

1. Reto 01: Encender y Apagar un Led.

1.1. Objetivo:

- ✓ Encender y apagar un LED con Arduino, en la cual se determinará el tiempo que debe permanecer en cada estado (para observar la intermitencia).
- ✓ Conocer y Aplicar los comandos e instrucciones Básicas del Lenguaje de Programación empleado en las Placas de Desarrollo Arduino.

1.2. ¿Qué Materiales emplearemos?:

- ✓ 1 Diodo LED.
- ✓ 1 Resistencias de 220 [Ω].
- ✓ 1 Tarjeta Arduino Uno.
- ✓ 1 cable USB (impresora).
- ✓ 1 computador.
- ✓ Varios Cables para el montaje del circuito.
- ✓ 1 Protoboard.

1.3. ¿Cómo conectamos un LED?:

Antes de empezar con el desarrollo de la práctica debemos saber que es un LED y como lo debemos conectar aprendiendo a diferenciar su ánodo y cátodo, que son sus terminales de conexión.

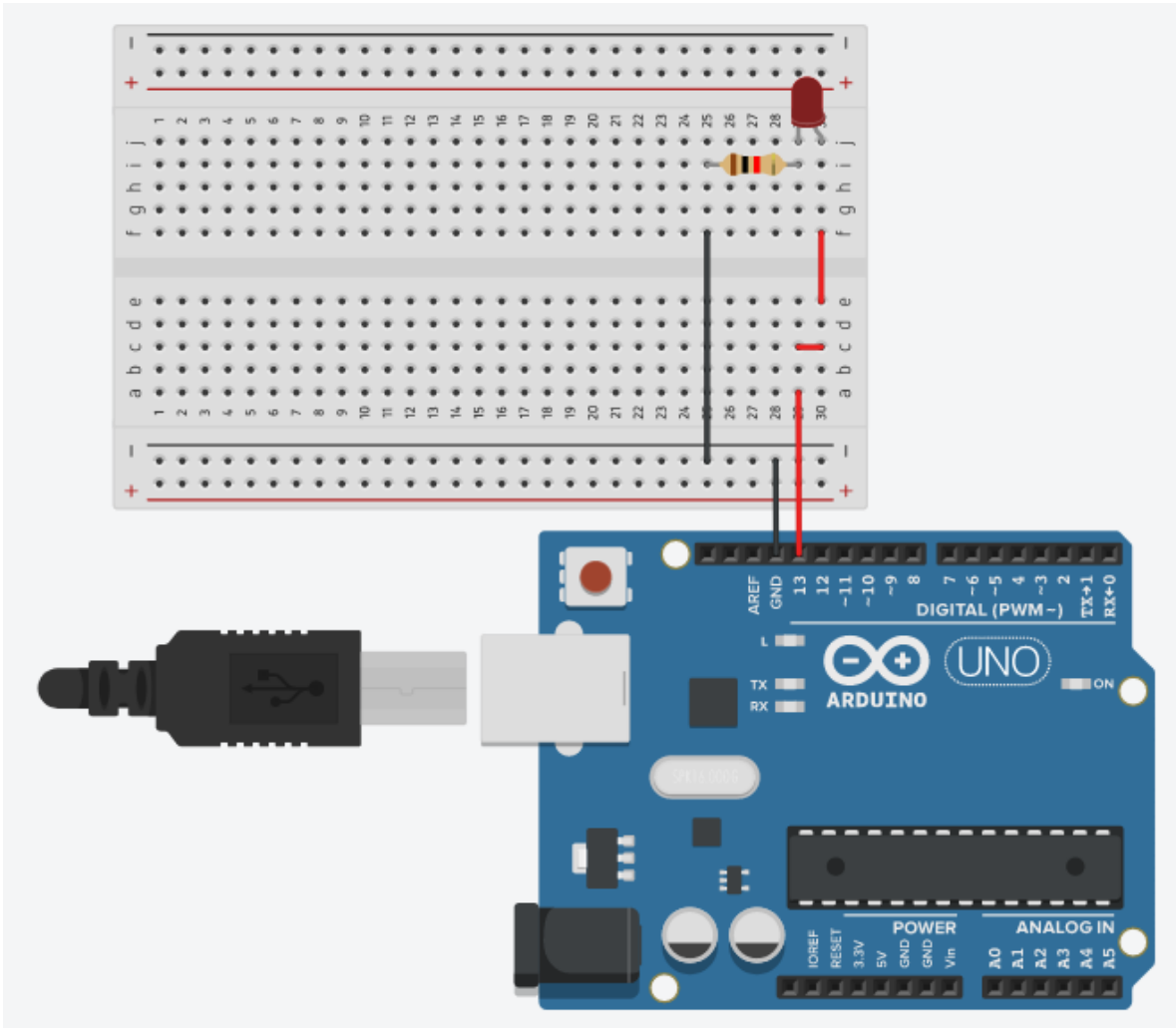
Un LED, es un dispositivo diodo emisor de luz que se usan como indicadores en muchos dispositivos y en iluminación. Normalmente emiten luz roja de baja intensidad, pero también existen otros que emiten una luz de alto brillo en el espectro infrarrojo, visible y ultravioleta. Un LED comienza a funcionar aproximadamente con 2[V].

Descubriremos dos formas de identificar los terminales y la polaridad de un LED:

- La pata más larga siempre va a ser el ánodo.
- En el lado del cátodo, la base del LED tiene un borde plano.

