



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 08/05/2026 20:25:08)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
INGENIERIA DEL SOFTWARE	ING. EN COMPUT.	28/12	2026	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
ABDELAHAD, CORINA NATALIA	Prof. Responsable	P.Adj Exc	40 Hs
BERNARDIS, EDGARDO	Auxiliar de Práctico	JTP Semi	20 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	3 Hs	6 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	90

IV - Fundamentación

Se introduce al estudiante en el desarrollo de software como una actividad ingenieril y trabajo en equipo. Se pretende que el estudiante aprenda los fundamentos básicos de cada una de las etapas que comprende el proceso de desarrollo de software así como las distintas técnicas y metodologías aplicables, conocimientos de los principios de diseño y las principales técnicas de validación y verificación del software incluyendo conceptos de gestión de proyectos.

V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso se espera que el estudiante sea capaz de asimilar los conceptos de procesos de desarrollo de software, desde su especificación, análisis, diseño hasta su verificación y validación, incorporando conceptos de gestión de proyectos. El objetivo es que el estudiante tome conocimiento de conceptos básicos de ingeniería de software desde un punto de vista gerencial, transversal al desarrollo en sí, así como también de la automatización de procesos de desarrollo de software. Durante el dictado de la asignatura se abordan los siguientes ejes transversales:

- Identificación, formulación y resolución de problemas de informática
- Utilización de técnicas y herramientas de aplicación en la informática
- Fundamentos para el aprendizaje continuo

VI - Contenidos

CONTENIDOS MÍNIMOS:
 El proceso de software: Ciclos de vida, herramientas. Ingeniería de requerimientos, introducción a los métodos formales. Análisis, diseño, implementación, verificación, validación y mantenimiento de software. Gestión de proyectos: Planificación, métricas, estimaciones, análisis y gestión del riesgo. Conceptos de calidad de software.

PROGRAMA:

Unidad I

Ingeniería de Software. Principios. Modelos. Software. Conceptos. Evolución del software. Características del software. Ciclos de vida. Modelos de procesos de producción de software. Producto. Proceso, metodologías y herramientas. Evolución de las metodologías de desarrollo de sistemas de software. Ingeniería de la Información.

Unidad II

Modelos en UML. Introducción. Modelos. Importancia de los modelos. Modelos estáticos. Clases: atributos, operaciones y responsabilidades. Relaciones: dependencia, generalización y asociación. Modelos dinámicos. Diagrama de Interacción: Secuencia y Colaboración. Modelo de Casos de Usos.

Unidad III

Proceso de Desarrollo. Análisis, diseño, implementación, pruebas y mantenimiento de software. Ingeniería de Requerimientos. Metodologías Ágiles. Conceptos. Principios. Ventajas y desventajas de las metodologías ágiles.

Unidad IV

Calidad del Software. Conceptos. Control. Verificación, Validación. Pruebas de errores del software. Medidas de fiabilidad y disponibilidad.

Unidad V

Métodos Formales. Conceptos básicos. Deficiencias de los enfoques menos formales. Las matemáticas en el desarrollo de software. Notación matemática para la especificación formal.

Unidad VI

Gestión de proyectos. Medidas, métricas e indicadores. Estimaciones. Análisis y Gestión del Riesgo. Planificación.

VII - Plan de Trabajos Prácticos

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.

Metodología aplicada en el proceso de enseñanza-aprendizaje:

Los trabajos prácticos vinculados a las unidades del programa consisten en ejercicios a resolver. Además, se abordan aspectos vinculados al ciclo de vida del software, ingeniería de requerimientos, análisis, diseño, verificación y validación, así como también conceptos asociados a calidad de software y gestión de proyectos.

Antes de iniciar cada trabajo práctico, los estudiantes deben estudiar los conceptos teóricos propuestos y el material bibliográfico correspondiente. Cada actividad práctica es revisada por los docentes con el fin de brindar devoluciones y acompañar el proceso de aprendizaje.

Laboratorio 1: Modelado Estático con UML. Ingeniería Directa e Ingeniería Inversa con Java.

Laboratorio 2: Modelado Dinámico con UML. Ingeniería Directa e Ingeniería Inversa con Java.

Laboratorio 3: Utilización de Herramientas que soportan la Gestión. Herramientas automáticas de estimación.

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

Práctico 2: Pruebas Estructural.

Práctico 3: Pruebas Funcional.

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

Eje: Identificación, formulación y resolución de problemas de informática

Cómo se aborda: Se aborda a partir de la unidad 2 mediante el desarrollo de trabajos prácticos, guiadas por clases teóricas, diapositivas de clase, apuntes teóricos, prácticos de aula y consultas grupales e individuales.

Cómo se evalúa: Mediante un seguimiento continuo de parte de los docentes. Control de ejercicios en el pizarrón. Además, mediante una evaluación parcial en las fechas predeterminadas en el cronograma de la materia, cada evaluación posee su respectiva recuperación.

Eje: Utilización de técnicas y herramientas de aplicación en la informática

Cómo se aborda: El estudiante debe modelar utilizando herramientas de modelados específicas, y la programación se desarrolla en el lenguaje Java.

