

EX-2025-00881070- -UNC-ME#FAMAF

## ANEXO

<b>PROGRAMA DE ASIGNATURA</b>	
<b>ASIGNATURA:</b> Sistemas de Tiempo Real	<b>AÑO:</b> 2026
<b>CARÁCTER:</b> Obligatoria	<b>UBICACIÓN EN LA CARRERA:</b> 5° año 1° cuatrimestre
<b>CARRERA:</b> Licenciatura en Ciencias de la Computación <b>PLAN:</b> 2002	
<b>RÉGIMEN:</b> Cuatrimestral	<b>CARGA HORARIA:</b> 120 horas

### **FUNDAMENTOS Y OBJETIVOS**

En el actual paradigma tecnológico, los sistemas embebidos —sistemas computacionales de propósito específico— han trascendido su rol secundario para erigirse como el centro neurálgico de innovaciones como el Internet de las Cosas (IoT), el Edge Computing y la Industria 4.0. Son el motor de aplicaciones tan diversas como la robótica, la automatización de procesos y los sistemas de misión crítica en la automoción, la energía y la medicina. Estos sistemas ya no son simples controladores secuenciales, sino nodos de procesamiento sofisticados que deben gestionar múltiples flujos de datos, interactuar con diversos periféricos y comunicarse a través de redes, todo ello sujeto a estrictas restricciones temporales y de recursos. Esto ha elevado su complejidad a un nivel que desborda los modelos de programación tradicionales ("bare-metal") basado en bucles infinitos, demandando un enfoque más riguroso y sistemático donde la gestión de tareas concurrentes con restricciones temporales estrictas es la norma.

Este curso se posiciona precisamente en la intersección de la Ingeniería de Software, la Arquitectura de Computadoras y los Sistemas Operativos para abordar dicha complejidad. Se parte de la premisa de que la gestión de la concurrencia, la sincronización y, fundamentalmente, la predictibilidad temporal, no constituyen meros desafíos de implementación, sino que son el objeto de estudio de una disciplina formal dentro de las Ciencias de la Computación: los Sistemas de Tiempo Real. Con este fin, se proporcionan al estudiante los fundamentos teóricos para analizar el comportamiento de sistemas con restricciones temporales, estudiando los problemas clásicos de la concurrencia (inversión de prioridad, deadlocks) y los algoritmos de planificación que los resuelven.

Como herramienta práctica para materializar estos conceptos, el curso se enfoca en el uso de un Sistema Operativo de Tiempo Real (RTOS), específicamente FreeRTOS, el estándar de-facto en la industria. Al implementar soluciones sobre hardware real, el estudiante no se limita al uso instrumental de una tecnología de alta demanda laboral, sino que avanza hacia una conexión real entre la teoría y la práctica. FreeRTOS se convierte así en el entorno para experimentar con algoritmos de planificación, medir

EX-2025-00881070- -UNC-ME#FAMAF

latencias y validar el comportamiento temporal de un sistema, habilidades indispensables para el diseño de sistemas embebidos robustos, escalables y mantenibles.

#### Objetivo General

Desarrollar en el estudiante la capacidad de diseñar, implementar y validar sistemas embebidos con restricciones de tiempo real, aplicando principios teóricos de sistemas operativos y concurrencia para la creación de soluciones robustas y eficientes.

#### Objetivos Específicos

- Analizar los principios de la programación concurrente y los mecanismos de un RTOS (tareas, semáforos, mutexes, colas) para la gestión de recursos y la sincronización en sistemas de tiempo real.
- Aplicar la API de un RTOS para codificar y depurar aplicaciones multitarea sobre hardware embebido, gestionando la ejecución concurrente de tareas y el manejo de interrupciones.

### CONTENIDO

#### **Unidad I: Fundamentos de los Sistemas de Tiempo Real**

Introducción a los Sistemas Embebidos. Limitaciones del enfoque "super-loop". Conceptos de Tiempo Real: Tareas, eventos, deadlines, latencia, jitter. Clasificación: Sistemas Hard, Firm y Soft Real-Time. Arquitectura de un RTOS (Kernel). Servicios del sistema. Recursos de hardware asociados. Introducción a FreeRTOS: Filosofía, arquitectura y licenciamiento.

