

# FUNCIÓN RANDOM Y MILLIS

Tutora: Angela Jazmín  
Miranda Flores



# FUNCIÓN RANDOM

Esta función nos genera números pseudoaleatorios pero bajo una misma secuencia, es decir que si iniciamos nuestra función random con los mismos valores se obtendrá la misma secuencia.

Sintaxis:

***Random (max);***

***Random (min,max);***

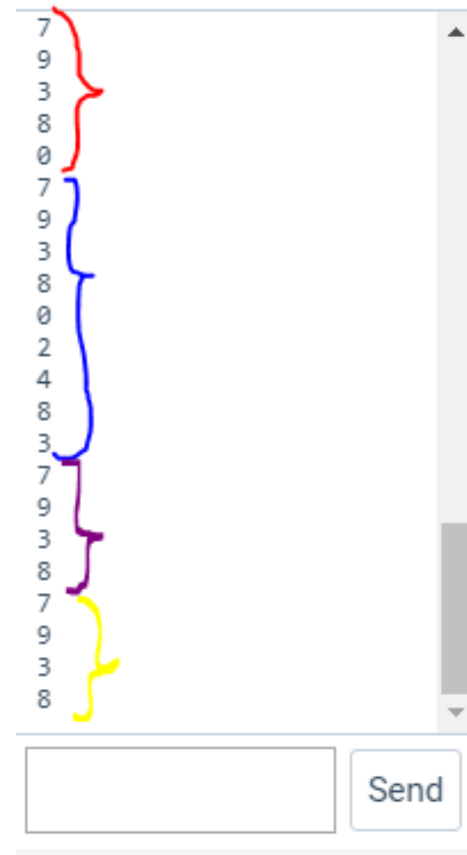
# EJEMPLO 1

Generar números random o aleatorios del 0-9

```
//AUTORA:Angela Miranda Flores
//CONTACTO angiejazminmiranda@gmail.com
void setup() {
  Serial.begin(9600);
}

void loop() {
  //genera el numero aleatorio del 0-9
  Serial.println(random(10));
}
```

Nótese que la  
secuencia de los  
números aleatorios  
se repite



# FUNCIÓN RANDOMSEED

Esta función inicializa el generador de los números pseudoaleatorios, pero genera una secuencia distinta cada vez que se reinicie el programa.

Sintaxis:

```
randomSeed(semilla);
```

parámetros para generar la semilla:

- int
- millis();
- analogRead();



## EJEMPLO 2

Generar números random o aleatorios del 0-9 y que la secuencia sea distinta luego de reiniciarse el sketch.

```
//AUTORA:Angela Miranda Flores
//CONTACTO angiejazminmiranda@gmail.com
void setup() {
  Serial.begin(9600);
  randomSeed(analogRead(A0));
  Serial.println("SECUENCIA DE NUMEROS ALEATORIOS...");
}
void loop() {
  Serial.println(random(10));
  delay(1000);
}
```

# FUNCIÓN MILLIS

Esta función nos devuelve el número de milisegundos desde que arduino empezó la ejecución del sketch.

Esta función se reiniciará después de aproximadamente 50 días.



