

Arduino no posee un sistema operativo y tampoco incorpora ningún otro sistema que gestione la multitarea como estamos acostumbrados en otros entornos de programación (por ejemplo en Scratch), el programa se ejecuta directamente por el microcontrolador y somos nosotros en el propio programa los que tenemos que gestionar la manera de ejecutar varias tareas simultaneamente intentando simular un sistema multitarea.

La teoría de un sistema multitarea es dividir los procesos en pequeños bloques, cada uno de estos bloques debe realizar un pequeño trabajo (en poco tiempo) y dejar paso al siguiente sin bloquear el funcionamiento. Al realizar este proceso de forma continua y a mucha velocida el resultado final es que todos los bloques se ejecutan simultáneamente.

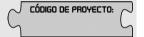
¿Qué debemos evitar dentro de los bloques de las tareas?

- Bloques de tipo "esperar"
- Bucles o repeticiones muy largas
- Condiciones que bloqueen la ejecución de la tarea

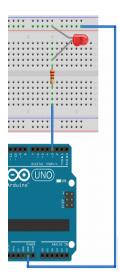
Para implementar un sistema sencillo de tareas utilizaremos el bloque "tiempo transcurrido". Este bloque nos permite obtener el tiempo en ms que ha pasado desde el inicio del programa (reset).

Tiempo transcurrido (milisegundos)

## **PRÁCTICA 15.1** Tarea parpadeo led cada 1s



El programa comprobará si han pasado 1000 ms (o más) desde la última ejecución de la tarea, si es así ejecutará el código de encender/apagar el led.

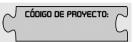


```
establecer led encendido v a lo establecer ultimo tiempo v a l'Tiempo transcurrido (milisegundos)

Bucle establecer tiempo actual v a l'Tiempo transcurrido (milisegundos) establecer diferencia de tiempo v a l'tiempo actual v - v ultimo tiempo v establecer ultimo tiempo v a l'Tiempo transcurrido (milisegundos) hacer establecer ultimo tiempo v a l'Tiempo transcurrido (milisegundos) establecer ultimo tiempo v a l'Tiempo transcurrido (milisegundos) establecer led encendido v a l' l'Escribir digital Pin 3 v ON v sino establecer led encendido v a l' l'Escribir digital Pin 3 v OFF v
```

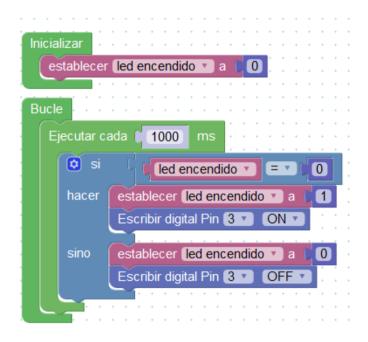


## **PRÁCTICA 15.2** Bloque de tareas simplificado

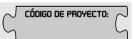


Todo el programa anterior de la práctica 15.1 se puede simplificar con un bloque que incorpora ArduinoBlocks llamado "*ejecutar cada*", donde le especificamos cada cuanto tiempo se deben ejecutar los bloques contenidos en su interior.

Programa equivalente a la práctica 15.1

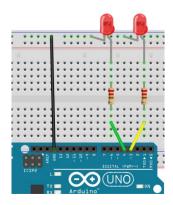


## PRÁCTICA 15.3 Dos tareas a la vez



Utilizando bloques "ejectura cada" vamos a realizar dos tareas simultáneas

- Tarea 1 (cada 500ms): hará parpadear un led conectado al pin 3 cada 500 ms
- Tarea 2 (cada 3000ms): hará parpadear un led conectado al pin 4 cada 3000 ms



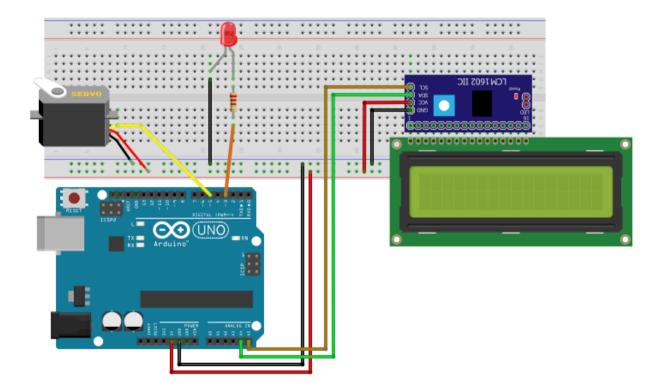


```
Establecer [led1]
                            Off
   Establecer led2
                                                           para (tarea1)
Bucle
                                                           Escribir digital Pin
                                                                                        led1
                                                           Establecer led1 •
                                                                                           led1
      tarea1
                                                        para (tarea2)
   Ejecutar cada
                    3000
                                                           Escribir digital Pin
                                                                                        led2 ▼
     tarea2
                                                           Establecer led2
                                                                                          led2
```

## **PRÁCTICA 15.4** Tres tareas a la vez

CÓDIGO DE PROYECTO:

Conecta la pantalla LCD con el módulo i2c, un led al pin 3 y un servo al pin 5.





El programa debe realizar 3 tareas:

- Tarea 1: Cada 250 ms parpadea el led
- Tarea 2: Cada 500 ms el servo se mueve 30 grados
- Tarea 3: Cada 1000 ms en la pantalla se muestra el valor de un variable que va aumentando

```
para tarea1
                               2x16 - ADDR 0x27 -
                                                             🧔 si
                                                                          led encendido 🔻 = 🔻 🖟 0
                                                                    establecer led encendido 🕝 a 🕌 🚺
estable cer (led encendido 🔻 a
                                                                    Escribir digital Pin 3 v ON v
estable cer posicion servo 🔻 a
estable cer contador v a 0
                                                                     establecer (led encendido 🕝 a 📲 🔾 🔾
                                                                    Escribir digital Pin 3 v OFF v
Ejecutar cada 250 ms
                                                           para (tarea2)
 tarea1
                                                                                                          Retardo (ms) 🚺 0
                                                                    Pin 5 T Grados posicion servo T
Ejecutar cada 500 ms
                                                             establecer posicion servo 🔻 a 🜘
                                                                                           posicion servo + 1 (10)
 tarea2
                                                                          posicion servo - > 180
Ejecutar cada 1000 ms
                                                                    establecer posicion servo - a
 tarea3
                                                           🔯 🕜 para (tarea3)
                                                             LCD limpiar
                                                             LCD imprimir Columna 0 T Fila 0 T
                                                                                                 " (Tarea 3: ))
                                                             LCD imprimir Columna 0 🔻 Fila 1 🔻
                                                                                                Número entero (contador -
                                                              establecer contador 🔻 a 🌗
                                                                                      contador - + 1 (1)
```

(prueba a modificar el tiempo de ejecución de cada tarea y comprueba el funcionamieno)