

## Actividades

El total de las actividades tienen un valor de 0,75 puntos dentro del bloque 1.

### Actividad 1.1: Dado electrónico (0,375 puntos)

Diseña un sistema con 3 leds y un botón. Cada vez que se pulse el botón haz que se genere un número aleatorio entre 1 y 3, el número obtenido se notificará al usuario encendiendo esa misma cantidad de leds.

Los leds se deben situar en una fila, y deberán encenderse siempre de forma ordenada, primero los encendidos y luego los apagados.

### Actividad 1.2: Juego de memoria (0,375 puntos)

El objetivo de esta actividad es desarrollar una versión simple del juego de memoria "simón". El sistema debe tener 2 leds de colores y 2 pulsadores (final de línea).

Cuando iniciamos la aplicación se debe generar una secuencia aleatoria de tres parpadeos en los leds, por ejemplo: Rojo, Rojo, Verde. (Entre parpadeo y parpadeo de cada led se debe esperar al menos un segundo).

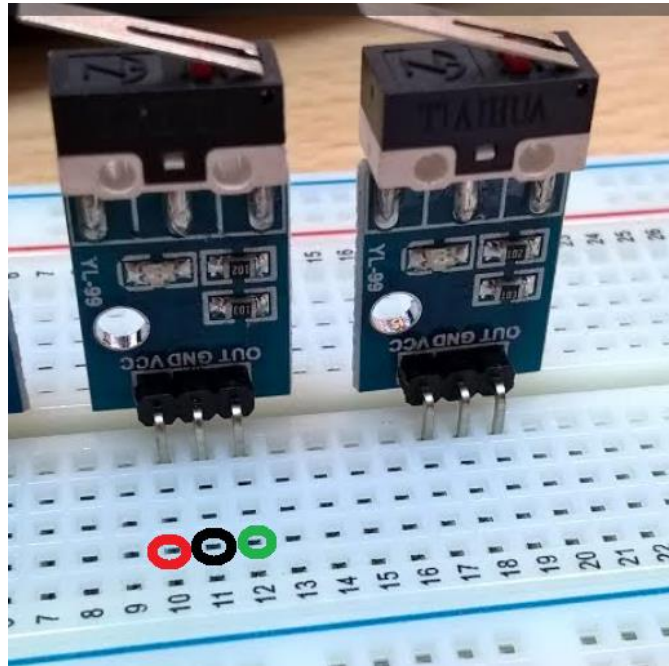
El usuario ahora debe reproducir esa secuencia utilizando los pulsadores, un pulsador se corresponderá a un color y el otro a otro. Es decir, para "acertar" la secuencia Rojo, Rojo, Verde el usuario debe pulsar el pulsador asociado al rojo dos veces y luego el pulsador asociado al verde.

Sí el usuario realiza la acción correctamente el juego sube de nivel, añadiendo un nuevo elemento aleatorio a la secuencia de parpadeos en los leds de colores. Cada vez que el usuario acierta la combinación se vuelve a repetir la secuencia anterior añadiendo un parpadeo adicional (Para ir aumentando la dificultad).

En el momento que el usuario realiza alguna pulsación incorrecta el juego finaliza y se inicia desde el principio, se debe volver a generar una secuencia aleatoria inicial.

Para esta actividad será necesario utilizar alguna estructura de datos y generación de números aleatorios, se debe dividir la funcionalidad del sistema en funciones.

Como utilizar el pulsador, se conecta de forma muy similar al potenciómetro. Lo encajamos en la protoboard a su patilla **VCC** hacemos llegar el voltaje, a la patilla **GND** la tierra y la patilla **OUT** un pin **digital** (devuelve HIGH o LOW dependiendo de si está pulsado o no) . Este pulsador ya tiene una resistencia integrada por lo que la conexión es directa, no tenemos que agregar ninguna resistencia adicional al circuito.



Conexión. VCC a 5V (Rojo) GND a GND (Negro) Out a un pin digital (Verde)

## Ampliaciones (no valorables en puntos)

Agregar un sonido característico a cada uno de los leds utilizando el zumbador, el sonido debe reproducirse cuando los leds parpadean o cuando el usuario pulsa el botón asociado a cada led.



**Zumbador:** capaz de producir varios sonidos. Se suele utilizar en alarmas y sonidos característicos del sistema.

Tiene una patilla positiva **VCC** una negativa **GND** y una patilla **I/O** que se conecta a un pin PWM, como el ~11, capaz de emular escrituras analógicas (de esta forma especificaremos los tonos). Dependiendo de la señal que le enviemos por el pin, emitirá un tono u otro.

```
int pzumbador = 11;

int cancion[] = {261, 349, 392, 440, 392, 330, -10, 261, 349, 392,
440, 392, -10, -10, 261, 349, 392, 440, 392, 330, -10, 330, 349, 330,
261, 261};
```

```

void setup()
{
  Serial.begin(9600);
  pinMode(pzumbador,OUTPUT);
}

void loop()
{
  for (int i = 0; i < sizeof(cancion)/sizeof(int); i++)
  {
    analogWrite(pzumbador,cancion[i]);
    delay(500);
  }
}

