



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 18/09/2026 10:47:46)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
INGENIERIA DE SOFTWARE I	ING. INFORM.	026/1 2- 08/15	2026	2° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
SANCHEZ, ALEJANDRO	Prof. Responsable	P.Adj Exc	40 Hs
MIRANDA, ENRIQUE ALFREDO	Responsable de Práctico	JTP Semi	20 Hs
ESCUDERO, LETICIA ALDANA	Auxiliar de Práctico	A.2da Simp	10 Hs

III - Características del Curso

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

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	2° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
03/08/2026	13/11/2026	15	75

IV - Fundamentación

Por ser la primera asignatura de Ingeniería de Software de la carrera, se plantean varios desafíos, entre ellos, enseñar una metodología para captura y análisis de requisitos. Los estudiantes trabajan a partir de diferentes situaciones problemáticas de la vida real, en equipos de pares.

Los contenidos abordados en la asignatura permiten al estudiante familiarizarse con los conceptos básicos del desarrollo de software, como una actividad ingenieril, a la vez que lo introducen al trabajo en equipo.

Teniendo en cuenta que se trata de la primera asignatura relacionada con la temática, se pretende que el estudiante aprenda los fundamentos básicos de cada una de las etapas que comprenden el proceso completo de desarrollo de software así como la aplicación de un método de desarrollo particular en las primeras etapas del ciclo de vida. Asimismo, se incursiona en las principales técnicas de validación y verificación del software.

Se fomenta un ambiente de trabajo en equipo que despierte el espíritu colaborativo de la práctica del desarrollo de software.

V - Objetivos / Resultados de Aprendizaje

Introducir al estudiante en los conceptos fundamentales de la Ingeniería de Software, sus métodos, técnicas y procesos, con hincapié en los conceptos y herramientas de modelado correspondientes a la producción de software con un enfoque de calidad.

Dado que se trata de la primera materia de Ingeniería de Software de la carrera, se busca profundizar en las primeras etapas del ciclo de vida (captura y análisis de requisitos), además de hacer incursionar al estudiante en tópicos relacionados con la validación y verificación, en especial en lo que hace a estrategias y técnicas de prueba del software. Se busca, a la vez, que el estudiante comience a desarrollar habilidades para trabajo en equipo, así como de comunicación en forma oral y escrita. 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 4, donde se comienza a trabajar con problemas que deben ser modelados con el soporte de herramientas. También se trabaja con este tipo de problemas en las evaluaciones parciales y en el Trabajo Práctico Integrador.

Se evalúa en la entrega de informe y presentación oral del Trabajo Práctico Integrador grupal así como en las evaluaciones escritas de la materia.

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

Se aborda a partir del trabajo práctico 2, donde se comienza a trabajar con problemas que deben ser modelados con el soporte de herramientas. También se trabaja con este tipo de problemas en las evaluaciones parciales y en el Trabajo Práctico Integrador.

Se evalúa en la entrega de informe y presentación oral del Trabajo Práctico Integrador grupal así en las evaluaciones escritas de la materia.

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

Se aborda a partir del trabajo práctico 2, donde se comienza a trabajar con problemas que deben ser modelados con el soporte de herramientas de modelado UML; también los estudiantes emplean herramientas para realizar pruebas automatizadas de unidad y cobertura, usando JUnit y JaCoCo. Asimismo, se trabaja con este tipo de técnicas y herramientas en el desarrollo del Trabajo Práctico Integrador.

Se evalúa en la entrega de informe y presentación oral del Trabajo Práctico Integrador grupal.

4) Fundamentos para el desempeño en equipos de trabajo.

Se aborda a partir del trabajo práctico 2 al 4, donde se trabaja en equipo con problemas que deben ser modelados con el soporte de herramientas para luego ser presentados en un Trabajo Práctico Integrador.

Se evalúa en la entrega de informe y presentación oral grupal del Trabajo Práctico Integrador.

5) 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 del Trabajo Práctico Integrador, el cual debe ser presentado y defendido públicamente frente a los docentes y pares. La misma se evalúa en esta exposición oral, donde se observa cómo se desenvuelve el estudiante y se le realizan las observaciones pertinentes al finalizar la misma.

La expresión escrita se aborda a través de la entrega del informe del Trabajo Práctico Integrador y se evalúa a través de devoluciones realizadas en el informe presentado del Trabajo Práctico Integrador. En las correcciones se hace hincapié no sólo en lo disciplinar sino también en cuestiones de redacción. Esto permite que el estudiante vaya desarrollando esta habilidad para las posteriores evaluaciones escritas de la materia.

6) Fundamentos para la acción ética y responsable.

