

## Sistemas operativos de tiempo real - 2025

### Práctica N°1 (Versión 2)

#### Creación y borrado de tareas - Prioridades

Todo el código debe entregarse. Puede compartirse en github (usuario del docente: **hernangmol**) o enviarse por correo electrónico ([her.gomez@bue.edu.ar](mailto:her.gomez@bue.edu.ar)) en una carpeta comprimida con toda la práctica completa. Identificar claramente el correo (por ejemplo: **RTOS 2025 - Práctica 1 - Hernán Gomez Molino**)

En cada ejercicio se indican en letra **negrita** los puntos que deben entregarse obligatoriamente además del código,. Las capturas de pantalla (salidas de programa) deben tener incluida la identificación del alumno y del ejercicio. La entrega debe hacerse en un documento en formato pdf donde se pegan las capturas y se responden las preguntas. Este documento debe estar incluido en la carpeta, si se responde por correo o puede compartirse como google docs si se entrega por github.

Fecha de entrega: **07/10/2025**

---

#### 1) Creación de tareas (desde main):

- a) Crear un clon del proyecto Hola\_mundo generado en las actividades previas.
  - b) Reemplazar el código del archivo main.c por el de la Plantilla 1\_1.
  - c) Reemplazar el código del archivo FreeRTOSConfig.h por el de FreeRTOSConfig.h 1\_1. Analizar el código pegado.
  - d) Ejecutar y comprobar funcionamiento. Realizar correcciones de ser necesarias.
  - e) Modificar para crear otra tarea adicional igual a la primera pero que imprima un mensaje distinto.
  - f) Ejecutar y comprobar funcionamiento. **Hacer captura de pantalla de la salida del programa.**
  - g) **Explicar en qué consiste la corrección que debió realizarse**
- 

#### 2) Creación de tareas (desde tarea):

- a) Clonar el programa del ejercicio 1
- b) Modificar para que la tarea 2 se cree dentro de la tarea 1

- c) Ejecutar y comprobar funcionamiento. **Hacer captura de pantalla de la salida del programa.**
- 

3) Uso de los parámetros de tareas:

Debe notarse que en los ejercicios anteriores ambas tareas ejecutan casi el mismo proceso, a excepción del mensaje que imprime cada tarea. Se puede mejorar el código evitando escribir dos veces el mismo proceso. para ello deben utilizarse los parámetros de tarea.

- a) Clonar el ejercicio 1
  - b) Eliminar la función que implementa la tarea 2
  - c) Modificar el código para que ambas funciones se creen con una sola tarea, teniendo en cuenta que ambas tareas deben imprimir mensajes distintos (usar parámetros).
  - d) Ejecutar y comprobar funcionamiento. **Hacer captura de pantalla de la salida del programa.**
- 

4) Experimentando con prioridades:

- a) Basándose en el ejercicio anterior, modificar para que las dos tareas corran a distintas prioridades.
  - b) Ejecutar y comprobar funcionamiento. **Hacer captura de pantalla de la salida del programa.**
  - c) **Contestar las siguientes preguntas:**
    - . **Describe brevemente el funcionamiento del programa.**
    - . **Explique las causas de este funcionamiento.**
    - . **Según su criterio ¿Esta forma de funcionar podría tener utilidad?**
- 

5) Cambiando prioridades:

- a) Generar un clon del programa del ejercicio anterior
- b) Modificar para que ambas tareas funcionen (*pista*: utilizar alguna función de la API de FreeRTOS no utilizada hasta acá).

- c) Utilizando algún parámetro global (`#define`) asegurar que las tareas se ejecutan del siguiente modo:
- . Tarea 1 > 80% del tiempo, Tarea 2 > 20% del tiempo
  - . Tarea 1 > 50% del tiempo, Tarea 2 > 50% del tiempo
  - . Tarea 1 > 30% del tiempo, Tarea 2 > 70% del tiempo
- d) Ejecutar y comprobar funcionamiento. **Hacer captura de pantalla de la salida del programa para cada opción.**
- 

6) Borrando tareas:

- a) Basándose en el ejercicio anterior (versión 80% - 20%), hacer que la tarea 2 se ejecute un número determinado de veces (parámetro de programa) y luego se borre.
- b) Ejecutar y comprobar funcionamiento. **Hacer captura de pantalla de la salida del programa.**