



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 20/04/2026 17:31:56)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
METODOS FORMALES	ING. INFORM.	026/1 2- 08/15	2026	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
FUNES, ANA MARIA	Prof. Responsable	P.Asoc Exc	40 Hs
SANCHEZ, ALEJANDRO	Prof. Colaborador	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	1° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
11/03/2026	23/06/2026	15	75

IV - Fundamentación

Hoy en día, las computadoras son empleadas para múltiples tareas en las cuales la presencia de fallas puede traer aparejado severas consecuencias, desde pérdidas materiales hasta vidas humanas. En este sentido, en las últimas décadas, las computadoras han pasado a jugar un rol esencial en el control aeroespacial, en la aeronavegación, en el control de trenes, autos, reactores nucleares y equipamiento hospitalario, por dar simplemente algunas pocas de las múltiples aplicaciones críticas donde son utilizadas.

Es por esta razón que resulta de vital importancia que los sistemas de software que están por detrás de estos sistemas críticos sean altamente confiables. Es en estas situaciones donde la aplicación de técnicas y métodos formales puede resultar de gran utilidad.

Para construir software de alta calidad, libre de errores y confiable, una de las vías más importantes, sin descartar otras alternativas, es definiendo el dominio y los requerimientos a través del uso de especificaciones formales. Una de las principales ventajas de las especificaciones formales es que no son ambiguas y que, además, permiten la verificación formal de propiedades deseables del sistema especificado, conduciendo al desarrollo de software de calidad.

V - Objetivos / Resultados de Aprendizaje

El objetivo de la materia es que los estudiantes tengan la posibilidad de incursionar en la aplicación de métodos formales como herramienta de producción de software de alta calidad, con especial énfasis en el uso de lenguajes de especificación

formal en las etapas iniciales del desarrollo de software.

Dado que se trata de la única materia asociada a la temática se busca que el estudiante adquiera un conocimiento integral sobre las múltiples variantes existentes de Métodos Formales, las situaciones en las cuales las mismas resultan adecuadas de aplicar así como que adquiera habilidades para su adopción y aplicación.

Durante el dictado de la asignatura se abordan los siguientes ejes transversales:

1) Identificación, formulación y resolución de problemas de informática.

Se aborda a partir del trabajo práctico 2 al 6, donde se trabaja con problemas que deben ser modelados con el soporte de herramientas. También se trabaja con este tipo de problemas en las evaluaciones parciales de la materia.

Se evalúa por medio de dos evaluaciones parciales.

2) Concepción, diseño y desarrollo de proyectos de ingeniería en sistemas de información/informática.

Se aborda fundamentalmente en el práctico de Alloy, donde se trabaja con problemas que deben ser modelados con el soporte de herramientas. También se trabaja con este tipo de problemas en la correspondiente evaluación parcial.

Se evalúa por medio de una evaluación parcial.

3) Utilización de técnicas y herramientas de aplicación en la informática.

Se aborda a partir del trabajo práctico 3, donde se comienza a trabajar con problemas que deben ser modelados con el soporte de herramientas de modelado y análisis de Redes de Petri; también los estudiantes emplean soporte automatizado para la especificación formal y la verificación de propiedades en Alloy.

En el caso de Alloy, el uso de la herramienta se evalúa en la correspondiente evaluación parcial donde los estudiantes hacen uso del Analizador Alloy.

4) Fundamentos para la comunicación efectiva.

Se busca que el estudiante vaya adquiriendo la capacidad de expresarse utilizando un vocabulario acorde a los contenidos vistos en la asignatura.

La expresión oral se aborda a partir de la interacción permanente con los docentes tanto durante las clases teóricas como en las prácticas. En ocasiones, en caso de que el examen final sea oral, también se evalúa en esa instancia.

La expresión escrita se evalúa a través de las evaluaciones parciales y, ocasionalmente, en los exámenes finales escritos.

5) Fundamentos para la acción ética y responsable.

Se implementa a través de la entrega desde el primer día de clase de la planificación de las actividades teóricas y prácticas de la asignatura para que los estudiantes puedan organizar adecuadamente sus horarios y cargas de estudio y trabajo. Asimismo, se informan las fechas de la Evaluación Global Integradora así como de los dos recuperatorios de los parciales.

