

# Universidad Tecnológica Nacional

## Facultad Regional Avellaneda



### Tecnicatura Universitaria en Programación

#### Guía de ejercicios de Sistemas de Procesamiento de Datos - Arduino 2022

### Contenido:

Contenido:	1
Antes de empezar lea lo siguiente:	1
Conceptos Básicos	1
Salidas por consola	2
Funciones (A partir de acá, usar siempre millis en vez de delay)	3
Entradas Digitales	3
Trabajo Práctico 01: Cronómetro Binario	4

## Arduino - Parte 1

### Antes de empezar lea lo siguiente:

Respetar las reglas de estilo en cuanto a la documentación del código, se prolij@. Recuerda que cuando programamos, solo dios y vos saben que quisiste escribir, en cambio si no sos prolij@, cuando quieras ver el código dentro de varios años solo dios sabrá que codeaste.

## 1. Conceptos Básicos

- 1.1. Hacer que a través del pin n° 7, un led rojo prenda y apague cada 1 segundo, estando justamente un segundo apagado y un segundo prendido, usar la función 'delay'.

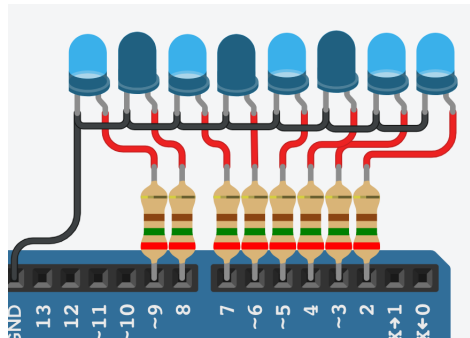
- 1.2. Hacer que a través del pin 7 y del pin 8, dos leds rojo y verde prendan y apaguen cada un segundo de manera tal que, cuando uno de ellos esté prendido, el otro debe permanecer apagado y viceversa. Usar la función 'delay'
- 1.3. Hacer que a través de los pines 7, 8 y 9 enciendan y apaguen 3 leds siguiendo una secuencia comenzando por el pin 7. Cuando un led está prendido, los otros dos deberán estar apagados. al llegar a un extremo la secuencia deberá continuar desde el extremo siguiente. Usar estructuras selectivas y flags o banderas de ser necesario.
- 1.4. Basandose en el ejercicio [1.3] hacer un circuito de tres leds (Verde, Amarillo y Rojo) que prendan y apaguen siguiendo una secuencia: Led verde prende por 2 segundos, Amarillo prende por 1 segundo y Rojo prende por 2 segundos (Como si fuese un semáforo). Cuando un led está prendido, el resto deberá estar apagado.
- 1.5. Mediante un contador, lograr que en un rango de 1 a 20, cuando el contador sea par se encienda un led verde en el pin 7, en caso de que sea impar se encienda un led azul en el pin 8. Para que el contador no avance rápido, se podrá usar la función 'delay'. Al llegar a 20, el contador deberá reiniciar tomando el valor de 0.
- 1.6. Tomando como base el ejercicio [1.5], lograr que además, cuando el contador sea múltiplo de 5, se encienda un led verde que estará en el pin 9. Usar estructuras selectivas.
- 1.7. Basandose en el ejercicio [1.4] Agregar un segundo semáforo (Leds Verde, Amarillo y Rojo) al circuito, el cual tendrá una secuencia de leds en dirección opuesta al primer semáforo. Cuando uno de ellos tenga prendido el led verde, el otro semáforo tendrá prendido el rojo.

