especificada, creando una onda cuadrada. Esta onda cuadrada activa el zumbador pasivo, haciéndolo emitir sonido a la frecuencia de la onda generada.

1. Para explorar cómo la frecuencia y la duración en tone(pin, frecuencia, duración) afectan el sonido del zumbador, por favor modifica el código bajo dos condiciones y registra los fenómenos observados en tu cuaderno:

- Manteniendo la frecuencia en 1000, aumenta gradualmente la duración de 100, 500 a 1000. ¿Cómo cambia el sonido del zumbador y por qué?
  - Duración de 100 ms: El sonido es un pitido corto.
  - Duración de 500 ms: El sonido es un pitido más largo, claramente audible y que dura medio segundo.
  - Duración de 1000 ms: El sonido es aún más prolongado, con una duración de un segundo completo.

A medida que aumentas la duración, el sonido emitido por el buzzer dura más. El tono o frecuencia del sonido se mantiene constante (ya que está establecido en 1000 Hz), lo que significa que la "nota" del tono no cambia, pero la duración del sonido que escuchas aumenta. Esto es útil para señalar diferentes duraciones de alertas, donde la urgencia o el tipo de alerta se pueden distinguir por la longitud del tono.

- Manteniendo la duración en 100, aumenta gradualmente la frecuencia, de 1000 a 2000 y luego a 5000. ¿Cómo cambia el sonido del buzzer y por qué?
  - Frecuencia de 1000 Hz: El sonido es un beep de tono medio.
  - Frecuencia de 2000 Hz: El sonido tiene un tono más agudo en comparación con 1000 Hz.
  - Frecuencia de 5000 Hz: El sonido es mucho más agudo, percibiéndose como más intenso y posiblemente incómodo a corta distancia.

Aumentar la frecuencia mientras se mantiene la duración constante provoca un cambio en el tono del sonido. Frecuencias más altas producen sonidos más agudos. Este principio es útil para distinguir entre diferentes tipos de notificaciones o señales según su urgencia o importancia, siendo los tonos más agudos a menudo utilizados para alertas más urgentes.

## Lección 22: Toca "Estrellita, ¿dónde estás?"

Responde la siguiente pregunta durante "Creación de Código - Arreglo"

1. También puedes realizar operaciones en los elementos del arreglo, como cambiar a `Serial.println(melody[i] * 1.3)`, ¿Qué datos obtendrás y por qué?

El número 1.3 es un número de punto flotante. Cuando un entero del arreglo `melody` (que es de tipo `int`) se multiplica por 1.3, el resultado de la operación se convierte automáticamente en un número de punto flotante (`float`).

Para cada frecuencia de nota en este arreglo, multiplicar por 1.3 y luego imprimir el resultado dará como resultado:

```
340.6  
340.6  
509.6  
509.6  
572.0  
572.0  
509.6  
...
```

Después de completar la lección, responde a la siguiente pregunta:

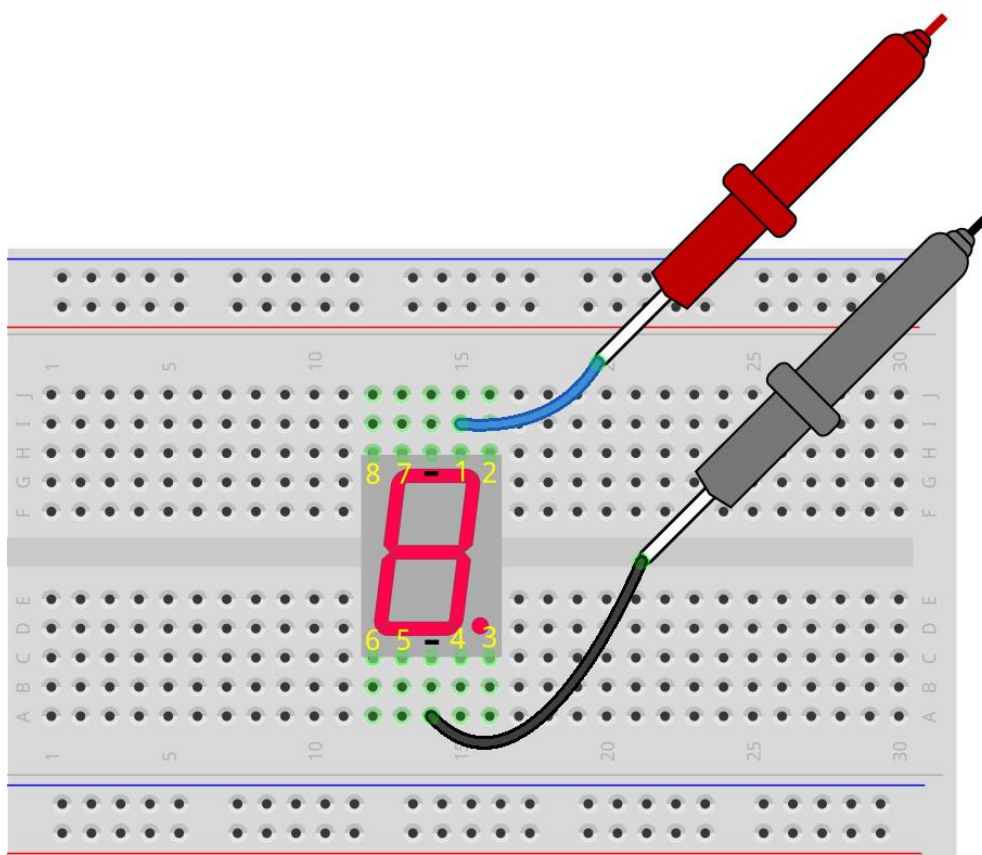
1. Si reemplazas el zumbador pasivo en el circuito por un zumbador activo, ¿puedes reproducir correctamente "Estrellita, ¿dónde estás?"? ¿Por qué?

