Ministerio de Cultura y Educación **(Programa del año 2026)**
Universidad Nacional de San Luis **(Programa en trámite de aprobación)**
Facultad de Ciencias Físico Matemáticas y Naturales **(Presentado el 19/08/2026 11:19:37)**
Departamento: Informatica
Area: Area IV: Pr. y Met. de Des. del Soft.

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
INGENIERIA DE SOFTWARE II	ING. INFORM.	OCD-3-2/2025	2026	2° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
ABDELAHAD, CORINA NATALIA	Prof. Responsable	P.Adj Exc	40 Hs
RIESCO, DANIEL EDGARDO	Prof. Colaborador	P.Asoc Exc	40 Hs
BERNARDIS, HERNAN	Responsable de Práctico	JTP Semi	20 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	2º Cuatrimestre

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
03/08/2026	13/11/2026	15	90

IV - Fundamentación

Brindar las bases teóricas y prácticas que permitan al Ingeniero de Software analizar, especificar, validar y gestionar los requerimientos de software, así como aplicar procesos y metodologías de desarrollo orientadas a objetos y enfoques ágiles para el diseño e implementación de productos de software.
Se promueve un entorno de trabajo colaborativo que favorezca el desarrollo de competencias para el trabajo en equipo, propias de la práctica profesional en el desarrollo de software.

V - Objetivos / Resultados de Aprendizaje

Desarrollar competencias para analizar, especificar, validar y gestionar los requerimientos de software, aplicando procesos, metodologías y herramientas de desarrollo orientadas a objetos y enfoques ágiles que permitan diseñar, construir e implementar productos de software.

Durante el dictado de la asignatura se abordan los siguientes ejes transversales:
Identificación, formulación y resolución de problemas de ingeniería de software.
o Concepción, diseño y desarrollo de proyectos de ingeniería de software.

o Utilización de técnicas y herramientas de aplicación en ingeniería de software.
o Fundamentos para el desempeño en equipos de trabajo.
o Fundamentos para una comunicación efectiva.
o Fundamentos para una actuación profesional ética y responsable.
o Fundamentos para evaluar y actuar en relación con el impacto social de su actividad profesional en el contexto global y local.
o Fundamentos para el aprendizaje continuo.

VI - Contenidos

Contenidos Mínimos:
Requerimientos de Software. Estudio de Factibilidad. Análisis y elicitación de requerimientos. Validación de Requerimientos. Gestión de requerimientos. Tipos de Sistemas. Proceso de Desarrollo orientado a objetos. Modelos de Objetos. Reuso de software. Patrones de Diseño. Frameworks de aplicaciones. Ingeniería de la información. Ingeniería de software basada en componentes. Modelo de componentes. Composición de componentes. Metodologías Ágiles.

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 ingeniería de software.
Cómo se aborda: Se aborda a partir de la unidad 1 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: Concepción, diseño y desarrollo de proyectos de ingeniería de software.
Cómo se aborda: El estudiante debe trabajar en la construcción de un software orientado a objetos usando herramientas que automatizan el proceso de desarrollo generando los distintos artefactos desde los requerimientos hasta su implementación con un caso de estudio real. Para ello se trabajará de manera incremental, primero el desarrollo de los modelos correspondientes, para luego llegar al código y posteriormente las pruebas.
Cómo se evalúa: A través de la presentación de un informe con los artefactos construidos y problemas surgidos. Además, deben realizar una presentación oral explicando el caso de estudio, mostrando los modelos construidos y posteriormente la ejecución de dicho sistema.

Eje: Utilización de técnicas y herramientas de aplicación en ingeniería de software.
Cómo se aborda: El estudiante debe modelar utilizando herramientas de modelados específicas, y la programación se desarrolla en el lenguaje de programación actuales.
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 desempeño en equipos de trabajo
Cómo se aborda: El proyecto Integrador se realiza conformando equipos de trabajo de 2 o 3 estudiantes.
Cómo se evalúa: A lo largo de las entregas parciales del proyecto integrador se verifica que cada integrante del grupo pueda explicar la tarea realizada.

Eje: Fundamentos para la comunicación efectiva
Cómo se aborda:
Expresión oral: Se realizan exposiciones de entregas parciales del Proyecto Integrador y se socializa con compañeros y docentes.
Expresión escrita: Se realiza un informe del Proyecto Integrador.
Cómo se evalúa:
Expresión oral: Mediante una rúbrica que se le entrega al estudiante con el enunciado del proyecto integrador. Se busca que el estudiante siga adquiriendo la capacidad de expresarse utilizando un vocabulario acorde a los contenidos vistos en la

asignatura.
Expresión escrita: En las correcciones informadas se hace hincapié no sólo en lo disciplinar sino también en cuestiones de redacción.

