

INGENIERÍA DE SOFTWARE II

Año 2026

Carrera/Plan:

Licenciatura en Sistemas, Planes 2015-2021

Licenciatura en Informática, Planes 2015-2021

Analista Programador Universitario, Planes 2015-2021

Analista en Tecnologías de la Información y la Comunicación Plan 2017-2021

Año: 3º**Régimen de Cursada:** Semestral**Carácter:** Obligatoria**Correlativas:** Ingeniería de Software I**Profesores:** Lic. Patricia Pesado

Dr. Alejandro González

Lic. Marcos Boracchia

Lic. Rocío Muñoz

Hs. semanales de teoría: 3 hs.**Hs. semanales de práctica:** 3 hs.**FUNDAMENTACIÓN**

Ingeniería de Software II, es una materia obligatoria correspondiente al primer semestre del tercer año de las carreras de Licenciatura en Informática, Licenciatura en Sistemas, Analista Programador universitario y Analista en TIC. Tiene como correlativas Ingeniería de Software I y Taller de lecto-comprensión y Traducción en Inglés.

En la materia los estudiantes reafirman los conocimientos brindados en Ingeniería de Software I, tales como calidad, modelos de procesos, metodologías ágiles y se amplían, abordando temas de gestión de proyectos.

Estos temas se aplican en la práctica, logrando identificar el alcance de dichos temas fortaleciendo su entendimiento y logrando la vinculación entre sí.

OBJETIVOS GENERALES

Continuar con los temas desarrollados en Ingeniería de Software I, a partir del diseño de sistemas de software. Introducir los conceptos de reingeniería e ingeniería inversa. Estudiar los temas de gestión, planificación y evaluación de proyectos de software, incluyendo el análisis de riesgo. El alumno deberá desarrollar sistemas concretos utilizando las metodologías/herramientas estudiadas.

RESULTADOS DE APRENDIZAJE

1.2. Describir las características de los últimos avances en hardware y software y sus correspondientes aplicaciones prácticas (Básico).

1.3. Describir los avances informáticos actuales e históricos y demostrar cierta visión sobre tendencias y avances futuros (Básico).

- 1.4. Aplicar e integrar conocimientos de otras disciplinas informáticas como apoyo al estudio de la propia área de especialidad (o áreas de especialidad) (Básico).
- 1.5. Demostrar sensibilización ante la necesidad de contar con amplios conocimientos a la hora de crear aplicaciones informáticas en otras áreas temáticas (Básico).
- 2.2. Describir un determinado problema y su solución a varios niveles de abstracción (Adecuado).
- 2.3. Seleccionar y utilizar los correspondientes métodos analíticos, de simulación y de modelización (Adecuado).
- 2.4. Escoger los patrones de solución, algoritmos y estructuras de datos apropiados (Adecuado).
- 2.5. Analizar la medida en la que un determinado sistema informático cumple con los criterios definidos para su uso actual y desarrollo futuro (Adecuado).
- 3.1. Definir y diseñar hardware/software informático/de red que cumpla con los requisitos establecidos (Básico).
- 3.2. Describir las fases implicadas en distintos modelos de ciclo de vida con respecto a la definición, construcción, análisis y puesta en marcha de nuevos sistemas y el mantenimiento de sistemas existentes (Básico).
- 3.3. Elegir y utilizar modelos de proceso adecuados, entornos de programación y técnicas de gestión de datos con respecto a proyectos que impliquen aplicaciones tradicionales, así como aplicaciones emergentes (Básico).
- 3.4. Describir y explicar el diseño de sistemas e interfaces para interacción persona-ordenador y ordenador-ordenador (Adecuado).
- 4.3. Identificar los requisitos jurídicos que rigen las actividades informáticas, incluyendo la protección de datos, derechos de propiedad intelectual, contratos, cuestiones de seguridad del producto y responsabilidad, cuestiones personales y riesgos laborales (Básico).
- 4.4. Explicar la importancia de la confidencialidad de la información y cuestiones relativas a la seguridad con respecto al diseño, desarrollo, mantenimiento, supervisión y uso de sistemas informáticos (Básico).
- 5.1. Demostrar conocimientos sobre los códigos y estándares de cumplimiento del sector (Adecuado)
- 5.2. Describir y explicar las técnicas de gestión correspondientes al diseño, implementación, análisis, uso y mantenimiento de sistemas informáticos, incluyendo gestión de proyectos, de configuración y de cambios, así como las técnicas de automatización correspondientes (Adecuado).
- 5.3. Identificar los riesgos, incluyendo riesgos de seguridad, laborales, medioambientales y comerciales y llevar a cabo una evaluación de riesgos, reducción de riesgos y técnicas de gestión de riesgos (Adecuado).
- 6.1. Organizar su propio trabajo de manera independiente demostrando iniciativa y ejerciendo responsabilidad personal (Básico).
- 6.2. Comunicar mensajes de forma efectiva tanto oralmente como por medio de otros medios de comunicación ante distintas audiencias (Básico).
- 6.3. Planificar su propio proceso de aprendizaje autodidacta y mejorar su rendimiento personal como base de una formación y un desarrollo personal continuos (Básico).
- 6.4. Identificar las distintas maneras de organizar equipos y los distintos roles dentro de dichos equipos (Adecuado).
- 6.5. Participar de manera efectiva en grupos de trabajo informático (Adecuado).

