# Aplicación: Implementación de un Sistema RTO

# Equipo 3

### Abad Dolores Lázaro (Desarrollo de código y prueba física)

### Rodríguez Hernández Erick Abimael (Desarrollo de código y Prueba física)

# Introducción

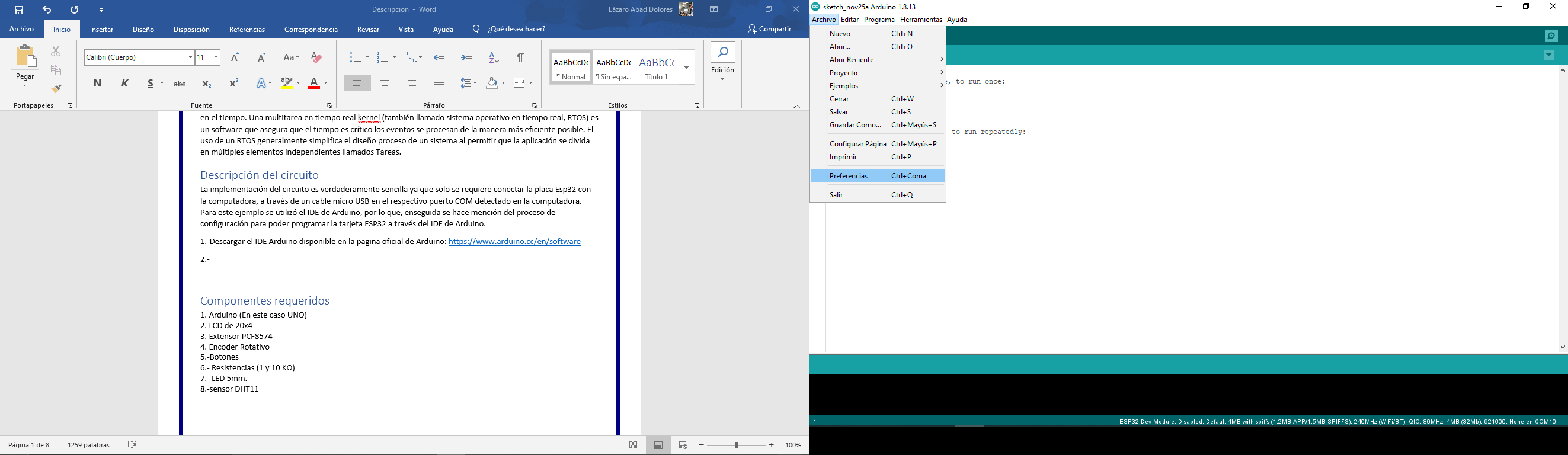
# Los sistemas en tiempo real se caracterizan por el hecho de que pueden producirse graves consecuencias si son lógicas y No se cumplen las propiedades de corrección de la sincronización del sistema. Una aplicación multitarea en tiempo real es un sistema en el que se deben procesar simultáneamente varias actividades críticas en el tiempo. Una multitarea en tiempo real kernel (también llamado sistema operativo en tiempo real, RTOS) es un software que asegura que el tiempo es crítico los eventos se procesan de la manera más eficiente posible. El uso de un RTOS generalmente simplifica el diseño proceso de un sistema al permitir que la aplicación se divida en múltiples elementos independientes llamados Tareas.

# Descripción del circuito

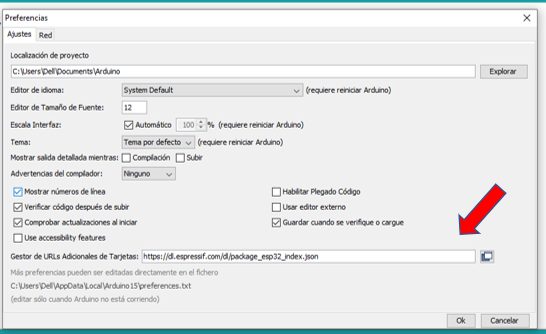
La implementación del circuito es verdaderamente sencilla ya que solo se requiere conectar la placa Esp32 con la computadora, a través de un cable micro USB en el respectivo puerto COM detectado en la computadora.  
Para este ejemplo se utilizó el IDE de Arduino, por lo que, enseguida se hace mención del proceso de configuración para poder programar la tarjeta ESP32 a través del IDE de Arduino.

1.-Descargar el IDE Arduino disponible en la pagina oficial de Arduino: <https://www.arduino.cc/en/software>

