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 2026)
(Programa en trámite de aprobación)
(Presentado el 22/04/2026 12:49:27)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
SISTEMAS DE TIEMPO REAL	ING. INFORM.	026/12-08/15	2026	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
SANCHEZ, ALEJANDRO	Prof. Responsable	P.Adj Exc	40 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	**Teóricas**	**Prácticas de Aula**	**Práct. de lab/ camp/ Resid/ PIP, etc.**	**Total**
Hs	2 Hs	1 Hs	2 Hs	5 Hs

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

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
11/03/2026	23/06/2026	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. Una falla puede acarrear consecuencias incomodas, 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 calificados para desempeñarse en proyectos de desarrollo de software donde las especificaciones temporales son un requerimiento fundamental. Esta materia procura que los alumnos adquieran tales conocimientos y capacidades necesarias. 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 STR.
O3. Utilizar técnicas y herramientas de la ingeniería en informática en STR.
O4. Contribuir a la generación de desarrollos tecnológicos de STR.
O5. Desempeñarse de manera efectiva en equipos de trabajo.
O6. Aprender de manera autónoma sobre problemáticas específicos de STR que refinan temas más generales ya vistos en la carrera.
O7. 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 el aprendizaje continuo.
E7. Fundamentos para una comunicación efectiva.

VI - Contenidos

Contenidos mínimos: Tiempo real: Concepto. Las restricciones temporales. Estudio de casos. Desarrollo de ejemplos. Ordenamiento de tareas. Ambientes de "tiempo real": hardware y dispositivos, kernels específicos, métodos de análisis y especificación. Programación a bajo nivel y sincronización de tareas. Requisitos de confiabilidad y tolerancia a fallas en Sistemas de Tiempo Real. Mecanismos de protección. Características de los sistemas con alta disponibilidad y balanceo de carga. Redundancia. Consideración de costos: tasa de error y consecuencias, modelado de la confiabilidad, disponibilidad, seguridad. Casos de estudio y simulación de su comportamiento. Tiempo discreto y tiempo denso. Herramientas de modelado. Model checking.

Unidad I: Conceptualización
Caractertización: sistema y sistema de tiempo real. Casos de estudio. Tiempo de respuesta. STR defectuoso. Sistema embebido. Restricciones de tiempo y sus tipos. Tipos de STR 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 STR. Comunicación y sincronización entre tareas. Gestión de memoria. Selección del sistema operativo de tiempo real.

Unidad IV: Lenguajes de programación para STR
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 STR
Fundamentos teóricos. Lógica temporal lineal (LTL). Lógica de árbol computacional (CTL). La herramienta NuSMV. Grafo temporizado. Redes de grafos temporizados. Lógica de árbol computacional temporizada (TCTL). La herramienta UPPAAL. Verificación de sistemas de hardware, software y protocolos de comunicación.

VII - Plan de Trabajos Prácticos

Metodología de enseñanza
- La teoría se dicta siguiendo una modalidad expositiva-participativa. Se emplean técnicas de trabajo compartido entre estudiantes y docentes, y se propicia la colaboración entre los participantes, tanto como el estudio personal y autónomo.
- Se impulsa a que los estudiantes recurran a la bibliografía de diferentes autores para la profundización, maduración y actualización del conocimiento. Con esta práctica se pretende que los estudiantes perciban las distintos enfoques propuestas por distintos autores para una misma problemática, promoviendo la permanente capacitación sobre temas específicos de la materia.
- Se desarrollan trabajos prácticos individuales y trabajos de laboratorio grupales con la finalidad de profundizar los temas vistos en la materia aplicando distintos recursos. Cada práctico parte de problemas básicos finalizando con ejercicios avanzados.
- Se realiza evaluación formativa solicitando la resolución de ejercicios a través de la herramienta Moodle – Aula Virtual, la cual permite brindar feedback de manera inmediata y automática. Esta forma de evaluación posibilita que el estudiante establezca una medida de apropiación del conocimiento. Los datos obtenidos de la herramienta se utilizan para detectar dificultades en el proceso. Alternativamene se realizan prácticos evaluados en el aula.
- Se procura que el estudiante adopte la conceptualización de la disciplina y se exprese utilizando un vocabulario acorde, discuta con sus pares sus respuestas y realice autocrítica de su nivel de comprensión.