## 2. Salidas por consola

- 2.1. Crear un programa que muestre por consola valores entre 0 y 25. Usar un delay de 500 y la función Serial.println().
- 2.2. Crear un programa que muestre por consola el cubo de números en un rango de 1 a 20. Usar un delay de 500.
- 2.3. Crear un programa que muestre por consola números primos menores que 100. Usar estructuras selectivas y el operador módulo (%) para determinar si un número es primo o no. Usar un delay de 250 entre numero y numero.
- 2.4. Tomando como base el ejercicio [2.3], hacer además que mediante el pin 7, encienda un led azul cada 5 números mostrados por consola. por ejemplo si muestra cuatro números primos, en el quinto que muestre encenderá el led durante la duración del delay y deberá apagarse.
- 2.5. Crear un programa que determine si un año es bisiesto usando como rango 1900 hasta 2021. Un año es bisiesto si es múltiplo de 4. Los años múltiplos de 100 no son bisiestos, salvo si ellos también son múltiplos de 400. Por ejemplo: el año 2000 es bisiesto pero 1900 no. Mostrar todos los años bisiestos en ese rango y al encontrar uno, hacer que encienda un led azul en el pin 7. Además mostrar un mensaje por consola con el formato: "El año X es bisiesto" (donde X será el año en cuestión). Nota: Utilizar un delay de 100, estructuras selectivas y el operador módulo (%).

### 3. Funciones (A partir de acá, usar siempre millis en vez de delay)

- 3.1. Reescribir el ejercicio [2.5] usando funciones para modularizar y así tener un mejor control del código. Cambiar el delay por la función 'millis()'
- 3.2. Desarrollar un programa que convierta a binario números entre 0 y 255, usando funciones. La cadena binaria resultante deberá ser usada para encender 8 leds ubicados en los pines 2 a 9. Recorrer la cadena binaria y prender el led correspondiente a la posición de la cadena donde haya un 1. Por ejemplo, si el número binario fuese '10101011', los leds encendidos deberán quedar como en la siguiente imagen. (Encender si hay un 1, apagar si hay un 0). Ayudarse de un contador para iterar números hasta 255. Usar millis para mostrar números cada segundo.



### 4. Entradas Digitales

- 4.1. Tomar como base el ejercicio [1.7] modificarlo usando funciones para modularizar el código y además agregarle un pulsador en el pin 11, para que al apretarlo una vez, comience la secuencia de los semáforos. Al pulsarlo nuevamente la secuencia debe terminar.  
**Implementar anti rebote en el pulsador.**
- 4.2. Tomando como base el ejercicio [3.2], agregarle tres pulsadores en los pines 10, 11 y 12. El pulsador del **pin 12 debe iniciar** y detener la simulación al pulsarse. El pulsador del **pin 11 debe disminuir** el número representado con los leds, mientras que el pulsador del **pin 10 debe aumentar** el número representado. Implementar funciones y anti rebote para los tres pulsadores.
- 4.3. Crear un circuito con cinco leds y un pulsador. La idea es que al apretar una vez al pulsador genere un número random entre 0 y 5 el cual será representado con la cantidad de leds encendidos. (Si el número que sale es un 3, se prenderán 3 leds, si sale 5 deberán estar los cinco leds encendidos). Para generar el número random, usar la función **random(min, max)** en la que 'min' será el número inicial del rango de números a generar y 'max' será el número máximo-1. Por lo tanto si min=0 y max=25, generará números entre 0 y 24. Tener en cuenta eso al momento de escribir el código. **Implementar funciones para modularizar y botón anti rebote.**

