

# Sistemas Operativos de Tiempo Real

## Trabajo práctico N° 4

Sistemas Operativos II  
DC - FCEFYN - UNC

Junio de 2019

## 1 Introducción

Toda aplicación de ingeniería que posea requerimientos rigurosos de tiempo, y que esté controlado por un sistema de computación, utiliza un Sistema Operativo de Tiempo Real (RTOS, por sus siglas en inglés). Una de las características principales de este tipo de SO, es su capacidad de poseer un kernel *preemptive* y un *scheduler* altamente configurable. Numerosas aplicaciones utilizan este tipo de sistemas tales como aviónica, radares, satélites, etc. lo que genera un gran interés del mercado por ingenieros especializados en esta área.

## 2 Objetivo

El objetivo del presente trabajo práctico es que el estudiante sea capaz de diseñar, crear, comprobar y validar una aplicación de tiempo real sobre un RTOS.

## 3 Desarrollo

Se pide que, sobre un sistema embebido de arquitectura compatible con *FreeRTOS*[1] (ej: ARM Cortex M4):

1. Se instale y configure FreeRTOS en el sistema embebido seleccionado.
2. Crear un programa con dos tareas simples (productor/consumidor) y realizar un análisis completo del Sistema con Tracealyzer (tiempos de ejecución, memoria) [2].
3. Diseñe e implemente una aplicación que posea dos productores y un consumidor. El primero de los productores es una tarea que genera strings de caracteres de longitud variable (ingreso de comandos por teclado). La segunda tarea, es un valor numérico de longitud fija, proveniente del sensor de temperatura del embebido. También que la primer tarea es aperiódica y la segunda periódica definida por el diseñador.

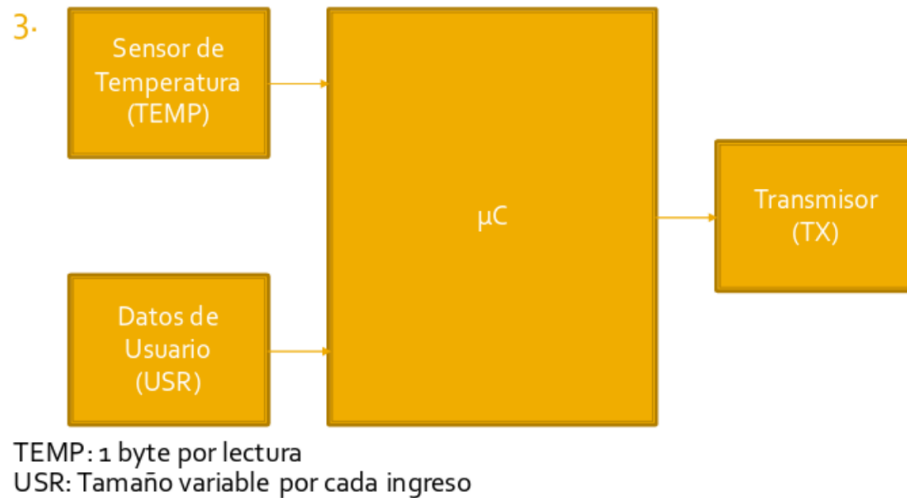


Figure 1: Arquitectura por componentes del sistema a implementar.

Por otro lado, el consumidor, es una tarea que envía el valor recibido a la terminal de una computadora por puerto serie (UART). La arquitectura por componentes del sistema se muestra en la Figura 1.

Y nuevamente, realizar un análisis del sistema con Tracealyzer.

## 4 Entrega

1. Informe con el esquema dado en clase, que incluya una guía al estilo “how to” de cómo se realizó el trabajo (paso por paso), y explicando cada script y programa implementado y el análisis realizado con el Tracealyzer.
2. Todo código de fuente desarrollado, el proyecto, con cualquier instructivo extra que crea necesario, Makefile, documentación, etc.

## 5 Evaluación

El presente trabajo práctico es individual deberá entregarse antes del viernes 22 de junio de 2019 a las 17:00 mediante el LEV. Será corregido y luego deberá coordinar una fecha para la defensa oral del mismo.

## 6 Referencias y ayudas

1. <https://www.freertos.org/>
2. <https://percepio.com/tracealyzer/>