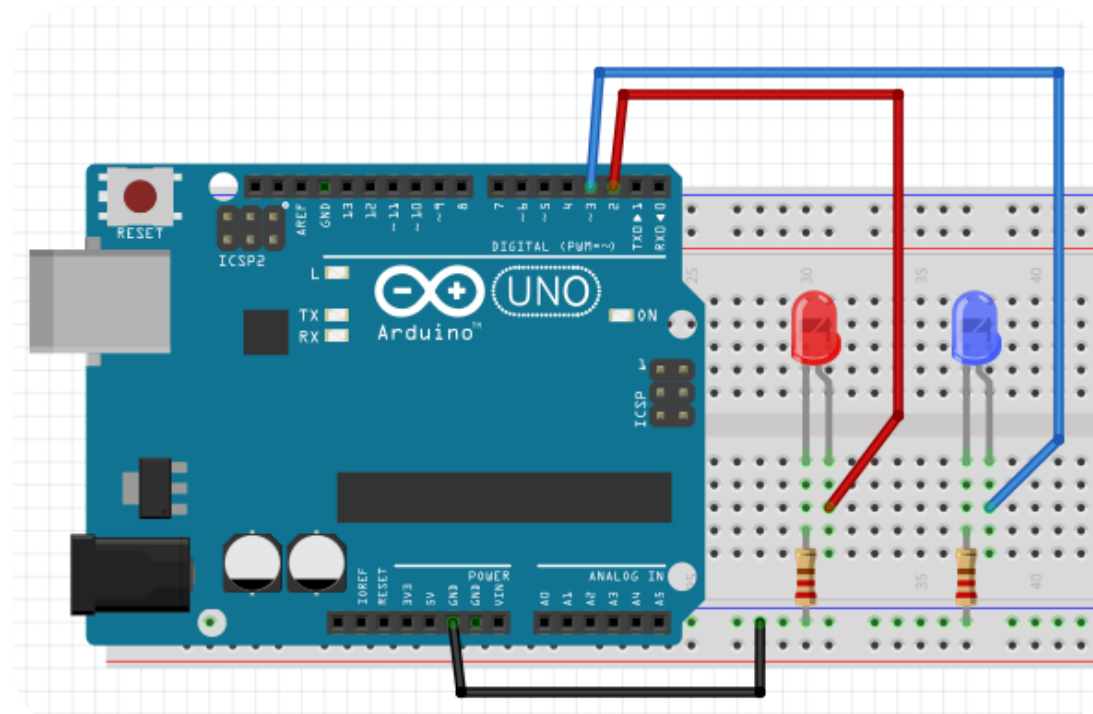
# EJEMPLO 3

Mostrar el tiempo que se ejecuta el programa

```
//AUTORA:Angela Miranda Flores
//CONTACTO angiejazminmiranda@gmail.com
unsigned long time;
void setup(){
    Serial.begin(9600);
}
void loop(){
    Serial.print("Time: ");
    time = millis();
    //imprime el tiempo desde que se ejecuta el programa
    Serial.println(time);
    // tiempo de espera de un segundo
    delay(1000);
}
```

# EJERCICIO EN CLASE PRIMERA SESIÓN

Simular las luces de un carro policía, los leds deben encenderse y apagarse a razón de 1 seg. Resolver sin utilizar delay's.



TUTORA: ANGELA JAZMIN MIRANDA FLORES



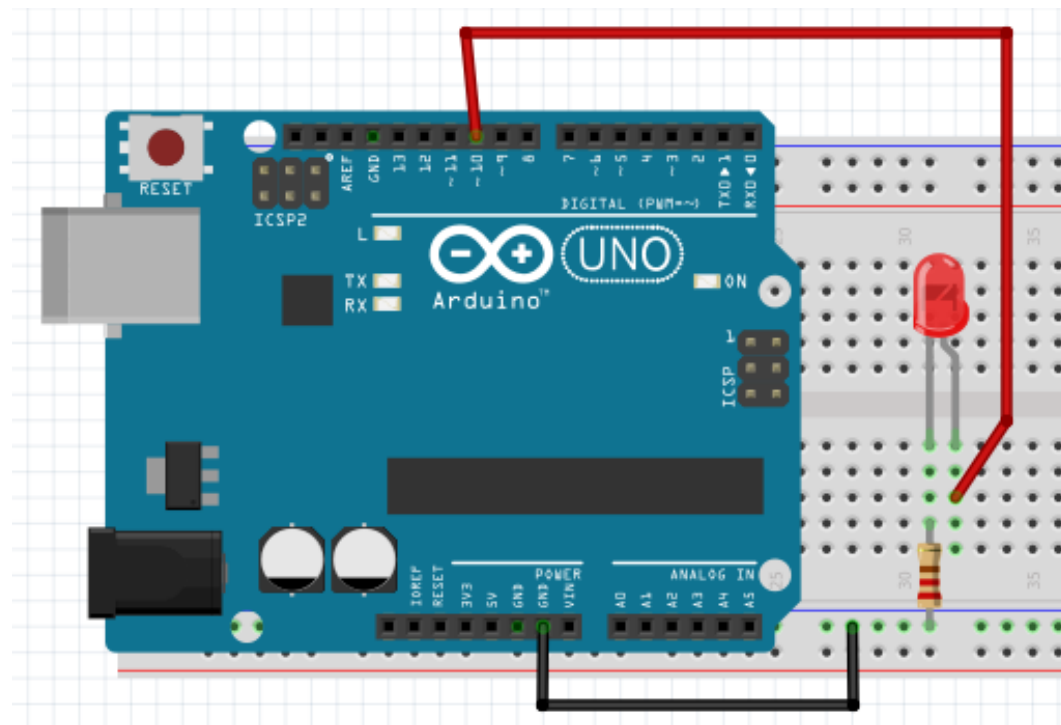
# SOLUCIÓN AL EJERCICIO EN CLASE

```
//AUTORA:Angela Miranda Flores
//CONTACTO angiejazminmiranda@gmail.com
unsigned long tiempo=0;
unsigned long t_delay=1000;
unsigned long t_actualizado=0;
bool estado=false;//led apagado
void setup() {
    Serial.begin(9600);
    pinMode(2,OUTPUT);
    pinMode(3,OUTPUT);
}
void loop() {
    tiempo=millis();//funcion millis
    if(tiempo==t_actualizado+t_delay){
        t_actualizado=tiempo;
        estado=!estado;//invierte el estado
        digitalWrite(2,estado);
        digitalWrite(3,!estado);
        Serial.println(tiempo);
    }
}
```

