



Ministerio de Cultura y Educación  
 Universidad Nacional de San Luis  
 Facultad de Ciencias Físico Matemáticas y Naturales  
 Departamento: Informatica  
 Area: Area IV: Pr. y Met. de Des. del Soft.

(Programa del año 2025)

### I - Oferta Académica

Materia	Carrera	Plan	Año	Período
SISTEMAS DE TIEMPO REAL	ING. INFORM.	026/1	2- 2025	1° cuatrimestre
		08/15		

### II - Equipo Docente

Docente	Función	Cargo	Dedicación
SANCHEZ, ALEJANDRO	Prof. Responsable	P.Adj Exc	40 Hs
MIRANDA, ENRIQUE ALFREDO	Responsable de Práctico	JTP Simp	10 Hs

### III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
3 Hs	Hs	1 Hs	1 Hs	5 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
12/03/2025	24/06/2025	15	75

### IV - Fundamentación

Los Sistemas de Tiempo Real (STR) se encuentran en la vida cotidiana. Por ejemplo, en los sistemas de control de lavavajillas, reproductores de DVD, CD, y hasta en los automóviles con sus sistemas antibloqueo de frenos, control de tracción y climatizador. Los sistemas de control como estos analizan el medio en el cual están embebidos y actúan en fracciones de segundo. También se encuentran en lugares más críticos como los sistemas de navegación y posicionamiento de aviones, y en lugares aún más críticos como las centrales termonucleares.

La corrección de los STR es de suma importancia sin lugar a dudas. Una falla parcial o total puede acarrear consecuencias incómodas, en el mejor de los casos, o catastróficas, en los casos extremos.

El desarrollo de estos sistemas requiere conocer múltiples disciplinas. Involucra conocimiento que va desde Ingeniería de Software hasta Sistemas Operativos de Tiempo Real, pasando por Arquitectura del Computador y Lenguajes de Programación con abstracciones para el manejo de conceptos de tiempo real. En particular, para la verificación y prueba son precisos modelos de sistemas y distintos tipos de lógicas.

Esta disciplina se vuelve más importante conforme crece la demanda de recursos humanos altamente calificado para desempeñarse en proyectos de desarrollo de software donde las especificaciones temporales son un requerimiento

fundamental. Esta materia procura que los alumnos de Ingeniería en Informática adquieran los conocimientos y capacidades necesarias para desarrollarse en tales proyectos. El enfoque seguido es teórico-práctico y recurre a la utilización de herramientas tanto académicas como industriales.

## V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso se espera que la/el estudiante adquiera la capacidad de:

- O1. Identificar, formular y resolver problemas de tiempo real utilizando el enfoque del ingeniero, es decir, modelando, animando, analizando y verificando el sistema a construir.
- O2. Concebir, diseñar y desarrollar proyectos de sistemas de tiempo real.
- O3. Utilizar técnicas y herramientas de la ingeniería en informática en sistemas de tiempo real.
- O4. Contribuir a la generación de desarrollos tecnológicos de sistemas de tiempo real.
- O5. Desempeñarse de manera efectiva en equipos de trabajo.
- O6. Actuar con responsabilidad profesional y compromiso social, considerando el impacto social y ambiental de los sistemas de tiempo real en el contexto local y global.
- O7. Aprender de manera autónoma sobre problemáticas específicas de STR que refinan temas más generales ya vistos en la carrera.
- O8. Dominar la jerga utilizada en el desarrollo de STR.

Estos objetivos se encuentran alineados con los siguientes ejes transversales, en respectivo orden:

- E1. Identificación, formulación y resolución de problemas de ingeniería en informática.
- E2. Concepción, diseño y desarrollo de proyectos de ingeniería en informática.
- E3. Utilización de técnicas y herramientas de aplicación de la ingeniería en informática.
- E4. Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.
- E5. Fundamentos para el desempeño en equipos de trabajo.
- E6. Fundamentos para evaluar y actuar en relación con el impacto social de su actividad profesional en el contexto global y local.
- E7. Fundamentos para el aprendizaje continuo.
- E8. Fundamentos para una comunicación efectiva.