COMPETENCIAS

- CGS1- Desempeñarse de manera efectiva en equipos de trabajo, con capacidad para organizarlos y liderarlos.
- CGS2- Comunicarse con efectividad en forma oral y escrita.

- CGS4- Aprender en forma continua y autónoma, con capacidad de planificar este aprendizaje.
- CGS6- Capacidad para interpretar la evolución de la Informática con una visión de las tendencias tecnológicas futuras.
- CGT1- Identificar, formular y resolver problemas de Informática.
- CGT3- Gestionar, planificar, ejecutar y controlar proyectos de Informática
- CGT8 Capacidad de interpretación y resolución de problemas multidisciplinares, desde los conocimientos de la disciplina informática.
- CGT9 Capacidad de organización de equipos de trabajo en proyectos de Informática, definiendo los roles en los mismos.
- LI - CE4 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real, especificación formal de los mismos, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software/sistemas de información que se ejecuten sobre equipos de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.
- LI - CE6 – Controlar las normas de calidad en el software o software integrado a otros componentes. Capacidad de evaluación de performance de sistemas de software y sistemas que integren hardware y software.
- LI - CE7 – Planificar, dirigir, realizar y/o evaluar proyectos de sistemas de seguridad en el almacenamiento y procesamiento de la Información. Especificación, diseño, desarrollo, implementación y mantenimiento de los componentes de seguridad de información embebidos en los sistemas físicos y en los sistemas de software de aplicación. Establecimiento y control de metodología de procesamiento de datos que mejoren la seguridad y privacidad incluyendo datawarehousing.
- LI- CE8 – Efectuar las tareas de Auditorías de los Sistemas Informáticos. Realizar arbitrajes, peritajes y tasaciones relacionados con los Sistemas Informáticos.
- LS - CE1 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real. Especificación formal, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software que se ejecuten sobre sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.
- LS - CE5 – Establecer métricas y normas de calidad y seguridad de software, contralando las mismas a fin de tener un producto industrial que respete las normas nacionales e internacionales. Control de la especificación formal del producto, del proceso de diseño, desarrollo, implementación y mantenimiento. Establecimiento de métricas de validación y certificación de calidad. Capacidad de evaluación de performance de sistemas de software y sistemas que integren hardware y software.
- LS - CE6 – Planificar, dirigir, realizar y/o evaluar los sistemas de seguridad en el almacenamiento y procesamiento de la información. Realizar la especificación, diseño, desarrollo, implementación y mantenimiento de los componentes de seguridad de información embebidos en los sistemas físicos y en los sistemas de software aplicados. Establecer y controlar las metodologías de procesamiento de datos orientadas a seguridad, incluyendo data-warehousing.
- LS - CE7 – Efectuar las tareas de auditoría de los sistemas informáticos. Realizar arbitrajes, pericias y tasaciones relacionados con los sistemas informáticos