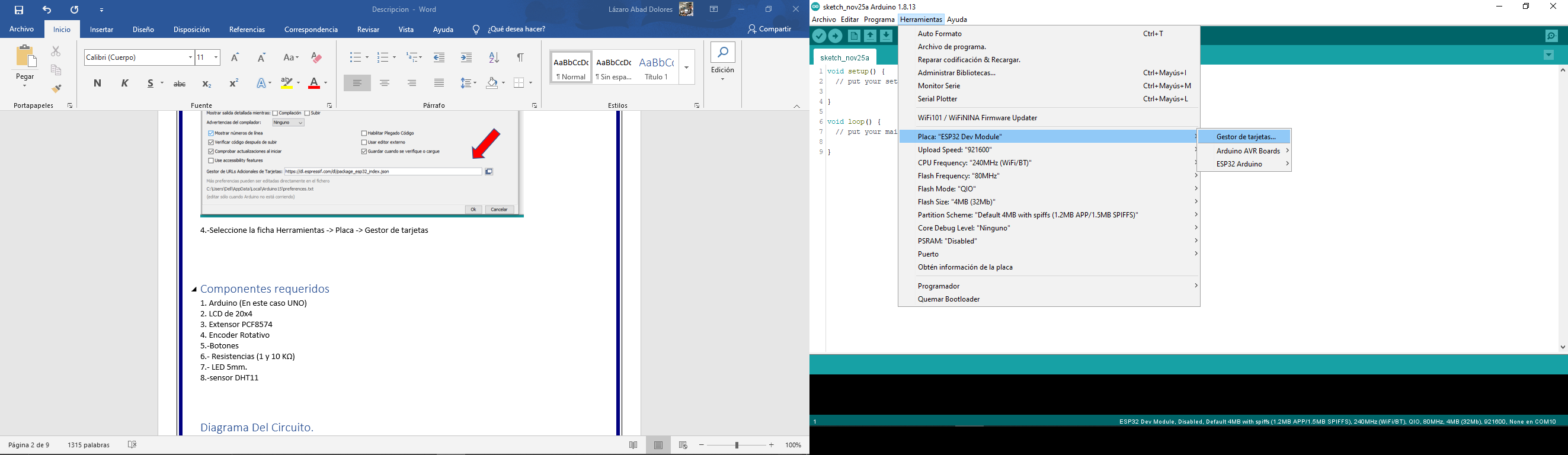
2.- Una ves en el entorno arduino:  
 Seleccione la ficha de **Archivo** -> **Preferencias**



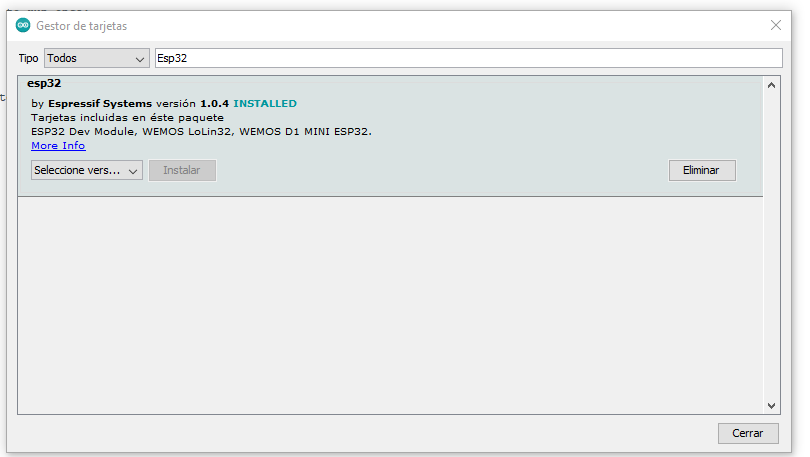
3.-AL presentarle la nueva ventana introduzca la siguiente liga: <https://dl.espressif.com/dl/package_esp32_index.json> en el espacio dispuesto como gestor de URL, es caso de haber uno existente, puede separarlo por una coma “,”, cierre la ventana.



4.-Seleccione la ficha Herramientas -> Placa -> Gestor de tarjetas



En la ventana del gestor de tarjetas el usuario debe ingresar el nombre de la placa, en este caso será **ESP32,** automáticamente se mostrarán las opciones disponibles, debe proceder a instalar esp32 o similar con la versión más reciente.



Por último, cierre la ventana.

5.- Seleccione nuevamente el fichero de **Herramientas -> Placa ->ESP32 Arduino**

Seleccione el modelo de placa de su modulo, Puede probar con: **ESP32 Dev Module**.

Terminados los pasos anteriores puede comenzar a programar.

# Componentes requeridos

1. Placa ESP32

2. Cable micro USB

# Diagrama Del Circuito.



Programación y estructuración de código

La ejecución y descripción del código se encuentra en el mismo, disponible en la carpeta con el nombre Codigo\_RTO localizada en el folder principal del proyecto.  
Su funcionamiento de reduce en la ejecución de 8 tareas, mismas que varían entre si sus parámetros principales:

* Nombre de tarea
* Tamaño del Stack o Pila
* Si recibe parámetros
* Prioridad de cada tarea
* Frecuencia de ejecución de tareas
* Envió de información entre tareas

En forma de descripción general cada tarea realiza lo siguiente:

Tarea 01: Se ejecuta 10 veces mandando un mensaje que evidencia que se esta ejecutando, cuando se realiza la séptima iteración manda información a la tarea 2, realizado con una estructura de switch (){}. Cuando concluye la tarea manda un mensaje advirtiendo que la tarea 1 ha terminado.

Tarea 02: Se ejecuta 10 veces mandando un mensaje que evidencia que se está ejecutando. Cuando concluye la tarea manda un mensaje advirtiendo que la tarea 2 ha terminado.