1.4. Montaje del Circuito del Reto:

En la siguiente figura se muestra el montaje del circuito electrónico en la plataforma TinkerCad. Colocaremos el ánodo en el pin 13 y el cátodo a tierra (ground), como se muestra a continuación:



Como se muestra en la figura anterior, para el montaje en la plataforma TinkerCad, al igual que en la vida real se utiliza un **Protoboard**, el cual es un tablero con orificios conectados eléctricamente entre sí, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables para el armado de prototipos de circuitos electrónicos y sistemas similares. Está hecho de dos materiales, un aislante, generalmente un plástico, y un conductor que conecta los diversos orificios entre sí. Uno de sus usos principales es la creación y comprobación de prototipos de circuitos electrónicos.

Posteriormente, luego de elaborar el circuito, procederemos a desarrollar el código que permitirá la programación del Arduino.

1.5. Algoritmo del Reto 01:

```

1 // 01 Practica: Encender y Apagar un Led.
2 // Diseño de: José Miguel.
3
4 const int LED_Rojo=13; //Declara y Asigna Nombre "LED_Rojo" de Variable al Pin 13.
5
6 void setup()
7 {
8   pinMode(LED_Rojo,OUTPUT); // Declara el Pin 13 ("LED_Rojo") como una Salida.
9 }
10 void loop() // Inicia el Ciclo que se repetirá durante la Ejecucion del Programa.
11 {
12   digitalWrite(LED_Rojo,HIGH); // Coloca en estado de ALTO el Pin (ON).
13   delay(1000); // Realiza una pausa de 1000 milisegundos.
14   digitalWrite(LED_Rojo,LOW); // Coloca en estado de BAJO el Pin (OFF).
15   delay(1000); // Realiza una pausa de 1000 milisegundos.
16 }

```

1.6. Análisis y Explicación del Algoritmo del Reto 01:

- ✓ Iniciamos indicando el nombre del programa a través de un comentario.
- ✓ Para insertar comentarios se emplea el símbolo `//`.
- ✓ Se declaran las variables. Si es de tipo constante, una variable que no cambia luego de definido su valor, su comando es `const`. Además, se debe especificar qué tipo de dato es la variable, para número entero su comando es `int`. Luego se da nombre a la variable y se conecta en la tarjeta Arduino el ánodo en el pin 13 asignándole el valor de 13 en el código.
- ✓ Se define el tipo de variable: entrada o de salida, mediante `void setup ()`, se abre llave “{” para saber qué es lo que contiene ese comando y al finalizar el comando se cierra con llave “}”. Internamente del corchete se declara que la variable “LED_Rojo” es de salida, mediante el comando `pinMode(LED_Rojo, OUTPUT)`; donde `OUTPUT` indica que la señal saldrá del pin 13.
- ✓ Luego de definir las variables, se desarrolla el código dentro del comando `void loop ()`, se abre llave “{” y se cierra luego de terminar su cumplimiento “}”. Internamente del corchete se establecen las instrucciones que ejecutará Arduino continuamente.
- ✓ El LED se enciende mediante el comando `digitalWrite`, la cual envía una señal digital al pin que anteriormente se indicó. Se indica `HIGH` para encender el pin. Los pines de Arduino que no tienen nada al lado, sino solamente el número ellos tienen una señal alta y baja; `LOW`=baja es cero (0) y `HIGH`=alta es uno (1). El código es `digitalWrite (LED_Rojo, HIGH)`. Luego se emplea el comando `delay` (milisegundos); que permite retrasar el tiempo que se le indique hasta la ejecución de la siguiente instrucción.
- ✓ El LED pueda apagarse indicándole esta vez una señal baja `LOW`, `digitalWrite (LED_Rojo, LOW)` y luego se emplea nuevamente el comando `delay`.

Observación: 1 Segundo = 1.000 milisegundos.

1.7. Código del Reto disponible para copiar:

```
// 01 Practica: Encender y Apagar un Led.  
// Diseño de: José Miguel.  
  
const int LED_Rojo=13; //Declara y Asigna Nombre "LED_Rojo" de Variable al Pin 13.  
  
void setup()  
{  
  pinMode(LED_Rojo, OUTPUT); // Declara el Pin 13 ("LED_Rojo") como una Salida.  
}  
void loop() // Inicia el Ciclo que se repetirá durante la Ejecución del Programa.  
{  
  digitalWrite(LED_Rojo, HIGH); // Coloca en estado de ALTO el Pin (ON).  
  delay(1000); // Realiza una pausa de 1000 milisegundos.  
  digitalWrite(LED_Rojo, LOW); // Coloca en estado de BAJO el Pin (OFF).  
  delay(1000); // Realiza una pausa de 1000 milisegundos.  
}
```

1.8. Para el siguiente Reto:

La siguiente experiencia consistirá en Encender y Apagar dos Led de Manera Alternada, es decir mientras uno esta encendido el otro se encuentra apagado.

Te invitamos a que te animes a intentar desarrollar el Reto teniendo presente las siguientes pistas:

- ✓ Se debe declarar una nueva variable para el segundo Led del Arduino.
- ✓ Configurar una nueva salida.
- ✓ Considerar que mientras un Led se encuentra en estado "HIGH" el otro se mantendrá en "LOW"