Trabajos Prácticos
1. Conceptualización. Se procura que el estudiante se apropie de conceptos claves del dominio.
2. LTL. El objetivo es que el estudiante logre modelar propiedades en LTL dominando la semántica de cada uno de los operadores de la lógica.
3. NuSMV. Se apunta a que el estudiante adquiera la capacidad de especificar programas en NuSMV, animar y verificar los mismos usando LTL.
4. CTL. Se procura que el estudiante logre modelar propiedades CTL, apropiándose de la semántica de sus operadores, y verificando propiedades en NuSMV.
5. UPPAAL. La finalidad de este práctico es que el alumno adquiera la capacidad de modelar autómatas temporizados y

verificar propiedades en TCTL.

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 STR
- 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 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.

E7. 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 diapositivas 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 item 4) y 5).

IX - Bibliografía Básica

[1] - Reactive systems: Modelling, specification and verification. Aceto, Luca et al. Cambridge University Press (2007). isbn: 9780511814105.
[2] - Principles Of Model Checking. Baier, Christel and Joost-Pieter Katoen. The MIT Press (2008). isbn: 9780262026499.
[3] - Real-Time Systems: Design Principles for Distributed Embedded Applications. Hermann Kopetz, Wilfried Steiner. Springer (2023). isbn: 9783031119941
[4] - Real-Time Embedded Systems. J. Wang. Wiley. 2017. isbn:9781118116173
[5] - Hard Real-Time Computing Systems. Giorgio C. Buttazzo. Springer (2024). isbn: 9783031454097
[6] - Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX. Burns, Alan and Andy Wellings. Addison-Wesley Educational Publishers Inc. (2009). isbn: 0321417453.
[7] - Real-Time Systems Design and Analysis. Laplante, Phillip A. and Seppo J. Ovaska. Wiley-IEEE Press (2011).

X - Bibliografia Complementaria

[1] - Real-time systems: scheduling, analysis, and verification. Albert M.K. Cheng. John Wiley & Sons. 2003.
[2] - Real-time embedded systems: design principles and engineering practices. Xiaocong Fan. Newnes. 2015.
[3] - NuSMV Version 2.7.1. Disponible en https://nusmv.fbk.eu/ (última visita 15/4/2026)

[4] - UPPAL sitio web. https://uppaal.org/documentation/ (última visita 15/4/2026)
[5] - A Tutorial on Uppaal 4.0. Gerd Behrmann et al. Department of Computer Science, Aalborg University. 2006 (Nov.).
[6] - Automata for modeling real-time systems. Rajeev Alur and David Dill. Springer (Lecture Notes in Computer Science). 1990. ISBN: 9783540528265.
[7] - Systems and Software Verification: Model-Checking Techniques and Tools. B. Berard et al. Springer Publishing Company, Incorporated. 2010. 1st ed. ISBN: 3642074782.
[8] - Model Checking. Edmund M. Clarke et al. The MIT Press. 1999. ISBN: 0262032708.
[9] - More Features in UPPAAL. Alexandre David and Kim G Larsen. Deliverable D5.12 (Industrial Handbook). 2011. pp. 49–76.
[10] - Spin Model Checker, the: Primer and Reference Manual. Gerard Holzmann. Addison-Wesley Professional. 2003. First ed. ISBN: 0321228626.
[11] - Real-Time C++: Efficient Object-Oriented and Template Microcontroller Programming. Christopher Kormanyos. Springer Publishing Company, Incorporated. 2015. 2nd ed. ISBN: 3662478099.
[12] - "Model-checking: A tutorial introduction". Markus Müller-Olm et al. Springer (Lecture Notes in Computer Science). 1999. ISBN: 3540664599.
[13] - "Real-Time Computing: A New Discipline of Computer Science and Engineering". Kang G. Shin and Parameswaran Ramanathan. IEEE (Proceedings of the IEEE). 1994. 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 STR
- 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 STR

XIII - Imprevistos

Por cualquier consulta ponerse en contacto con el profesor responsable a la cuenta aljsanchez de g00gle.

XIV - Otros

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA	
	Profesor Responsable
Firma:	
Aclaración:	
Fecha:	