Se implementa a través de la entrega a los estudiantes, desde el primer día de clase, de la planificación de las actividades teóricas y prácticas de la asignatura para que puedan organizar adecuadamente sus horarios y cargas de estudio y trabajo.

Asimismo, se informan las fechas de entrega y presentación del Trabajo Práctico Integrador, de las dos evaluaciones parciales con sus respectivos recuperatorios y de la Evaluación Global Integradora.

La evaluación de este eje se da por medio de la verificación del respeto a fechas y formatos pactados, producciones propias y entregas grupales, con participación de todos los integrantes.

7) 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 de la participación de los estudiantes en las clases prácticas, y formalmente, por medio de la evaluación del Trabajo Práctico Integrador, las evaluaciones parciales y la Evaluación Global Integradora.

8) Fundamentos para el desarrollo de una actitud profesional emprendedora.

Se aborda desde el Trabajo Práctico Integrador donde los estudiantes trabajan de forma grupal, como una aproximación a un desarrollo real, con vencimientos de entrega y presentación.

VI - Contenidos

CONTENIDOS MINIMOS (OCD-026-12):

Software e Ingeniería de software. Modelo de Procesos de software. Iteración de proceso. Actividades de proceso. Ingeniería de Software Asistida por Computadora. Proceso de diseño de interfaz de usuario. Modelos del Sistema. Modelo de Contexto. Modelo de comportamiento. Modelo de datos. Desarrollo de software rápido y prototipado. Verificación y Validación. Inspecciones de software. Análisis estático automatizado. Verificación formal. Estrategias de prueba del software. Técnicas de prueba del software. Automatización de las pruebas. Métricas del producto para el software.

METODOLOGÍA DE TRABAJO:

La asignatura introduce al estudiante en la formación experimental y profesional del desarrollo de software, ubicándolo en un entorno de trabajo similar al real y cotidiano, focalizado en especial en las etapas iniciales del desarrollo de software.

La teoría y la práctica se encuentran estrechamente vinculadas. Estas instancias son semanales. La teoría trabaja lineamientos conceptuales que se aplican luego en los trabajos prácticos de aula y laboratorio.

Luego de cubrir las teorías de los temas de la unidad 2 y 3, se organizan grupos de estudiantes a los cuales se les asigna, por grupo, un problema diferente de la vida real, a partir del cual deben producir los correspondientes modelos de casos de uso del sistema con las correspondientes especificaciones de los casos de uso, el modelo del dominio y un prototipo visual de las interfaces de usuario del sistema. El trabajo propuesto se realiza en grupos de 3 (tres) alumnos y no menos de 2 (dos). En este trabajo práctico se realiza un seguimiento exhaustivo por parte del equipo docente, quienes se reparten esta tarea de seguimiento por grupos.

Finalmente, en una fecha preestablecida, todos los grupos deben enviar un informe de su trabajo y hacer una exposición oral y defensa del mismo frente a los docentes y pares. Al mismo tiempo, se plantean ejercitaciones de aula de complejidad creciente, resolviendo los ejercicios con herramientas de soporte y desarrollo, típicas en la comunidad de software libre.

Se trabaja con los siguientes recursos:

- Apuntes, diapositivas, videos, libros, tutoriales.
- Multimedia, computadoras, demostraciones de usos de herramientas con ejemplos en vivo.
- Herramientas: herramientas de modelado UML de libre distribución, JUnit, JaCoCo, herramienta de prototipado de GUIs de libre distribución.

Para cada unidad, se deja disponible el material correspondiente, en el repositorio digital de la asignatura: diapositivas de las clases, apunte teórico, bibliografía, y su correspondiente trabajo práctico.

Para una mejor organización, tanto de los estudiantes como del equipo docente, se presenta, al iniciar el dictado, un cronograma con la descripción de las actividades que se realizarán cada día de clase. El mismo puede sufrir, durante su desarrollo, adaptaciones debido a contingencias que afecten el normal curso del dictado, las cuales son oportunamente comunicadas.

PROGRAMA ANALITICO:

Unidad 1: Conceptos de Software e Ingeniería de Software

Aspectos históricos. Crisis del software. Software e Ingeniería del Software. Tendencias en la Ingeniería del Software.

Cuestionamientos e intereses de la Ingeniería del Software. Aspectos claves de la Ingeniería del Software. Características del software.

Unidad 2: El proceso de Desarrollo de Software

Modelos de ciclo de vida del software. Modelo teórico. Actividades de Proceso. Actividades de protección o "sombra".

Especificación. Diseño. Implementación. Validación. Evolución. Modelo en cascada. Modelo de construcción de prototipos.

