



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 2023)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
SISTEMAS DE TIEMPO REAL	ING. INFORM.	026/1	2- 2023	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
13/03/2023	24/06/2023	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 tendencia de utilizar hardware y software para el monitoreo y control en STR. Sin lugar a dudas, la corrección de estos sistemas es de suma importancia. Una falla parcial o total puede acarrear consecuencias incómodas, en el mejor de los casos, o catastróficas, en los casos extremos.

El desarrollo de STR requiere conocer múltiples disciplinas. Involucra conocimiento que va desde Ingeniería de Software hasta Sistemas Operativos de Tiempo Real (SOTR), 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 calificados 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 el alumno adquiera la capacidad de:

- Identificar, formular y resolver problemas de tiempo real utilizando el enfoque del ingeniero, es decir, modelando, animando, analizando y verificando el sistema a construir.
- Concebir, diseñar y desarrollar proyectos de sistemas de tiempo real.
- Gestionar - planificar, ejecutar y controlar - proyectos de tiempo real.
- Utilizar técnicas y herramientas de la ingeniería de sistemas de tiempo real.
- Contribuir a la generación de desarrollos tecnológicos de sistemas de tiempo real.
- Desempeñarse de manera efectiva en equipos de trabajo.
- 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.

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.

Sobrepaso del tiempo de ejecución en el peor caso. Sobrepaso de tiempo en eventos esporádicos. Sobrepaso 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

01. Conceptualización
02. Hardware
03. Sistemas Operativos de tiempo real
04. Lenguajes de Programación para Sistemas de Tiempo Real
05. Ingeniería de requerimientos de tiempo real
06. Ingeniería de software de tiempo real
07. Análisis de desempeño
08. Confiabilidad y tolerancia a fallas
09. Verificación de sistemas – Tiempo discreto
10. Verificación de sistemas – Tiempo continuo

VIII - Regimen de Aprobación

Para regularizar el alumno debe:

- 1) Asistir a un 60% de las clases
- 2) Realizar un 80% de las evaluaciones diagnósticas
- 3) Aprobar un trabajo de laboratorio (60/100)
- 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 3) y 4).

Para promocionar el alumno debe:

- 1) Cumplir con las condiciones para regularizar
- 2) Asistir a un 80% de las clases
- 3) Aprobar el 80% de las evaluaciones diagnósticas
- 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 de la de regularización con la del EIG.

Estudiantes Libres: no se admiten por las características propias de los trabajos de laboratorio a desarrollarse durante todo el cuatrimestre.

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

El presente programa puede presentar ajustes en respuesta a la situación epidemiológica por COVID-19. Toda modificación será acordada con los alumnos e informada a Secretaría Académica.

En caso de imprevistos ponerse en contacto con el profesor responsable al correo [aljsanchez de gmail](mailto:aljsanchez@gmail.com).

XIV - Otros