Eje: Fundamentos para evaluar y actuar en relación con el impacto social de su actividad en el contexto global y local

Cómo se aborda: Desde el inicio se fomentará una actitud crítica que les permita evaluar el impacto social que tendrá el sistema a desarrollar.
Como se evalúa: Cada entrega del proyecto integrador posee un análisis sobre el impacto social que tendrá el sistema.

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.

Los contenidos mínimos se desglosan en las siguientes unidades:

Unidad 1: Requerimientos
Introducción. Requerimientos de Software. Tipos. Procesos de la Ingeniería de Requerimientos. Estudio de Factibilidad. Obtención y Análisis. Especificación. Modelado. Validación de Requerimientos. Gestión de los Requerimientos.

Unidad 2: Modelos Avanzados Orientados a Objetos en UML
Introducción. Modelos. Importancia de los modelos. Modelos estáticos. Modelos dinámicos. Persistencia. Concurrencia. Estado. Comportamiento. Mecanismos comunes. Estereotipos. Restricciones. Máquinas de Estado. Modelo Arquitectónico. Componentes.

Unidad 3: Proceso Unificado: Framework.
Introducción. Dirigido por Casos de Usos. Centrado en la Arquitectura. Iterativo e Incremental. Modelo de Casos de Usos. Captura de requisitos. Contexto del Sistema. Modelo del Dominio. Distintas Instanciaciones del Proceso. Análisis de la arquitectura. Relación con el Diseño. Componentes.

Unidad 4: Patrones de Diseño.
Introducción. Conceptos. Descripción. Selección de un patrón de Diseño. Utilización. Problema. Solución. Consecuencia. Catálogo de Patrones de Diseño. Patrones Creacionales. Patrones Estructurales. Patrones de Comportamiento. Patrón de diseño de acceso a datos.

Unidad 5: Proceso Unificado: Análisis y Diseño.
Introducción. Propósito. Diferencias. Artefactos. Modelo del Análisis. Arquitectura. Flujo de Trabajo. Rol del diseño. Artefactos. Modelo del Diseño. Subsistemas. Interfaz. Modelo de Desarrollo. Aplicación de Patrones en el Diseño.

Unidad 6: Ingeniería de la Información y Basada en Componentes.
Ingeniería de la Información. Clasificación. Modelado del área de Negocio. Modelo de Negocio. Notación de Modelado de Procesos. BPMN. Tecnología. Ingeniería de software basada en componentes. Reuso. Modelo de componentes. Desarrollo Basado en Componentes. Composición de Componentes. Arquitectura de Componentes.

Unidad 7: Seguridad para Ingenieros de Software
Definiciones. Dimensiones de Seguridad. Capas. Políticas de seguridad. Requisitos de seguridad. Casos de uso de seguridad. Seguridad Operativa. Diseño de sistemas seguros. Compromisos de Diseño. Evaluación de Riesgos de Diseño. Diseño Arquitectónico. Pautas de Diseño para Ingeniería de Seguridad. Pautas para una Programación de Sistemas Seguros. Autenticación. Basada en el objeto. Basada en el conocimiento. Cracking. Biometría.

Unidad 8: Metodologías Ágiles
Metodologías Ágiles. Concepto, diferencia con metodologías tradicionales. Manifiesto de desarrollo ágil. Principios.

VII - Plan de Trabajos Prácticos

Metodología de Enseñanza

Los trabajos prácticos correspondientes a las unidades del programa consisten en problemas que deben ser resueltos por los estudiantes guiados por los docentes. En cada trabajo práctico los estudiantes deben resolver ejercicios que están relacionados con la teoría que se ha dictado en ese momento. Los ejercicios están organizados por orden de complejidad comenzando por ejercicios más simples, luego los de complejidad media y por último los más complejos.
Las clases están pensadas para que el estudiante participe activamente respondiendo preguntas que el docente realiza las cuales pueden ser de teóricas o prácticas y resolviendo ejercicios en el pizarrón.

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

Laboratorio 1: Modelado Estático y Dinámico con UML. Ingeniería Directa e Ingeniería Inversa con Java.
Laboratorio 2: Uso de Herramientas CASE en Ingeniería Reversa de Modelos de Datos / Objetos
Laboratorio 3: Herramienta CASE para la construcción de Modelos de Negocio.
Laboratorio 4: Patrones de Diseño.

Práctico 1: Modelos Estáticos en UML. Ingeniería Directa e Ingeniería Inversa con Java.
Práctico 2: Modelos de Dominio.
Práctico 3: Modelos Dinámicos en UML. Ingeniería Directa e Ingeniería Inversa con Java.