La evaluación de este eje se da por medio de la verificación del respeto a fechas y formatos pactados.

6) Fundamentos para el aprendizaje continuo.

Este eje es abordado por medio de las actividades teóricas, donde los estudiantes pueden participar e interactuar con el docente, respondiendo preguntas o planteando inquietudes, todo esto complementado con lectura de material bibliográfico o apuntes de la cátedra. Por otro lado, las actividades prácticas permiten que el estudiante ponga a prueba y practique los conocimientos que haya ido adquiriendo en las clases teóricas.

La evaluación de este eje se realiza, de forma más informal, por medio del nivel de participación de los estudiantes en las clases prácticas, y formalmente, por medio de dos evaluaciones parciales y de un examen final o, en el caso de los estudiantes en condición de promocionar, de una Evaluación Global Integradora.

VI - Contenidos

Contenidos mínimos:

Especificaciones basadas en lógica de primer orden y teoría de conjuntos. Semántica de lenguajes basada en la teoría de conjuntos. Ejemplos: Métodos Z, VDM y B. Especificaciones algebraicas. Descripción de estructuras de datos estableciendo tipos y operaciones sobre esos tipos. Operaciones de un tipo definidas a través de un conjunto de axiomas o ecuaciones que especifican las restricciones que deben satisfacer las operaciones. Ejemplo de métodos: Larch, OBJ, TADs. Especificación de comportamiento. Métodos basados en álgebra de procesos. Interacción entre procesos concurrentes. Ejemplos: Métodos basados en Redes de Petri, Formalización de eventos concurrentes. Métodos basados en lógica temporal. Estudio detallado de RAISE y RSL.

Metodología de Trabajo:

La asignatura introduce al estudiante en la aplicación de métodos para la especificación y la verificación formal de propiedades de sistemas de software.

Las instancias de teoría y práctica son semanales. La teoría trabaja lineamientos conceptuales que se aplican luego en los

trabajos prácticos de aula y laboratorio. La teoría y la práctica se encuentran estrechamente vinculadas. En la práctica, se plantean ejercitaciones de aula de complejidad creciente, resolviendo los ejercicios con herramientas de soporte y desarrollo típicas en la comunidad de software libre.

En las clases se trabaja con los siguiente recursos:

- Apuntes, diapositivas, videos, libros, tutoriales.
- Multimedia, computadoras, demostraciones de usos de herramientas con ejemplos en vivo.
- Herramientas: herramientas de modelado de Redes de Petri y analizador de Alloy.

Para cada unidad se deja disponible el material correspondiente a los contenidos de la unidad, las diapositivas de clase, el apunte teórico y su correspondiente trabajo práctico en el repositorio digital de la asignatura.

Para una mejor organización, tanto de los estudiantes como del equipo docente, se presenta un cronograma con la descripción de las actividades que se realizan cada día de clase.

Unidad I: Aspectos Generales de los Métodos Formales

Introducción al curso y a los Métodos Formales. El papel de los Métodos Formales en Ingeniería de Software. Pros y contras de los Métodos Formales. Diversas clasificaciones de los Métodos Formales. Estilos de desarrollo formal. Grados de rigor.

Unidad II: Especificaciones Formales

Estilos de especificaciones de software. Características. Enfoques de especificación de un sistema. Especificaciones basadas en Lógica de Primer Orden y Teoría de Conjuntos. Especificaciones aplicativas, imperativas, concurrentes. Especificaciones orientadas al modelo. Especificaciones algebraicas. Especificaciones orientadas a la propiedad. Especificaciones axiomáticas. Ejemplos en RSL.

Unidad III: Integración de Técnicas Formales y Semi-formales con OCL.

Limitaciones de las especificaciones semiformales. Limitaciones de los diagramas de clase UML. El lenguaje OCL. Características. Especificación de valores iniciales de atributos. Especificación de reglas de derivación (valores de atributos derivados). Especificación del cuerpo de operaciones de consulta. Especificación de invariantes. Diseño por Contrato. Especificación de pre y post condiciones de operaciones.