CONTENIDOS MINIMOS (de acuerdo al Plan de Estudios)

- Diseño e Implementación.
- Verificación y validación.
- Mantenimiento.
- Interacción hombre-máquina.
- Reingeniería e ingeniería inversa.
- Gestión de proyectos. Planificación. Métricas.
- Estimaciones. Análisis y gestión del riesgo.
- Conceptos de Auditoría y Peritaje.

PROGRAMA ANALÍTICO

1. Gestión de Proyectos

- Conceptos. El problema de las 4 "P" (personal, producto, proceso, proyecto). Actividades de gestión, planificación del proyecto, hitos y entregas. El plan de proyecto.
- Métricas y Estimaciones.
 - Clasificación de las métricas. Métricas del proceso y del proyecto. Métricas orientadas al tamaño, a la función, a casos de uso. Recopilación, cálculo y evaluación de métricas.
 - Estimación de proyectos. Técnicas de descomposición. Modelos empíricos
- Planificación Temporal: calendarización del proyecto, distribución del esfuerzo, redes de tareas, seguimiento de la planificación. Métodos PERT, Gantt.
- Planificación Organizativa: del equipo y del proyecto.
- Gestión del Riesgo: identificación de riesgos, proyección, impacto, reducción, supervisión y gestión. Planes de contingencia. El plan de RSGR.
- Gestión de la configuración del software: Línea base, gestión del cambio, control de versiones, auditoría.

2. Diseño

- Conceptos. Abstracción, arquitectura, patrones, modularidad, ocultamiento de la información, independencia funcional, cohesión, acoplamiento, refinamiento.
- El modelo de diseño: diseño de datos, diseño arquitectónico, diseño de interfaz, diseño al nivel de componentes.
- Diseño Arquitectónico.
 - Organización del sistema: modelo de repositorio, modelo cliente-servidor, Modelo de capas. Arquitecturas de Sistemas Distribuidos: multiprocesador,

c-s, objetos distribuidos, interorganizacional (peer-to-peer, sistemas orientados a servicios).

- Descomposición modular: orientada a objetos, orientada a flujos de funciones.
- Control: centralizado, dirigido por eventos.
- Diseño de interfaces de usuario: interacción del usuario, presentación de la información, análisis del usuario, prototipo de la interfaz, evaluación de la interfaz. Concepto de Diseño de Experiencias de Usuario
- Diseño a nivel de componentes: notaciones gráficas, notaciones tabulares, lenguajes de diseño.
- Características de un buen diseño. Técnicas para la mejora del diseño. Evaluación y validación del diseño. Documentando el diseño.

3. Implementación

- Estándares de programación y procedimientos
- Pautas para la programación
- Documentación

4. Verificación y Validación

- Técnicas de Prueba
 - Pruebas de Caja blanca: camino básico, bucles.
 - Pruebas de Caja negra: partición equivalente, análisis de valores límites.
- Estrategias de Prueba
 - Defectos y fallas. Planificación. Diseño de casos de prueba. Resultados. Documentación de las pruebas. Automatización.
 - Pruebas de unidad (arquitecturas convencionales y arquitecturas orientadas a objetos)
 - Pruebas de integración (arquitecturas convencionales y arquitecturas orientadas a objetos)
 - Pruebas de validación: alfa y beta.
 - Pruebas del sistema: de recuperación, de seguridad, de resistencia, de desempeño.
 - Pruebas de regresión.
 - La depuración: proceso, estrategia, corrección del error.