#### **Unidad II: Gestión de Tareas**

El concepto de Tarea (Task). Estados de una tarea: Running, Ready, Blocked, Suspended. Creación y eliminación de tareas. Parámetros: prioridad, stack. El Planificador (Scheduler): Algoritmos de planificación. Preemptive vs. Cooperative scheduling. El "Context Switch". La tarea "Idle" y "hooks". Gestión de memoria heap: asignación dinámica de memoria

#### **Unidad III: Sincronización y Comunicación entre Tareas**

El problema de los recursos compartidos. Secciones críticas y condiciones de carrera. Mutex para exclusión mutua. El problema de la inversión de prioridad. Semáforos: Binarios (para sincronización) y contadores (para gestión de recursos). Colas de Mensajes (Queue) para la comunicación segura de datos entre tareas. Timers por software, notificaciones de tareas y grupos de eventos. Delays

#### **Unidad IV: Gestión de Interrupciones y Recursos**

Interrupciones en un entorno RTOS. Prioridades de interrupción vs. prioridades de tareas. Funciones de API seguras para ISR. Estrategias de manejo: Procesamiento diferido de interrupciones. Gestión de memoria en RTOS: esquemas de asignación

EX-2025-00881070- -UNC-ME#FAMAF

heap. Modos de bajo consumo y el "Tickless Idle Mode". Gestion de recursos: secciones de código críticas, suspensión del planificador.

### **Unidad V: Diseño y Análisis de Sistemas**

Patrones de diseño comunes en sistemas de tiempo real. Técnicas de depuración para sistemas concurrentes. Herramientas de visualización y trazado. Introducción a la configuración y porting de un RTOS. Interfaces estándares de periféricos

### **BIBLIOGRAFÍA BÁSICA**

- "Mastering the FreeRTOS™ Real Time Kernel: A Hands-On Tutorial Guide", Richard Barry. Release Version - 1.1.0, Real Time Engineers Ltd, (2024).
- "Embedded System Design with Arm Cortex-M Microcontrollers: Applications with C, C++ and MicroPython", Cem Ünsalan, Hüseyin Deniz Gürhan, Mehmet Erkin Yücel. Ed. Springer Nature Switzerland AG (2022).

### **BIBLIOGRAFÍA COMPLEMENTARIA**

- "Reference Manual for FreeRTOS version 10.0.0 issue 1". Amazon Inc., (2017).
- "Electronics for Embedded Systems", Ahmet Bindal, Ed. Springer International Publishing, Switzerland, (2017).

### **FORMAS DE EVALUACIÓN**

La evaluación de la asignatura se centrará en la realización de un proyecto integrador individual. Este consistirá en el desarrollo de un sistema de tiempo real de mediana complejidad, implementado sobre una plataforma embebida para dar solución a un problema de aplicación práctica.

El alcance específico del proyecto y los hitos de avance serán acordados entre cada estudiante y el docente al inicio del cursado.

1. Evaluaciones Parciales (2): Se realizarán dos instancias de seguimiento que funcionarán como exámenes parciales. Cada una requerirá:

- Una demostración funcional en laboratorio del avance logrado.
- La entrega de un informe técnico que documente el trabajo realizado.

2. Exmen Final: Consistirá en la defensa individual del proyecto integrador ya concluido. En esta instancia se evaluará tanto el informe escrito final como la exposición y defensa oral del trabajo desarrollado.

En la evaluación se considerará los siguientes aspectos:

- Funcionalidad y Robustez: Cumplimiento de todos los requerimientos especificados.
- Calidad del Diseño de Software: Modularidad, claridad y correcta aplicación de los primitivos del RTOS.
- Documentación Técnica: Informe final que detalle la arquitectura de software, las decisiones de diseño y los resultados de las pruebas.
- Defensa Oral: Presentación y demostración funcional del proyecto, con preguntas por

EX-2025-00881070- -UNC-ME#FAMAF

parte del docente.

## REGULARIDAD

Para obtener la condición de alumno regular, el estudiante deberá:

1. Cumplir un mínimo de 70% de asistencia a clase teóricas, prácticas, o de laboratorio.
2. Aprobar al menos dos evaluaciones parciales o sus correspondientes recuperatorios.

## PROMOCIÓN

Para obtener la condición de alumno promocional, el estudiante deberá:

1. cumplir un mínimo de 80% de asistencia a clases teóricas, prácticas, o de laboratorio.
2. aprobar todas las evaluaciones parciales con una nota no menor a 6 (seis), y obteniendo un promedio no menor a 7 (siete).
3. aprobar todos los Trabajos Prácticos o de Laboratorio, o el Informe Final de la Práctica de la Enseñanza con una nota no menor a 6 (seis).

## CORRELATIVIDADES

Para Cursar:

- Organización del Computador
- Sistemas Operativos

Para Rendir: (aprobadas)

- Organización del Computador
- Sistemas Operativos