Unidad IV: Especificación Formal del Comportamiento con Redes de Petri.

Definición formal de Red de Petri. Modelado del comportamiento con Redes de Petri. Interacción de procesos concurrentes. Formalización de eventos concurrentes y asincrónicos. Algunas extensiones a las Redes de Petri tradicionales. Análisis de una Red de Petri. Tipos de problemas a analizar. Árbol de alcanzabilidad y árbol de cobertura. Ecuaciones Matriciales.

Unidad V: Métodos Formales Livianos

Alloy. Especificaciones formales y verificación en Alloy. La lógica de Alloy: átomos, relaciones, constantes, funciones predefinidas, operadores, restricciones, expresiones, declaraciones, multiplicidades. El lenguaje Alloy: módulos, signaturas, hechos, predicados, funciones, aserciones y comandos run y check. Especificaciones estructurales y del comportamiento. Modelado Declarativo Implícito de Sistemas de Transición de Estados en Alloy. Especificación en Alloy de propiedades temporales haciendo uso de la Lógica Temporal Lineal (LTL). Uso del analizador Alloy para validar el modelo y verificar propiedades.

Unidad VI: Verificación Formal de Sistemas de Software

Aproximaciones. Verificación deductiva por demostración de teoremas. Sistema formal. Sintaxis, semántica y sistema de prueba. Relación de refinamiento. Refinamiento por pasos sucesivos. Refinamiento de funciones. Refinamiento de datos. Ejemplos en RAISE. Métodos formales transformacionales de desarrollo. Model Checking. Lógicas Temporales. Diseño por contrato. Métodos Formales Livianos.

Unidad VII: Especificación Formal en JML

Lenguajes de especificación de la interfaz de comportamiento. El lenguaje JML. Aplicaciones de JML. Aserciones en JML y su uso en tiempo de ejecución. Runtime-assertion-checking (RAC). Diseño por Contrato. Verificación deductiva en JML.

VII - Plan de Trabajos Prácticos

Práctico 1 (Aula). Introducción a los Métodos Formales.

Objetivos: Dadas las características netamente teóricas de los temas cubiertos en la unidad 1 del programa analítico, en este práctico se solicita que el estudiante responda una serie de preguntas acudiendo al conocimiento adquirido del estudio de dicha unidad.

Práctico 2 (Aula). OCL

Objetivo: En este práctico el estudiante aprenderá sobre la sintaxis y semántica del lenguaje OCL y su aplicación para complementar la semántica de diagramas de clase en UML.

Práctico 3 (Aula y Laboratorio). Modelado del comportamiento con Redes de Petri

Objetivo: En este práctico el estudiante aprenderá a modelar sistemas asincrónicos y concurrentes por medio de Redes de Petri tradicionales haciendo uso de alguna herramienta de modelado de libre distribución.

Práctico 4 (Aula y Laboratorio). Análisis de Redes de Petri.

Objetivo: El estudiante aprenderá a realizar un análisis de los problemas de seguridad, acotabilidad, conservación, vitalidad, alcanzabilidad y cobertura de modelos de redes de Petri de forma analítica y su corroboración por medio de herramientas automatizadas.

Práctico 5 (Laboratorio). Alloy: Modelado estático

Objetivo: El estudiante trabajará en el modelado estático de un sistema, especificando las propiedades estructurales del mismo y realizando la verificación de las propiedades planteadas, todo esto haciendo uso del analizador de Alloy.

Práctico 6 (Laboratorio). Alloy: Modelado dinámico

Objetivo: El estudiante trabajará en el modelado del dinámico de un sistema, especificando el comportamiento del mismo a lo largo del tiempo y realizando la verificación de las propiedades planteadas, todo esto haciendo uso del analizador de Alloy.

VIII - Regimen de Aprobación

Condiciones para REGULARIZAR la asignatura:

- Haber asistido al menos al 80% de las clases teóricas y prácticas de la asignatura.
- Haber aprobado dos parciales (o sus respectivas recuperaciones) con nota mayor o igual a 6 sobre 10.