TUTORA: ANGELA JAZMIN MIRANDA FLORES

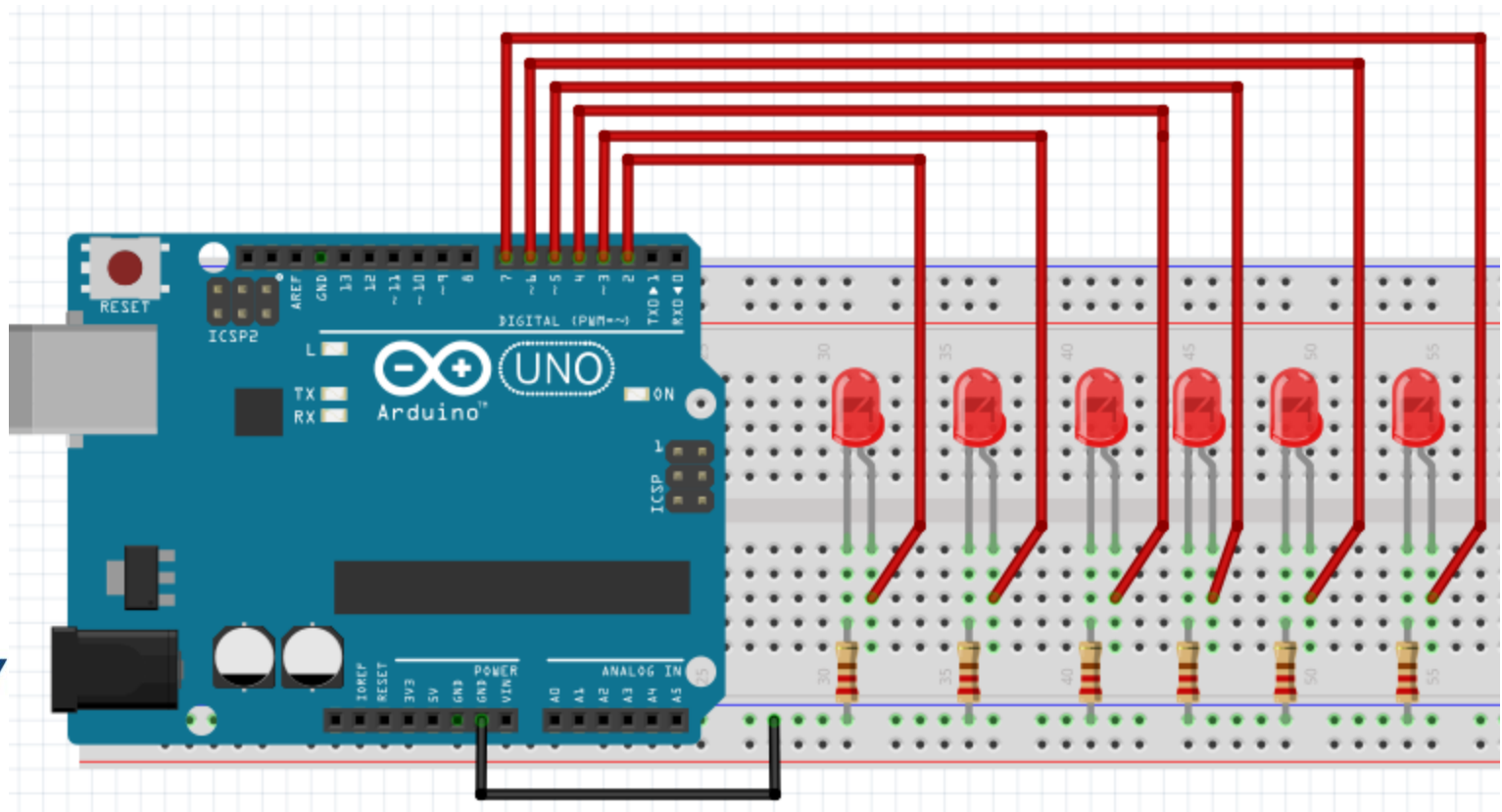
# EJERCICIO DE APLICACIÓN

**Encender un led y simular una salida de tipo PWM generando un número aleatorio entre 0 y 255 para regular el brillo del led.**