```

La api de Arduino provee de dos métodos específicos `tone` & `notone` que podemos utilizar para manejar el zumbador (**Utilizar este**)

<https://www.arduino.cc/en/pmwiki.php?n=Reference/Tone>

```

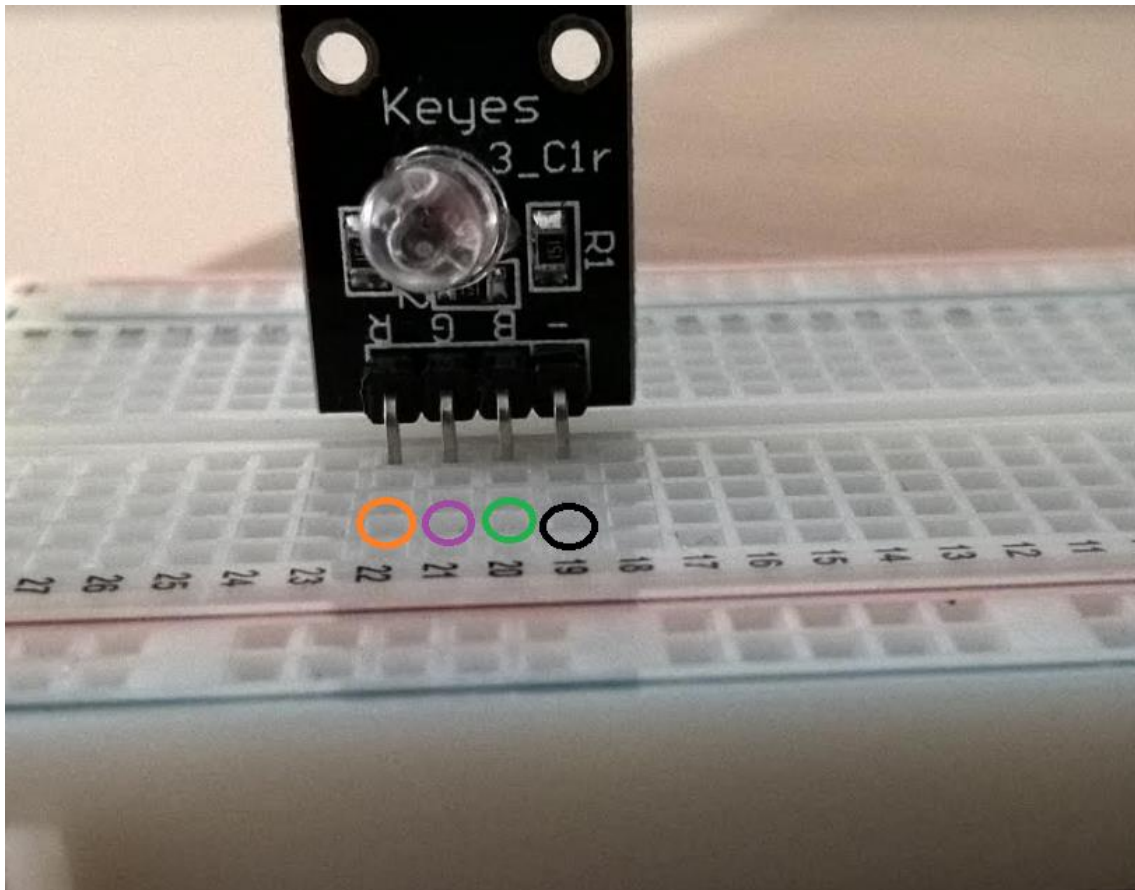
void loop()
{
  for (int i = 0; i < sizeof(cancion)/sizeof(cancion[0]); i++)
  {
    tone(pzumbador,cancion[i]);
    delay(500);
  }

  noTone(pzumbador);
  delay(5000);
}

```

En lugar de utilizar 2 leds normales utilizar un LED RGB. El led RGB no requiere utilizar una resistencia externa (está integrada). Dispone de una patilla negativa – (que se debe conectar al GND) y de tres patillas que deben ser conectadas a pines digitales, cada una de las patillas se corresponde con un color R, G, B. El color se iluminará si enviamos voltaje como salida por el pin correspondiente (Podemos utilizar solo una patilla de color R, G, B o varias, al enviar voltaje de forma simultanea por varias patillas podemos realizar combinaciones de colores).

Luz RGB	Placa
<b>R</b>	Pin digital
<b>G</b>	Pin digital
<b>B</b>	Pin digital
-	GND



*Conexión. VCC a 5V (Rojo) GND a GND (Negro) Out a un pin digital (Verde)*

Podemos utilizar la función `digitalWrite( <pin>, LOW/HIGH )` , si queremos que alumbrar al máximo de la capacidad del LED o apagarlo, si quisiéramos hacer combinaciones más complejas de colores, por ejemplo un poco de rojo y mucho verde podemos utilizar el `analogWrite( <pin> , [0-255] )` pero en este caso tenemos que estar seguros de usar pines PWM ( los señalados con ~ )

Si por ejemplo solo queremos utilizar el led como Rojo y verde basta con conectar los pines digitales a la patilla R y G, podemos dejar la B sin conexión.