Condiciones para PROMOCIONAR la asignatura:

- Haber asistido al menos al 80% de las clases teóricas y prácticas de la asignatura.
- Haber aprobado dos parciales (o sus respectivas recuperaciones) con nota mayor o igual a 7 sobre 10.
- Haber aprobado, al final de la cursada, un examen integrador con nota mayor o igual a 7 sobre 10.
- En caso de haber obtenido nota de promoción en todas las instancias de evaluación, el alumno aprobará la materia con una nota que surgirá del promedio de las mismas.

Recuperaciones:

- Se otorgarán dos recuperaciones para cada examen parcial. Se considera como válida la última nota obtenida en cada recuperación.
- El examen integrador no se recupera.

Examen Final:

En caso de regularizar la materia, el alumno deberá rendir un examen final, el cual podrá ser oral o escrito, en cualquiera de los turnos de examen establecidos.

Exámenes Libres:

Para aprobar un examen libre, el alumno deberá aprobar en una primera instancia una evaluación sobre temas prácticos de la materia con una nota de al menos 6 sobre 10. En caso de aprobar la evaluación práctica, el alumno además deberá aprobar una evaluación sobre temas teóricos con una nota de 4 sobre 10. La nota final surgirá del promedio de ambas notas.

IX - Bibliografía Básica

- [1] V.S. Alagar K. Periyasamy. Specification of Software Systems, 2nd edition, Springer (2011).
- [2] A. Haxthausen. An Introduction to Formal Methods for the Development of Safety-critical Applications. Technical report. (2010).
- [3] A. Funes y A. Dasso . Formal Methods Overview, Encyclopedia of Information Science and Technology, 3ra Edición, IGI Global. (2014).
- [4] Daneil Jackson. Software Abstractions, The MIT Press, (2012).
- [5] Julien Brunel, David Chemouil, Alcino Cunha, and Nuno Macedo. Formal Software Design with Alloy 6. Disponible en <https://haslab.github.io/formal-software-design/> (última visita 15/4/2026)
- [6] OMG. Object Constrain Language version 2.4. Disponible en <https://www.omg.org/spec/OCL/About-OCL/> (última visita 15/4/2026)
- [7] Warmer, Jos & Kleppe, Anneke. The Object Constraint Language: Getting Your Models Ready for MDA, Second Edition, Addison-Wesley Professional. (2003).
- [8] Peterson, James Lyle. Petri Net Theory and the Modeling of Systems, Prentice Hall. Disponible en <http://www.jklp.org/profession/books/pn/index.html> (última visita 15/4/2026)
- [9] Tutorial de JML, Disponible en <https://www.openjml.org/tutorial> (última visita 15/4/2026)

X - Bibliografía Complementaria

- [1] Daniel Jackson. A Language and Tool for Exploring Software Designs. Commun. ACM 62(9): 66-76. (2019).
- [2] Hung Dang Van, Chris George, Tomasz Janowski, and Richard Moore (Eds). Specification Case Studies in RAISE. FACIT series. Springer-Verlag. (2002).
- [3] The RAISE specification language, method, and tools. Disponible en <https://raisetools.github.io/> (última visita 15/4/2026)
- [4] B. Bérard, M. Bidot, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci et al. System and Software Verification, Model-Checking Techniques and Tools. Springer. (1999).
- [5] Funes, A., George, C. Formalizing UML class diagrams, capítulo 8 de UML and the Unified Process, Idea Group, (2003).
- [6] Huth, Michael & Ryan, Mark. Logic in Computer Science: Modelling and Reasoning about Systems, 2nd Edition. Cambridge University Press, (2004).

XI - Resumen de Objetivos

IncurSIONAR en los métodos formales como herramienta de producción de software de alta calidad.

XII - Resumen del Programa

Introducción a los Métodos Formales. Especificaciones formales. Aproximaciones a la verificación formal de sistemas de software y de sus propiedades. Métodos Formales livianos. Alloy. Model Checking. Verificación deductiva. JML. Otros estilos de especificaciones formales. Redes de Petri. OCL.

XIII - Imprevistos

XIV - Otros

Contacto: afunes@email.unsl.edu.ar

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA**Profesor Responsable**

Firma:

Aclaración:

Fecha: