

**Carrera:** Especialización en Sistemas Embebidos

**Curso de Posgrado:** Sistemas operativos de Tiempo Real

**Carga Horaria** <sup>1</sup>: 45

**Docente/s a cargo:**

**Esp. Ing. Esteban Volentini (FACET UNT)**

**Mgt Bioing. Juan Manuel Reta (FIUNER)**

**Semestre: 2**

**Año: 2021**

1:

Carga horaria: la cantidad de horas reloj.

2:

Curso teórico: curso donde se desarrolla en forma expositiva una temática propia de la disciplina

Curso teórico-práctico: curso que articula la modalidad del curso teórico con una actividad de la práctica con relación a la temática de estudio. Lo teórico y lo práctico se dan simultáneamente en forma interrelacionada.

3:

Carácter: si son obligatorios u optativos.

Programa Analítico: de foja: 1 a foja: 2

Bibliografía de foja: 2 a foja: 3

### **Programa Analítico**

#### Contenidos del Curso:

Conceptos de Sistemas Operativos: Procesos. Estado de los Procesos. Cambio de Contexto. Multitarea cooperativa y expropiativa. Algoritmos de planificación del procesador. Prioridades. Sistemas Operativos de Tiempo Real. Sistemas Operativos Estáticos y Dinámicos. Temporización: Espera pasiva. Alarmas. Suspensión de la ejecución. Comunicación entre procesos: Problema de la sección crítica. Sincronización. Métodos de mutua exclusión. Abrazo Mortal. Herencia de Prioridad. Eventos. Control de los dispositivos: Capa de abstracción de hardware. Gestión de las interrupciones. Pruebas: Diferentes tipos de pruebas. Pruebas unitarias. Funciones de reemplazo y espías. Pruebas de integración. Pruebas de sistema. Desarrollo guiado por pruebas.

#### Estructura del Curso:

#### **Clases Teóricas**

Teoría 1: Sistemas Operativos de Tiempo Real para Embebidos. → Volentini - Reta

Teoría 2: Cambio de contexto.→Volentini

Teoría 3: Temporización en RTOS → Volentini - Reta

Teoría 4: Sincronización de tareas → Mutex. - Volentini - Reta

Teoría 5: Gestión de Interrupciones→Volentini

Teoría 6: Comunicación entre Tareas → Volentini - Reta

Teoría 7: Diseño de Aplicaciones→ Volentini - Reta

### **Prácticas propuestas:**

Práctica 1: Generación de Tareas y Cambio de Contexto

Práctica 2: Temporización y Sincronización de Tareas

Práctica 3: Interrupciones y Comunicación entre Tareas

## **BIBLIOGRAFÍA**

### **Bibliografía:**

- [1] James W. Grenning. Test-driven Development for Embedded C. Pragmatic Bookshelf Series. Pragmatic Books helf, 2011. isbn: 9781934356623. url: <https://books.google.com.ar/books?id=QuUBRQAACAAJ>.
- [2] Qing Li y Caroline Yao. Real-Time Concepts for Embedded Systems. 1st. Boca Raton, FL, USA: CRC Press, Inc., 2003. isbn: 1578201241, 9781578201242.
- [3] Rupak Majumdar y Bjorn Brandenbug. Foundations of Ciber-Physical Systems. 2014. url: <http://rts.mpisws.org/classes/fall14/foundations-cps/>.
- [4] 2005 OSEK/VDX Steering committee. OSEK-VDX Operating System - Specification 2.2.3. url: <http://portal.osek-vdx.org/files/pdf/specs/os223.pdf>.
- [5] 2005 OSEK/VDX Steering committee. OSEK-VDX System Generation - OIL: OSEK Implementation Lenguaje-Version 2.5 2.2.3. url: <http://portal.osek-vdx.org/files/pdf/specs/oil25.pdf>.
- [6] Ragunathan (Raj) Rajkumar y col. "Cyber-physical Systems: The Next Computing Revolution". En: DAC '10 (2010), pags. 731-736. doi: 10.1145/1837274.1837461. url: <http://doi.acm.org/10.1145/1837274.1837461>.
- [7] Abraham Silberschatz, Peter B. Galvin y Greg Gagne. Operating System Concepts. 9th. Wiley Publishing, 2012. isbn: 1118063333, 9781118063330.
- [8] Lothar Thiele. Lecture: Embedded Systems (FS 2017). 2017. url: <http://www.tik.ee.ethz.ch/education/lectures/ES/>.
- [9] Mark VanderVoord, Mike Karlesky y Greg Williams. Throw The Switch: C Code that doesn't suck. Url: <http://www.throwtheswitch.org>.