Si reemplazas el zumbador pasivo por un zumbador activo para reproducir "Estrellita, ¿dónde estás?", no funcionará como se espera. Los zumbadores activos solo pueden producir un tono único porque tienen un oscilador integrado. Por lo tanto, no podrás controlar la tonalidad para tocar la melodía con precisión; solo escucharás un pitido repetitivo en el ritmo de la canción, no las notas reales.

## Lección 23: Datos Cibernéticos

Responde a las siguientes preguntas durante "Comprendiendo el Display de 7 Segmentos"

1. Si un segmento se ilumina, consulta este diagrama para registrar el número de pin del segmento y su posición aproximada en la tabla del Manual.



Pin	Número de segmento	Posición
1	A	La parte superior
2	B	La parte superior derecha
3	C	La parte inferior derecha
4	D	La parte inferior
5	E	La parte inferior izquierda
6	F	La parte superior izquierda
7	G	El centro
8	punto decimal	El punto

1. A partir de las pruebas anteriores, se sabe que el display del kit es de cátodo común, lo que significa que solo necesitas conectar el pin común a GND y proporcionar un voltaje alto a los otros pines para encender los segmentos correspondientes. Si deseas que el display muestre el número 2, ¿qué pines deben recibir un voltaje alto? ¿Por qué?



Para el número 2, es necesario activar los segmentos a, b, d, e y g (configurarlos a un voltaje alto), ya que son los segmentos que forman el número 2 en el display. Los segmentos f, c y dp (punto decimal, si está presente) deben permanecer apagados (bajo voltaje) ya que no forman parte de la representación del número 2.

Por lo tanto, los pines que deben recibir un voltaje alto son aquellos conectados a los segmentos a, b, d, e y g para mostrar correctamente el número 2.

## Lección 24: Luz Fluorescente con 74HC595

Responde a la siguiente pregunta durante "Creación de Código - Encendido de LEDs"

1. ¿ Qué sucede si cambiamos `MSBFIRST` por `LSBFIRST` en `shiftOut(DS, SHcp, MSBFIRST, B11101110);`? ¿Por qué?

Si cambias `MSBFIRST` por `LSBFIRST`, el orden de los bits se invierte, y el byte se envía comenzando desde el bit menos significativo (el más a la derecha). Si estás utilizando el registro de desplazamiento para controlar LEDs, cambiar el orden de los bits invertirá la secuencia en la que se encienden los LEDs. En lugar de iluminarse en la secuencia originalmente programada, se encenderán en el orden inverso.

Una vez completada la lección, responde a la siguiente pregunta.

1. Si queremos que tres LEDs estén encendidos al mismo tiempo y que parezca que "fluyen", ¿cómo deberían modificarse los elementos del arreglo `dataArray[]`?

Comenzarías con los primeros tres LEDs encendidos y luego moverías un LED hacia la derecha en cada patrón subsiguiente hasta que los últimos tres LEDs estén encendidos. Aquí te mostramos cómo podrías definir estos patrones en binario:

```
B11100000: LEDs 1, 2, 3 are on; others are off.  
B01110000: LEDs 2, 3, 4 are on; others are off.  
B00111000: LEDs 3, 4, 5 are on; others are off.  
B00011100: LEDs 4, 5, 6 are on; others are off.  
B00001110: LEDs 5, 6, 7 are on; others are off.  
B00000111: LEDs 6, 7, 8 are on; others are off.
```

```
byte dataArray[] = { B11100000, B01110000, B00111000, B00011100, B00001110,  
B00000111 };
```



## Lección 25: Mostrar Número

Completa esta tabla durante "Números Binarios para los Dígitos del 0 al 9"

1. Ahora que conocemos las representaciones binarias para los dígitos 0 y 2, por favor completa los números binarios para los dígitos restantes en la tabla a continuación.

Número	Binario
0	<i>B00111111</i>
1	<i>B00000110</i>
2	<i>B01011011</i>
3	<i>B01001111</i>
4	<i>B01100110</i>
5	<i>B01101101</i>
6	<i>B01111101</i>
7	<i>B00000111</i>
8	<i>B01111111</i>
9	<i>B01101111</i>

Completa esta tabla durante "Conversión Binaria"

1. Por favor, convierte los números binarios que representan los dígitos del 0 al 9 en números decimales y hexadecimales utilizando una calculadora, y completa la tabla. Esto te proporcionará una guía de referencia rápida para las conversiones de base.

Número	Binario	Decimal	Hexadecimal
0	B00111111	63	0x3F
1	B00000110	6	0x06
2	B01011011	91	0x5B
3	B01001111	79	0x4F
4	B01100110	102	0x66

5	B01101101	<i>109</i>	<i>0x6D</i>
6	B01111101	<i>125</i>	<i>0x7D</i>
7	B00000111	<i>7</i>	<i>0x07</i>
8	B01111111	<i>127</i>	<i>0x7F</i>
9	B01101111	<i>111</i>	<i>0x6F</i>