Cómo se evalúa: A través de la presentación de los modelos, luego en la presentación oral del mismo se observa el código fuente del sistema desarrollado.

Eje: Fundamentos para el aprendizaje continuo

Cómo se aborda: Todas las unidades tienen actividades prácticas para que los estudiantes respondan participando de las clases teóricas y clases prácticas. En cada práctico se hace un seguimiento de los ejercicios realizados por el estudiante.

Cómo se evalúa: Cada entrega/consulta tiene una corrección informada.

VIII - Regimen de Aprobación

La materia se desarrolla con la modalidad de promoción sin examen final.

Existen dos niveles:

a) Regularización: Para regularizar la materia se deberá:

1.- Tener como mínimo un 70% de asistencia a clases.

2.- Aprobar dos evaluaciones parciales o sus respectivas recuperaciones con una nota mayor o igual a 6 (seis).

Cada evaluación parcial posee dos recuperaciones. En caso de no aprobar una evaluación parcial, deberá aprobar alguno de sus recuperatorios con nota mayor o igual a 6 (seis).

b) Promoción sin examen final: Para regularizar y aprobar la materia se deberá:

1.- Tener como mínimo un 80% de asistencia a clases.

2.- Aprobar dos evaluaciones parciales o sus respectivas recuperaciones con una nota mayor o igual a 7 (siete). Cada evaluación parcial posee dos recuperaciones. En caso de no aprobar una evaluación parcial, deberá aprobar alguno de sus recuperatorios con nota mayor o igual a 7 (siete).

3.- Aprobar un coloquio de carácter integrador oral o escrito con una nota mayor o igual a 7 (siete).

La nota será el promedio de la nota de los parciales y del coloquio de carácter integrador.

Aquellos estudiantes que sólo regularicen la materia deberán rendir un examen final, en los turnos establecidos.

La asignatura puede rendirse libre.

IX - Bibliografía Básica

[1] Pressman, Roger; "Ingeniería del Software", Mc Graw Hill, 9na Edición, 2021.

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

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

[4] J. O' Brien & G. Marakas, "Introduction to Information Systems", 16th edition. 2012.

[5] Sommerville, Ian; "Software Engineering", Pearson, 10th Edición. 2015.

[6] Pankaj Jalote; "An Integrated Approach to Software Engineering, 3ra edición". Springer. 2005.

[7] Jorgensen, Paul C., DeVries, Byron; "Software Testing, a Craftsman's Approach", 5th edición, CRC Press, 2021.

[8] Ghezzi, Carlo, Jazayeriy Mehdi, Mandrioli, Dino; "Fundamentals of Software Engineering", Pearson, 2002.

[9] Martín Fowler, The New Methodology, <http://www.martinfowler.com/articles/newMethodology.html>, (última visita 08/05/2026)

[10] Kent Beck, "Extreme Programming Explained: Embrace Change ", Addison-Wesley, 2nd edición, 2004.

[11] Sutherland, Jeff; "Scrum: The Art of Doing Twice the Work in Half the Time" Crown Currency; 2014.

[12] Scott Ambler, "Agile Modeling: Effective practices for Extreme Programming and the Unified Process", John Wiley & Sons, 2008.

[13] Apuntes de la Cátedra.

X - Bibliografía Complementaria

[1] Booch, Grady. Object-Oriented analysis and design with applications. The Benjamin/Cummings Publishing Company Inc. 1994.

[2] Stephen Schach. "Ingeniería de Software Clásica y Orientada a Objetos, 6ta ed.", Mc Graw Hill, 2005.

[3] Beck, Kent. Test-driven development by example. Pearson Education, 2003.

[4] Martin, James, "Information Engineering", Prentice Hall, 1991.

[5] Hammer M., Champy, J. "Reingeniería", Norma, 1993.

[6] Fenton, N.E., Pfleeger, S.L., Software Metrics: a Rigorous and Practical Approach, 2nd Ed., PWS Publishing

Company, 1997.

[7] Arthur, Lowell Jay, "Rapid Evolutionary Development", Wiley, 1992.

[8] Scott Ambler, "Agile Modeling and the Unified Process", <http://www.agilemodeling.com/essays/agileModelingRUP.htm>, 2002.

[9] Pekka Abrahamsson, Outi Salo, Jussi Ronkainen & Juhani Warsta, "Agile Software Development Methods: Review and Analysis", VTT, 2002.

[10] "Extreme Programming: A gentle introduction", <http://www.extremeprogramming.org/>

XI - Resumen de Objetivos

Asimilar los conceptos de procesos de desarrollo de software, desde su especificación, análisis, diseño hasta su verificación y validación, incorporando conceptos de gestión de proyectos.

XII - Resumen del Programa

Introducción a la Ingeniería de Software. Métodos Formales. Gestión de Proyectos de Software. Modelos en UML. Procesos de Desarrollo. Análisis y Gestión del Riesgo. Calidad del Software.

XIII - Imprevistos

XIV - Otros

Contacto: cabdelah@unsl.edu.ar

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA

	Profesor Responsable
Firma:	
Aclaración:	
Fecha:	