Bibliografía Disponible en Formato Digital

- Mariano Cerdeiro. Breve introducción a OSEK-VDX. 2014. url: [http://www.proyecto-ciaa.com.ar/devwiki/lib/exe/fetch.php?media=desarrollo:firmware:breve\\_introduccio\\_n\\_a\\_osek-vdx.pdf](http://www.proyecto-ciaa.com.ar/devwiki/lib/exe/fetch.php?media=desarrollo:firmware:breve_introduccio_n_a_osek-vdx.pdf).
- RETMS: Quick Start - <https://docs.rtems.org/branches/master/user/start/index.html>

Tutoriales de la plataforma Moodle para alumnos

- [Tutoriales para estudiantes](http://ingenieria.uner.edu.ar/distancia/index.php/recursos-para-estudiantes-y-docentes). Área de educación a distancia. Facultad de Ingeniería. Universidad Nacional de Entre Ríos. Disponible en: <http://ingenieria.uner.edu.ar/distancia/index.php/recursos-para-estudiantes-y-docentes>

## PLANIFICACIÓN DEL CURSO

### Objetivos Generales:

Que el alumno incorpore el concepto de Sistema Operativo en Tiempo Real y sus políticas de scheduling, comprenda las diferentes alternativas y las aplique con criterio a ejercicios de trabajos prácticos.

### Objetivos Particulares:

- Diseñar e implementar Sistemas Embebidos que emplean los servicios de un Sistema Operativo de Tiempo Real (RTOS).
- Discernir la diferencia entre RTOS dinámicos y estáticos, y en qué casos es conveniente la utilización de cada tipo.
- Resolver los problemas derivados de la interacción de tareas utilizando los servicios de un RTOS.
- Entender los motivos de las diferentes políticas de procesador que implementan los RTOS a diferencia de los Sistemas Operativos de propósito general.
- Conocer y utilizar herramientas de pruebas para mejorar la calidad del software desarrollado.

### Metodología de Trabajo:

Los materiales educativos del curso se encuentran en el aula virtual “Sistemas Operativos de Tiempo Real” alojada en el campus de la FIUNER (plataforma moodle).

El aula virtual está organizada por pestañas con la siguiente estructura:

**Pestaña de presentación:** La misma tiene información general del curso (programa, equipo docente, cronograma, objetivos, materiales utilizados a lo largo del curso, descripción de la metodología de trabajo y evaluación, bibliografía, foro de presentación, etc.)

### Pestaña de comunicación:

- Un **foro de novedades** destinado a informar novedades acerca del cursado, información general, publicación de actividades y materiales semanales, modificaciones en cronogramas, recordatorios.
- Un **foro de orientaciones y consultas** destinado a resolver consultas o dudas que surjan durante el cursado.

- Enlace a las **videoconferencias**.

**Pestaña de despedida** y encuesta de evaluación del curso.

**Pestañas de temas:** Cada semana se habilitará una pestaña de temas correspondiente a las unidades temáticas en curso. Las clases se diseñan para facilitar al estudiante los materiales y herramientas para que él mismo construya el conocimiento en forma continua y autónoma; donde se “aprenda haciendo” y los conocimientos nuevos se incorporan a los preexistentes. La misma incluye:

- La **hoja de ruta y orientaciones** para la realización de las actividades.
- El **material educativo** organizado en material obligatorio y complementario. Este material se publica en formatos diversos (documentos, libros, material audiovisual, material interactivo), para proveer distintas vías de acceso al conocimiento.
- Las **actividades colaborativas** que fomenten la interacción de los alumnos en el aula virtual. Los mismos se realizarán utilizando herramientas colaborativas que permitan al docente a realizar el seguimiento y evaluación del trabajo. Estas actividades serán supervisadas y retroalimentadas por el docente del curso.

Respecto de las clases:

Las clases prácticas son en modalidad hands-on (los alumnos tienen el material en sus hogares), utilizando placas EDU\_CIAA\_NXP con microcontrolador LPC4337JBD144 perteneciente a la familia ARMv7M y una pantalla TFT de 320x240. Esta modalidad, implica mucho tiempo de interacción con el hardware previsto para la realización de las actividades, que se completa con las consultas sincrónicas (encuentro virtual) y asincrónicas (foros y correo electrónico).

En las actividades prácticas se orientará al alumno en el diseño de soluciones para problemas de aplicación en Sistemas Embebidos, empleando herramientas de gestión de dependencias y documentación de código. Dichas soluciones se implementarán sobre placas de desarrollo. Se trabajará en el diseño e implementación de una aplicación que se usará como hilo conductor de todas las unidades temáticas: *Reloj Despertador Digital*. Se desarrollará la aplicación de manera incremental e iterativa incorporando recursos de sistemas operativos de tiempo real disponibles por Free RTOS en su porting para el firmware de la EDU-CIAA.