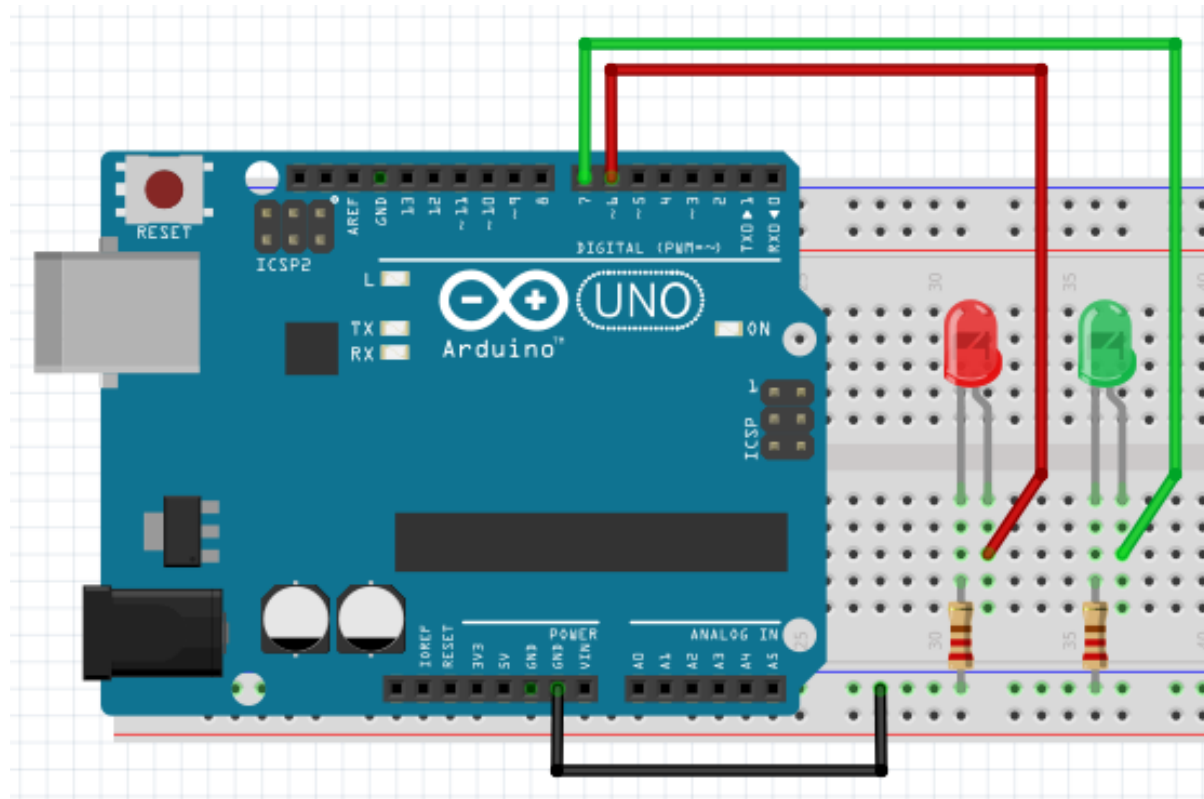
# EJERCICIO DE APLICACIÓN

Realizar un dado electrónico; la secuencia tiene que ser distinta al reiniciarse el programa y debe encender la cantidad de led's que salió en el numero aleatorio por 1 segundo y medio; es decir si salió el número aleatorio 2 deben encenderse 2 led's.



# EJERCICIO DE APLICACIÓN

**Encender 2 leds usando millis(), el primer led debe encenderse y apagarse a razón de un segundo, el segundo led debe encenderse a razón de 500 milisegundos.**



# EJERCICIO DE APLICACIÓN

Encender 3 led's simultáneamente, uno debe encenderse a razón de 500 milisegundos, el segundo a razón de 800 milisegundos y el tercero a razón de un segundo.