## VI - Contenidos

### **Unidad I: Conceptualización**

Caracterización: sistema y sistema de tiempo real. Casos de estudio. Tiempo de respuesta. Sistema de tiempo real defectuoso. Sistema embebido. Restricciones de tiempo y sus tipos. Tipos de sistemas de tiempo real según sus restricciones temporales. Eventos y tipos de eventos. Factor de utilización del CPU.

### **Unidad II: Hardware**

Arquitectura von Neumann. Memoria. CPU. Dispositivos periféricos. Microprocesadores y microcontroladores. Protocolos y arquitecturas para sistemas distribuidos de tiempo real.

### **Unidad III: Sistemas Operativos de tiempo real**

Conceptos preliminares. Kernels y sistemas operativos para aplicaciones de tiempo real. Fundamentos teóricos de la planificación de sistemas de tiempo real. Comunicación y sincronización entre tareas. Gestión de memoria. Selección del sistema operativo de tiempo real.

### **Unidad IV: Lenguajes de programación para sistemas de tiempo real**

Programación con restricciones temporales. Programación de software naturalmente concurrente. Exigencias de alta confiabilidad en la programación a bajo nivel y en la sincronización de tareas. Lenguajes assembler, procedural, orientado a objetos. Lenguajes de tiempo real específicos.

### **Unidad V: Ingeniería de requerimientos de tiempo real**

El proceso. Tipos de requerimientos. Especificación de software de tiempo real. Métodos formales. Métodos semi-formales.

El documento de requerimientos.

### **Unidad VI: Ingeniería de software de tiempo real**

Calidad de software. Principios de la ingeniería de software. Enfoques de diseño procedural. Enfoques de diseño orientado a objetos. Modelos de ciclo de vida.

### **Unidad VII: Análisis de desempeño**

Técnicas de análisis de desempeño. Aplicaciones de teorías de colas. Desempeño de entrada/salida. Análisis de requerimientos de memoria.

### **Unidad VIII: Confiabilidad y tolerancia a fallas**

Fiabilidad, fallas y defectos. Tipos de fallas. Prevención y tolerancia a fallas. Programación de N-versiones. Redundancia dinámica de software. El enfoque de bloque de recuperación. Redundancia dinámica y excepciones. Métricas y predicciones. Seguridad, fiabilidad y confiabilidad. Redundancia dinámica y fallas temporales. Detección de incumplimiento de tiempos. Sobrepasso del tiempo de ejecución en el peor caso. Sobrepasso de tiempo en eventos esporádicos. Sobrepasso en el uso de recursos. Confinamiento de daños. Recuperación de errores.

### **Unidad IX: Verificación de sistemas de tiempo real**

Fundamentos teóricos. Lógica temporal lineal. Lógica de árbol computacional. La herramienta NuSMV. Grafo temporizado. Redes de grafos temporizados. Lógica de árbol computacional temporizada. La herramienta UPPAAL. Verificación de sistemas de hardware, software y protocolos de comunicación.

## **VII - Plan de Trabajos Prácticos**

### Trabajos Prácticos

1. Conceptualización
2. Verificación de sistemas de tiempo discreto
3. Verificación de sistemas de tiempo continuo
4. Análisis de desempeño
5. Confiabilidad y tolerancia a fallas

### Trabajo grupal

Siguiendo la guía de estudio preparada por la cátedra, realizar informe y presentación a compañeros en uno de los siguientes temas específicos para STR:

- Hardware
- Sistemas Operativos de tiempo real
- Lenguajes de Programación para Sistemas de Tiempo Real
- Ingeniería de requerimientos de tiempo real
- Ingeniería de software de tiempo real

### Trabajos de laboratorio

- Verificación de sistemas de tiempo discreto
- Verificación de sistemas de tiempo continuo

A continuación se describe cómo se abordan y cómo se evalúan los ejes transversales trabajados en la asignatura:

E1. Identificación, formulación y resolución de problemas de ingeniería en informática.

¿Cómo se aborda?

Los prácticos del 2do al 5to presentan al estudiante problemas que típicamente emergen en la ingeniería de STR, más específicamente problemas de: concurrencia, coordinación, puntualidad, correctitud, desempeño, robustez y confiabilidad.

¿Cómo se evalúa?