Se trabajará a partir de ejemplos de código que serán provistos por los docentes a través del campus. Los mismos deberán incorporarse a los repositorios de cada alumno/a e integrarse al desarrollo personal del firmware de la aplicación.

Las interacciones docentes-alumnos se realizarán de forma asíncrona a través del Foro de novedades y foro de consultas del aula virtual y en forma síncrona a través de las videoconferencias semanales utilizando la plataforma meet.

La interacción alumnos entre sí se realizará a través de herramientas de trabajo colaborativas y grupales para la resolución de los trabajos prácticos.

En el caso que sea necesario se realizarán interacciones entre el docente-alumno por mensajería privada o mail para resolución de dudas o consultas puntuales.

En cada semana, se proponen dos encuentros sincrónicos mediante videoconferencia con el objetivo siguiente:

**Teoría:** Con una duración total de 1:30 hs, realizada los días viernes de 18 a 19:30 hs.

Se realizarán exposiciones de temas, actividades de presentación de temas, interacción y discusión incluyendo análisis de aplicaciones y se presentará el material multimedia disponible en el campus.

Los encuentros de teoría quedarán grabados, y el enlace a la clase se colocará en el campus a disposición de los alumnos. De esta manera, no se obliga al alumno a conectarse sincrónicamente ya que tiene la opción de hacerlo a posteriori.

**Consulta:** En la semana siguiente a la teoría, el día martes de 18 a 19 hs.

Se realizarán actividades de consulta y discusión de resolución de los problemas y las actividades planteadas. Los encuentros de consulta quedarán grabados, y el enlace a la clase se colocará en el campus a disposición de los alumnos

#### Equipo docente:

- Ing. Esteban Daniel Volentini - evolentini@gmail.com, Profesor adjunto de **Sistemas con Microprocesador y Microcontroladores** y **Arquitectura de Computadoras** en la Facultad de Ciencias Exactas y Tecnología de la Universidad Nacional de Tucumán.
- Mg. Bioing. Juan Manuel Reta – juan.reta@uner.edu.ar, Profesor Titular Ordinario de **Electrónica Programable**, Prof. Adjunto a cargo de **Sistemas de Adquisición y Procesamiento de Señales** en la Facultad de Ingeniería de la Universidad Nacional de Entre Ríos.

#### Cronograma del Curso:

Semana	Contenido	Actividades	Tiempo estimado total (Encuentros sincrónicos más actividades asincrónicas)
Semana 1 Viernes 29/10 Martes 02/11	Sistemas Operativos de Tiempo Real para Embebidos.	Instalación de herramientas, prueba de compilación – Ejemplo de Prueba.	Total: 6 horas (3 horas sincrónicas + 3 horas asincrónicas)
Semana 2 Viernes 05/11 Martes 10/11	Cambio de contexto	Generación de Tareas y Cambio de Contexto – Parte I	Total: 6 horas (3 horas sincrónicas + 3 horas asincrónicas)
Semana 3 Viernes 12/11 Martes 17/11	Temporización	Generación de Tareas y Cambio de Contexto – Parte II	Total: 6 horas (3 horas sincrónicas + 3 horas asincrónicas)
Semana 4 Viernes 19/11 Martes 24/11	Sincronización de tareas	Temporización y Sincronización de Tareas – Parte I	Total: 6 horas (3 horas sincrónicas + 3 horas asincrónicas)
Semana 5 Viernes 03/12 Miércoles 09/12	Gestión de Interrupciones	Temporización y Sincronización de Tareas – Parte II	Total: 6 horas (3 horas sincrónicas + 3 horas asincrónicas)
Semana 6 Viernes 10/12 Martes 15/12	Comunicación entre Tareas	Interrupciones y Comunicación entre Tareas – Parte I	Total: 6 horas (3 horas sincrónicas + 3 horas asincrónicas)
Semana 7 Viernes 17/12	Diseño de Aplicaciones	Interrupciones y Comunicación entre Tareas – Parte II	Total: 6 horas (3 horas sincrónicas + 3 horas asincrónicas)

**Condiciones de Regularidad y Promoción:**

El curso se aprobará con la presentación y aprobación de cada una de las entregas preliminares de la aplicación planteada y la entrega final. La mismas deberán contar con los siguientes entregables:

Entregables para evaluación:

- Modelado del software (o de alguna parte del mismo) (Diagrama de flujo o máquinas de estado u otro)
- Repositorio con el código
- Video del funcionamiento

**Infraestructura necesaria:**

PC de escritorio o notebook con conexión a Internet, una placa EDU-CIAA con cable usb, pantalla LCD tipo TFT color de 320x240 y acceso al Campus virtual de la FIUNER