Modelos evolutivos. Modelo en espiral. Proceso Unificado. Metodologías ágiles. Herramientas y técnicas para modelado de procesos.

Unidad 3: Modelado Básico del Comportamiento

Modelado del Comportamiento con Casos de Uso. Diagramas de casos de uso. Términos y conceptos. Caso de uso: actores, flujos de eventos, escenarios. Organización de los casos de uso. Relaciones entre casos de uso. Relaciones entre actores.

Relaciones entre actores y casos de uso. Especificación de los Casos de Uso. Técnicas comunes de modelado: modelado del contexto del sistema, modelado de los requisitos funcionales de un sistema, modelado de los procesos del negocio.

Unidad 4: Modelado del Dominio

Modelo Conceptual o Modelo del Dominio. Elementos básicos de los Diagramas de Clases UML para la construcción de modelos conceptuales: clases, atributos, asociaciones, multiplicidades, clase asociación, agregación, composición, asociaciones n-arias, generalizaciones.

Unidad 5: Modelo de Presentación o Interfaces

Sistemas interactivos y diseño de interfaces. La Interfaz de Usuario (IU). Especificación del diálogo Hombre-Máquina. Descripción estática y dinámica. Diagrama de navegación de ventanas. El concepto de Usabilidad. La importancia y los beneficios del diseño de una IU usable. Heurísticas de usabilidad para el diseño de la IU.

Unidad 6: Validación y Verificación del Software

Conceptos de validación y verificación del software. Técnicas estáticas y dinámicas. Inspecciones de software. Análisis estático automatizado. Concepto de verificación formal. El proceso de prueba (Testing). Estrategias: prueba de unidad, prueba de integración, prueba de validación, prueba de sistema. Técnicas de Prueba: Prueba funcional y prueba estructural. Automatización de las pruebas. JUnit.

Unidad 7: Métricas del Software

Introducción a la calidad del software. Métricas del Producto de Software: Métricas del modelo de Análisis, métricas del modelo de Diseño, métricas del código fuente, métricas para pruebas, métricas de mantenimiento. Estimación de Proyectos de Software. Puntos de Casos de Uso.

Unidad 8: Metodologías Ágiles

Metodologías Ágiles. Concepto, diferencia con metodologías tradicionales. Manifiesto de desarrollo ágil. Principios. Programación Extrema (XP).

VII - Plan de Trabajos Prácticos

Práctico 1: La Ingeniería de Software y los modelos de ciclo de vida del software (aula).

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

Práctico 2: Modelado básico del comportamiento - Diagramas de casos de uso (aula y laboratorio).

Objetivos: En este práctico el alumno aprenderá sobre:

- la sintaxis y semántica de los diagramas de casos de uso de UML.
- la aplicación de los diagramas de casos de uso para construir modelos del comportamiento de un sistema.
- uso de alguna herramienta de modelado con diagramas de casos de uso de UML.

Práctico 3: Modelado del dominio con diagramas de clases (aula y laboratorio).

Objetivos: En este práctico el alumno aprenderá sobre:

- la sintaxis y semántica de un subconjunto de elementos de modelado de los diagramas de clases de UML, necesarios para la creación de modelos conceptuales del sistema.
- la aplicación de los diagramas de clases para construir modelos conceptuales del sistema.
- uso de alguna herramienta de modelado de diagramas de clases de UML.

Práctico 4: Modelo de presentación – Interfaces de usuario (aula y laboratorio).

Objetivos: En este práctico el alumno aprenderá sobre:

- el diseño de GUIs
- uso de herramienta de prototipado de GUIs.

Práctico 5: Prueba estructural (aula).

Objetivos: En este práctico el alumno aprenderá sobre el uso de las principales técnicas de Caja Blanca. En el práctico 7, harán uso de la herramienta de cobertura JaCoCo, en combinación con las pruebas de unidad, para medir porcentajes de

algunas de las coberturas estudiadas.

Práctico 6: Prueba funcional (aula).

Objetivo: En este práctico el alumno aprenderá sobre el uso de las principales técnicas de Caja Negra.

Práctico 7: Prueba de unidad con JUnit (laboratorio).

Objetivos: En este práctico el alumno aprenderá a implementar pruebas de unidad haciendo uso de JUnit.

Práctico 8: Métricas del software (aula).

Objetivos: En este práctico el alumno aprenderá a estimar la funcionalidad de un proyecto haciendo uso de la métrica de Puntos de Caso de Uso.

Práctico Integrador (laboratorio): Proyecto grupal, con presentación oral y pública, del desarrollo del modelo de requisitos (modelo de casos de uso, modelo del dominio, modelo de interfaces) para un dominio particular asignado por la cátedra a cada grupo.