Tarea 03: Se ejecuta 7 veces mandando un mensaje que evidencia que se está ejecutando, cuando se realiza la quinta iteración manda información a la tarea 4, realizado con una estructura de for (){}. Cuando concluye la tarea manda un mensaje advirtiendo que la tarea 3 ha terminado. Cambia el valor de la variable global “c”.

Tarea 04: Se ejecuta 6 veces mandando un mensaje que evidencia que se está ejecutando, cuando la variable global “c” es igual 1 la tarea 04 recibe información de la tarea 08, realizado con una estructura de for (){}. Cuando concluye la tarea manda un mensaje advirtiendo que la tarea 04 ha terminado.

Tarea 05: Se ejecuta 5 veces mandando un mensaje que evidencia que se está ejecutando, realizado con una estructura de for (){}. Cuando concluye la tarea manda un mensaje advirtiendo que la tarea 05 ha terminado.

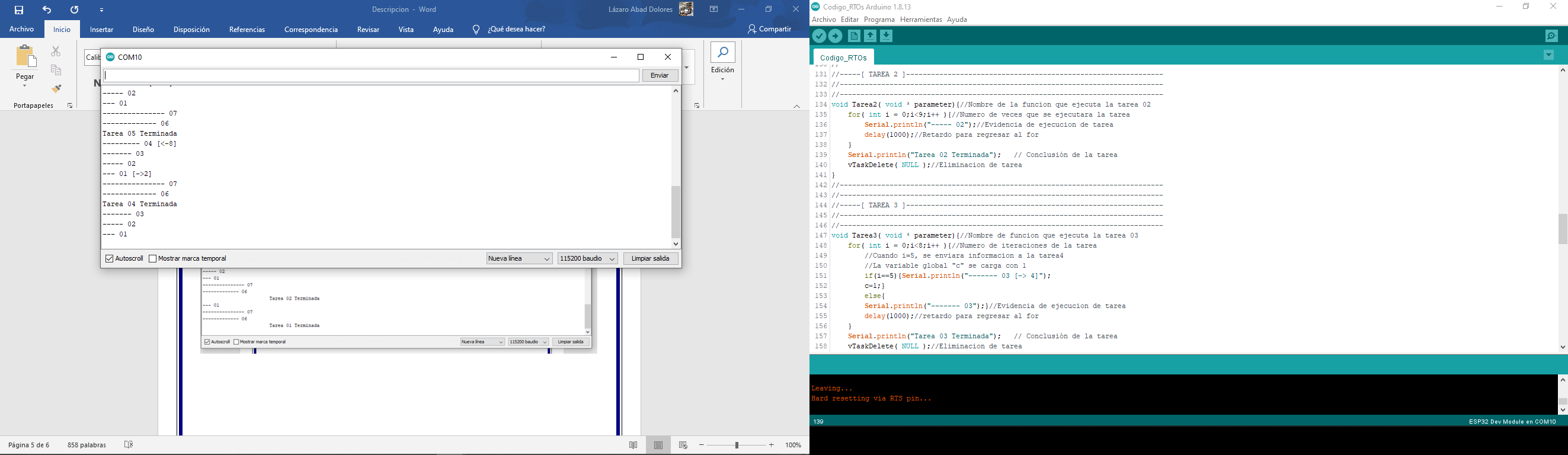
Tarea 06: Se ejecuta de manera infinita mandando un mensaje que evidencia que se está ejecutando, realizado con una estructura de While (){}. Cuando concluye la tarea manda un mensaje advirtiendo que la tarea 06 ha terminado.

Tarea 07: Se ejecuta de manera infinita mandando un mensaje que evidencia que se está ejecutando, realizado con una estructura de While (){}. Cuando concluye la tarea manda un mensaje advirtiendo que la tarea 07 ha terminado.

Tarea 08: Se ejecuta 2 veces mandando un mensaje que evidencia que se está ejecutando. Cuando concluye la tarea manda un mensaje advirtiendo que la tarea 2 ha terminado.

Funcionamiento en físico

El Funcionamiento efectivo del programa lo podemos verificar a través del monitor serial, en él podremos observar los mensajes de evidencia que muestran que tareas se están ejecutando, la prioridad que tiene una sobre otra, la frecuencia de ejecución de tareas así como cuando una tarea finalice.



# Conclusión

Los sistemas de tiempo real (STR) juegan un papel muy importante ya que como todo sistema debe tener la capacidad de satisfacer tareas en un tiempo de respuesta mínimo.

Los STR están presentes en todos los aspectos de nuestra vida diaria como aeronaves, teléfonos móviles, automóviles, entre muchas otras aplicaciones que nos benefician ampliamente.

Se podría también concluir que los sistemas de tiempo real están en constante desarrollo puesto que con los [avances tecnológicos](https://www.monografias.com/trabajos5/cienteysoc/cienteysoc.shtml) se construyen nuevas máquinas que operan en tiempo real y que por consiguiente necesitan tener sistemas controlados por un [computador](https://www.monografias.com/trabajos15/computadoras/computadoras.shtml) que tenga la capacidad de interactuar con el mundo físico.