Laboratorio: Construcción de un software orientado a objetos usando herramientas que automatizan el proceso de desarrollo generando los distintos artefactos desde los requerimientos hasta su implementación. Deberán aplicar los distintos conceptos aprendidos y utilizados en teoría, en los laboratorios y en las prácticas.

VIII - Regimen de Aprobación

La materia se desarrolla con la modalidad de promoción sin examen final. Existen dos niveles:
a) Regularización solamente: Para regularizar la materia se deberá:
1.- Tener como mínimo un 80% de asistencia a clases teóricas y prácticas.
2.- Tener los prácticos, solicitados por la cátedra, aprobados, como método aplicado por la cátedra para la evaluación continua del estudiante.
3.- Presentación y aprobación del proyecto integrador de laboratorio con nota mayor o igual a 7 (siete).
4.- Aprobar una evaluación parcial, o cualquiera de sus dos recuperaciones, con nota mayor o igual a 6 (seis).
b) Promoción sin examen final: Para regularizar y aprobar la materia se deberá:
1.- Cumplir con los requisitos a.1, a.2 y a.3.
2.- Aprobar la evaluación parcial, o cualquiera de sus dos recuperaciones, con una 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).
Aquellos estudiantes que sólo regularicen la materia deberán rendir un examen final, en los turnos establecidos.
Estudiantes Libres: Por las características propias del proyecto integrador de laboratorio a desarrollarse durante todo el cuatrimestre, no se aceptan estudiantes libres.

IX - Bibliografía Básica

[1] •Modern Software Engineering: Doing what works to build better software faster, David Farley, Addison-Wesley Professional, 2021.
[2] •Software Engineering: A Practitioners Approach, Roger Pressman; Mc Graw Hill, 9na Edición, 2019.
[3] •El Proceso de Desarrollo de Software Unificado. Ivar Jacobson, Grady Booch, and James Rumbaugh, Addison-Wesley, 1999.
[4] •The Unified Modeling Language User Guide, 2nd Edition. Ivar Jacobson, Grady Booch, and James Rumbaugh, Addison-Wesley, 2005.
[5] •Software Engineering, Ian Sommerville, Addison Wesley; 9na edition, 2011.
[6] •Software engineering, Ian Sommerville, Pearson, 10th edition, 2016.
[7] •Hands-On Design Patterns with Java: Learn design patterns that enable the building of large-scale software architectures, Edward Lavieri, Packt Publishing Ltd, 2019.

[8] •Design Patterns: Elements of Reusable Object-Oriented Software. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Addison-Wesley. 1995.
[9] •Security for Software Engineers.James Helfrich. CRC Press. 2019.
[10] •Java™ Coding Guidelines: 75 Recommendations for Reliable and Secure Programs, Fred Long et al., Addison Wesley, 2014
[11] •Application Security Program Handbook: A guide for software engineers and team leaders, Derek Fisher, Version 2, Manning Publications, 2021
[12] •Agile modeling: effective practices for extreme programming and the unified process. Scott Ambler. John Wiley & Sons, 2002.
[13] •Engineering software products. Ian Sommerville. Vol. 355. London, UK: Pearson, 2020.

X - Bibliografia Complementaria

[1] •Patterns in Java. Volume 1. A Catalog of Reusable Design Patterns Illustred with UML. Mark Grand. John Wiley & Sons Inc. 1998.
[2] •UML Semantics. http://www.omg.org
[3] •UML Notation Guide. http://www.omg.org
[4] •UML y Patrones: Introducción al análisis y diseño orientado a objetos. Craig Larman, Prentice Hall, 1999.
[5] •The Unified Modeling Language Reference Manual, 2nd Edition. Booch, Rumbaugh, Jacobson. Addison-Wesley, 2005.
[6] •Component-Based Software Engineering: Putting the Pieces Together, George T. Heineman, William T. Councill, Addison-Wesley Professional, 2001.

XI - Resumen de Objetivos

Introducir al estudiante en el desarrollo de sistemas aplicando métodos de desarrollo que permiten producir software de manera fiable y que funcione en máquinas reales cubriendo las distintas etapas del proceso de desarrollo.

XII - Resumen del Programa

Requerimientos
Modelos Avanzados Orientados a Objetos en UML
Proceso Unificado: Framework
Patrones de Diseño
Proceso Unificado: Análisis y Diseño.
Ingeniería de la Información y Basada en Componentes.
Seguridad para Ingenieros de Software
Metodologías Ágiles

XIII - Imprevistos

XIV - Otros

Contacto con la cátedra: cabdelah@email.unsl.edu.ar

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