VIII - Regimen de Aprobación

Condiciones para REGULARIZAR la asignatura:

- Haber asistido al menos al 70% de las clases teóricas y prácticas de la asignatura.
- Haber aprobado el Trabajo Práctico Integrador (TPI) solicitado por la cátedra con nota mayor o igual a 6.
- Haber aprobado dos exámenes parciales o una de sus respectivas dos recuperaciones con nota mayor o igual a 6.

Condiciones para PROMOCIONAR la asignatura:

- Haber asistido al menos al 80% de las clases teóricas y prácticas de la asignatura.
- Haber aprobado el TPI solicitado por la cátedra con nota mayor o igual a 7.
- Haber aprobado dos exámenes parciales o una de sus respectivas dos recuperaciones con nota mayor o igual a 7.
- Haber aprobado una Evaluación Global Integradora (EGI) con nota mayor o igual a 7.
- En caso de obtener nota de promoción, el alumno aprobará la materia con una nota que surgirá del promedio de las notas obtenidas de cada una de las instancias de evaluación.

Trabajo Práctico Integrador (TPI):

El TPI es grupal, con grupos de 3 estudiantes; excepcionalmente se aceptarán grupos de 2 o de 4 estudiantes. Al finalizar, el grupo deberá presentar un informe escrito del mismo y todos los miembros del grupo deberán defenderlo en una exposición oral y pública.

Recuperaciones:

Cada parcial cuenta con dos recuperaciones, quedando la última nota obtenida. El TPI y la EGI no se recuperan.

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 en el Calendario Académico. En dicho examen se evalúan conceptos teóricos en los cuales se basa la práctica regularizada.

Exámenes Libres:

Dada las características de la práctica de la asignatura, no se admitirán exámenes de alumnos libres.

IX - Bibliografía Básica

[1] Alianza Ágil, <http://www.agilealliance.org>

[2] Ambler, Scott, "Agile Modeling: Effective practices for Extreme Programming and the Unified Process", John Wiley & Sons, 2002.

[3] Beck, Kent, "Extreme Programming Explained". Addison-Wesley. 2004.

[4] Bolaños, Daniel et al. "Pruebas de software y JUnit", Prentice-Hall, 2008.

[5] Booch, Rumbaugh, Jacobson. The Unified Modeling Language User Guide, 2nd Edition. Addison-Wesley, 2005.

[6] Booch, Rumbaugh, Jacobson. The Unified Modeling Language Reference Manual, 2nd Edition. Addison-Wesley, 2005.

- [7] Jacobson, Ivar et al. Object Oriented Software Engineering. A Use Case Driven Approach. Addison Wesley, 1992.
- [8] Jorgensen, Paul C. "Software Testing, a Craftsman's Approach", 4ta edición, CRC Press, 2014.
- [9] Pressman, Roger. "Ingeniería del Software - Un enfoque práctico", Mc Graw Hill, 9na Edición. 2021.
- [10] Material producido por la Cátedra.

X - Bibliografía Complementaria

- [1] Briano, César Ariel; Compilación de apuntes sobre conceptos fundamentales de la Ingeniería de Software, 2da Edición, 2023. Biblioteca Digital UBA (http://bibliotecadigital.econ.uba.ar/download/libros/Briano_compilacion_apuntes.pdf).
- [2] Brooks, Frederik P. - No Silver Bullet. Essence and Accidents in Software Engineering. IEEE Computer. Abril 1987.
- [3] Fenton, Norman E. & Pfleeger, Shari Lawrence. "Software Metrics, A Rigorous and Practical Approach, 2nd edition", 1997.
- [4] Pankaj Jalote. "An Integrated Approach to Software Engineering, 3rd ed.", Springer, 2005.
- [5] Sommerville, Ian; "Software Engineering", Addison-Wesley, 9na edición. 2011.

XI - Resumen de Objetivos

Introducir conceptos y herramientas básicos correspondientes a la producción de software con un enfoque industrial.

XII - Resumen del Programa

La Ingeniería de software. Modelos de ciclo de vida del software. Modelado Básico del Comportamiento. Modelado del Dominio. Modelo de Presentación. Interfaces de Usuario. Usabilidad. Validación y Verificación del software. Métricas del software. Estimaciones. Metodologías Ágiles.

XIII - Imprevistos

El presente programa podría presentar ajustes. Toda modificación será acordada y comunicada con el estudiantado e informada a Secretaría Académica.

XIV - Otros

Docentes de Contacto: aljsanchez@unsl.edu.ar y eamiranda@unsl.edu.ar

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