Se realiza una evaluación (con sus respectivas recuperaciones) la cual permite acreditar los conocimientos.

E2. Concepción, diseño y desarrollo de proyectos de ingeniería en informática.

¿Cómo se aborda?

Las cuestiones específicas al desarrollo de proyectos de STR en el contexto de una ingeniería de software se enfocan a través de guías de estudio con preguntas que los alumnos deben responder en el Trabajo Grupal.

Por otro lado, los trabajos de laboratorio le presentan al alumno proyectos que deben resolver con la utilización de las herramientas para su modelación, análisis y verificación. Cabe destacar que estos modelos sirven de prototipos ya que pueden ejecutarse simulaciones a partir de ellos.

¿Cómo se evalúa?

Se evalúa tanto el informe como la presentación del trabajo grupal.

A su vez, durante las evaluaciones sumativas se requiere la prototipación de proyectos de STR.

E3. Utilización de técnicas y herramientas de aplicación de la ingeniería en informática.

¿Cómo se aborda?

Los prácticos 2 y 3 introducen las herramientas NuSMV y UPPAAL para la modelización, análisis y verificación de STR.

¿Cómo se evalúa?

Los estudiantes completan las evaluaciones, tanto parciales como los trabajos de laboratorio, utilizando las herramientas.

E4. Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.

¿Cómo se aborda?

Las herramientas utilizadas permiten el rápido prototipado de proyectos, facilitando la prueba y puesta a punto de ideas innovadoras, pudiendo verificar características de seguridad, vitalidad y temporales de las soluciones.

¿Cómo se evalúa?

Para aprobar la evaluación sumativa el estudiante debe acreditar dominio de las herramientas propuestas.

E5. Fundamentos para el desempeño en equipos de trabajo.

¿Cómo se aborda?

Este eje se aborda en el trabajo grupal, donde los estudiantes deben desarrollar las consignas en equipos de 3 o a lo sumo 4 personas.

¿Cómo se evalúa?

Se evalúa el informe entregado y su presentación.

E6. Fundamentos para evaluar y actuar en relación con el impacto social de su actividad profesional en el contexto global y local.

¿Cómo se aborda?

Al desarrollar prototipos de STR en los trabajos prácticos 2, 3 y 5, la/el estudiante debe considerar consecuencias en sociales el impacto ambiental y social de la actividad profesional en el contexto global y local.

¿Cómo se evalúa?

Las evaluaciones sumativas incorporan propiedades/características que implican un impacto social y ambiental del STR que la/el estudiante debe proyectar, validar y verificar.

E7. Fundamentos para el aprendizaje continuo.

¿Cómo se aborda?

El trabajo grupal requiere retomar temas ya vistos en la carrera y construir a partir un refinamiento de los mismos, contemplando características específicas de los STR.

¿Cómo se evalúa?

Tanto en el informe entregado, como en la presentación del tema, se evalúa la apropiación de los conceptos específicos de STR.

E8. Fundamentos para una comunicación efectiva.

¿Cómo se aborda?

Se aborda desde la expresión oral a partir de la participación de la/del estudiante en cada una de las actividades del curso, principalmente en el trabajo grupal y durante su presentación.

A su vez, también se considera la expresión escrita en el informe y filminas del trabajo grupal.

¿Cómo se evalúa?

Se evalúa tanto la presentación del trabajo grupal como el informe y filminas entregadas por las/los estudiantes.

## VIII - Regimen de Aprobación

Para regularizar el alumno debe:

- 1) Asistir a un 80% de las clases
- 2) Aprobar el trabajo grupal
- 3) Aprobar los trabajos de laboratorio
- 4) Aprobar cada uno de los parciales con un mínimo de 60/100. Cada parcial tiene dos recuperatorios. Se considera como nota válida la de la última instancia rendida

La nota de regularización es un promedio entre las calificaciones obtenidas en 4).

Para promocionar el alumno debe:

- 1) Cumplir con las condiciones para regularizar
- 2) Promocionar el trabajo grupal
- 3) Promocionar los trabajos de laboratorio
- 4) Puntuar al menos 70/100 en el parcial (o en recuperatorio)
- 5) Aprobar el Examen Integrador Global (EIG) con al menos 70/100

La nota de promoción es el promedio considerando las notas del ítem 4) y 5).