## 5. Trabajo Práctico 01: Cronómetro Binario

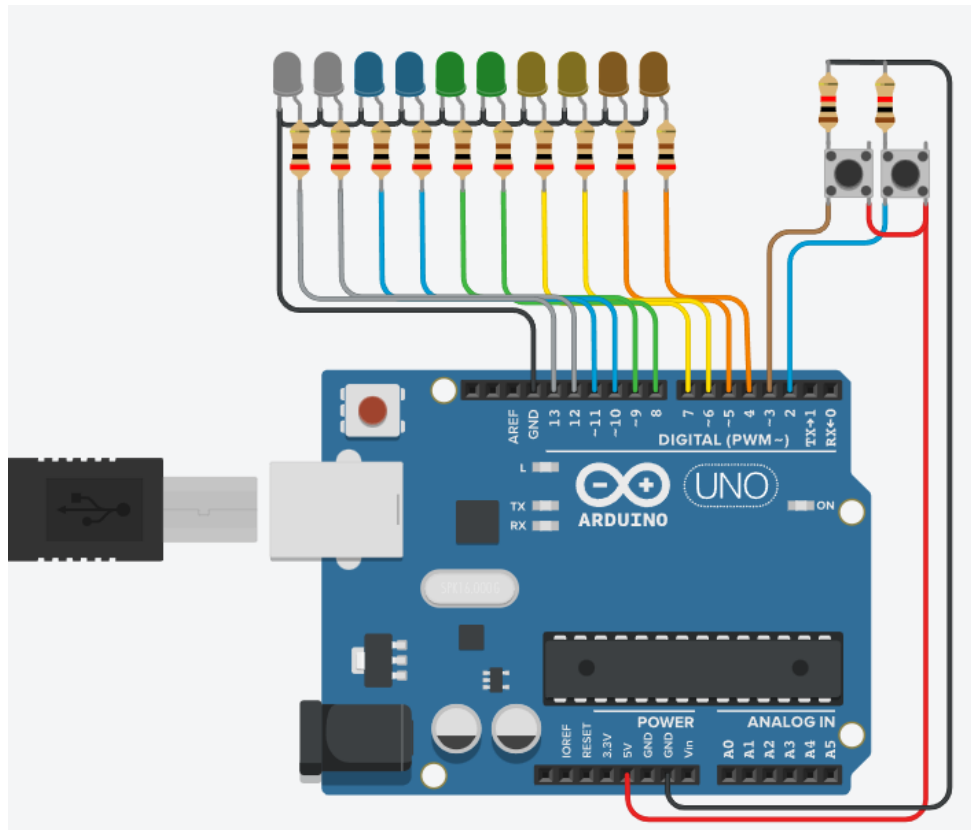
Crear un circuito con los conocimientos y buenas prácticas aprendidas el cual siga las siguientes directivas:

- 5.1. Al iniciar la simulación, todos los led deben comenzar apagados.
- 5.2. Cada 1 segundo que transcurra, se deben encender los led de acuerdo al número que representan.
- 5.3. El numero 15 (0000001111) se representaria encendiendo los leds AMARILLO y NARANJA.
- 5.4. Al pasar al número 16 (0000010000), se deben apagar todos los led anteriores y encenderse uno de los leds VERDES. (Nótese, en el ejemplo los 0 representan los led apagados y los 1 los led encendidos).
- 5.5. Al presionar el botón START, debe iniciar el cronómetro, al volver a presionarlo hará que la secuencia se detenga. (Como si se hubiese presionado pausa).
- 5.6. Al presionar el botón RESET, el cronómetro debe reiniciarse y arrancar de 0.
- 5.7. El número máximo a mostrar (tanto con leds como por consola) será el 1023 (Todos los leds prendidos) al pasarlo se deberán apagar los leds y no mostrar nada por consola.
- 5.8. Tip: Modularizar la función que controla el encendido de los LEDS y de ser posible, todo el código para evitar repetir líneas. Usar millis para controlar el tiempo del contador para que el cambio de los leds encendidos sea perceptible para el ojo humano y documentar cada función creada en el código. (Un breve comentario que diga que es lo que hace esa función).

Copiarse el circuito base del [SIGUIENTE LINK](#)

**[NOTA] Más información detallada dentro del código del circuito a modo de comentario.**

### Imagen del Circuito:



### Ejercicio Opcional Extra: [Suma 1 punto extra en la nota]

[OPCIONAL]: Pueden formatear un mensaje de consola el cual indique que segundo está imprimiendo y su representación en cadena binaria, pueden tomar como referencia la siguiente imagen.