## IX - Bibliografía Básica

- [1] - Aceto, Luca et al. (2007). Reactive systems: Modelling, specification and verification. isbn: 9780511814105.
- [2] - Baier, Christel and Joost-Pieter Katoen (2008). Principles Of Model Checking. isbn: 9780262026499.
- [3] - Behrmann, Gerd et al. (Nov. 2006). A Tutorial on Uppaal 4.0. Department of Computer Science, Aalborg University.
- [4] - Burns, Alan and Andy Wellings (2009). Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX. 4th. Glenview, IL, USA: Addison-Wesley Educational Publishers Inc. isbn: 0321417453.
- [5] - Cavada, Roberto, Alessandro Cimatti, Gavin Keighren, et al. (2015). NuSMV 2.6 Tutorial. url: <http://nusmv.fbk.eu/NuSMV/tutorial/v26/tutorial.pdf> (visited on 03/15/2020).
- [6] - Cheng, Albert M.K. (2003). Real-time systems: scheduling, analysis, and verification. John Wiley & Sons.
- [7] - Fan, Xiaocong (2015). Real-time embedded systems: design principles and engineering practices. Newnes.
- [8] - Laplante, Phillip A. and Seppo J. Ovaska (2011). Real-Time Systems Design and Analysis.

## X - Bibliografía Complementaria

- [1] - Alur, Rajeev and David Dill (1990). "Automata for modeling real-time systems". In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). isbn:9783540528265.
- [2] - Berard, B. et al. (2010). Systems and Software Verification: Model-Checking Techniques and Tools. 1st. Springer Publishing Company, Incorporated. isbn:3642074782.
- [3] - Buttazzo, Giorgio C (2011). Hard real-time computing systems: predictable scheduling algorithms and applications. Springer Science & Business Media.
- [4] - Cavada, Roberto, Alessandro Cimatti, Charles Arthur Jochim, et al. (2010). NuSMV 2.6 User Manual. url: <http://nusmv.fbk.eu>. (visited on 03/15/2020).
- [5] - Clarke, Edmund M. et al. (1999). Model Checking. The MIT Press. isbn: 0262032708.
- [6] - David, Alexandre and Kim G Larsen (2011). "More Features in UPPAAL". In: Deliverable no.: D5. 12 Title of Deliverable: Industrial Handbook, pp. 49–76.
- [7] - Holzmann, Gerard (2003). Spin Model Checker, the: Primer and Reference Manual. First. Addison-Wesley Professional. isbn: 0321228626.
- [8] - Kopetz, Hermann (2011). Real-Time Systems: Design Principles for Distributed Embedded Applications. 2nd. Springer Publishing Company, Incorporated. isbn: 1441982361.

[9] - Kormanyos, Christopher (2015). Real-Time C++: Efficient Object-Oriented and Template Microcontroller Programming. 2nd. Springer Publishing Company, Incorporated. isbn: 3662478099.

[10] - Müller-Olm, Markus et al. (1999). "Model-checking: A tutorial introduction". In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). isbn: 3540664599.

[11] - Shin, Kang G. and Parameswaran Ramanathan (1994). "Real-Time Computing: A New Discipline of Computer Science and Engineering". In: Proceedings of the IEEE. issn: 15582256.

## **XI - Resumen de Objetivos**

Adquirir las competencias necesarias para integrar equipos de desarrollo de productos de software con restricciones de tiempo real.

## **XII - Resumen del Programa**

- Unidad I: Conceptualización
- Unidad II: Hardware
- Unidad III: Sistemas operativos de tiempo real
- Unidad IV: Lenguajes de programación para sistemas de tiempo real
- Unidad V: Ingeniería de requerimientos de tiempo real
- Unidad VI: Ingeniería de software de tiempo real
- Unidad VII: Análisis de desempeño
- Unidad VIII: Confiabilidad y tolerancia a fallas
- Unidad IX: Verificación de sistemas de tiempo real

## **XIII - Imprevistos**

Por cualquier consulta ponerse en contacto con el profesor responsable a la cuenta aljsanchez de GMail.

## **XIV - Otros**