5. Entrega

- Entrenamiento
- Documentación

6. Mantenimiento

- Evolución del software. Tipos de mantenimiento: correctivo, adaptativo, perfectivo, preventivo.
- Sistemas heredados.
- Métricas, técnicas y herramientas para el mantenimiento.
- Rejuvenecimiento del software: redocumentación, reestructuración, ingeniería inversa, reingeniería.

7. Auditoría y Peritaje

- Conceptos
- Objetivos
- Planeamiento de Auditoría

BIBLIOGRAFÍA OBLIGATORIA

- ✓ Ingeniería de Software. 10ma Edición. Ian Sommerville. Pearson. 2016.
- ✓ Software Engineering: Theory and Practice. 4th Edition. Shari Pfleeger. Prentice Hall. 2010s. (Edición en castellano: Ingeniería de Software. Teoría y Práctica. Shari Pfleeger. Pearson Education. 2002)
- ✓ Ingeniería de Software. Un enfoque práctico. 9na Edición. Roger Pressman. McGraw- Hill. 2021.
- ✓ Auditoria en Informática 2da Edición. José Antonio Echenique García. McGraw-Hill. 2001.

BIBLIOGRAFÍA COMPLEMENTARIA

- ✓ Systems Analysis and Design, 9/E. Kendall & Kendall. Pearson. 2013. (Edición en castellano: Análisis y diseño de sistemas. 8va Edición. Kendall & Kendall. Pearson. 2011)
- ✓ Análisis de Sistemas. Diseño y Métodos. 7ma Edición. Whitten y Bentley. 2008.

METODOLOGÍA DE ENSEÑANZA

El curso consta de clases teóricas, explicaciones de práctica y clases prácticas. La asignatura utiliza la plataforma *MOODLE* para interactuar con los alumnos del curso.

Durante la cursada, grupos de alumnos desarrollan un proyecto aplicable a un dominio particular, que es monitoreado por un docente de la cátedra a través de la plataforma y las consultas en las clases prácticas. Los proyectos tienen estipulados un conjunto de entregas y reentregas y un coloquio integrador fijado en el calendario.

La cátedra organiza actividades por equipos de trabajo, con un número variable de alumnos (entre 4 y 5) los cuales tienen roles específicos. En la evaluación del trabajo en equipo, hay presentaciones donde se responden aspectos generales de las tareas asignadas y respuestas/explicaciones individuales que permiten calificar diferentes aptitudes de los miembros del equipo (conocimientos / modo de expresarse / predisposición al trabajo colaborativo).

La cátedra evalúa a los alumnos en coloquios en los cuales deben exponer la tarea realizada en forma individual y donde el docente evalúa no sólo los conocimientos sino la claridad de la presentación, su organización y la forma de expresión. Esto se refleja en planillas escritas que conforman documentación de evaluación del coloquio. En estas planillas se indican puntos relacionados con cuestiones de seguridad y de riesgo asociados al proyecto, así como puntos vinculados con expresión oral y escrita.

La cátedra organiza actividades planificadas para los alumnos, en las que se les propone investigar el dominio del proyecto presentado, buscando posibles alternativas de solución del problema, teniendo en cuenta la tecnología existente y la evolución de la misma, así como los riesgos de cada solución; y generando la documentación necesaria para llevar a cabo la implementación del proyecto. Presentando el trabajo realizado en un coloquio.

Eventualmente por razones organizativas alguno de los coloquios puede ser reemplazado por una evaluación escrita.

EVALUACIÓN

Los alumnos obtienen la cursada aprobando las entregas del proyecto y un coloquio integrador.

Para la aprobación final de la asignatura los alumnos tienen dos posibilidades:

Alumnos por promoción:

- Deben concurrir al 80% de las clases teóricas.
- Deben desarrollar una actividad de teoría en forma grupal y rendir y aprobar una evaluación teórica, ambas actividades se aprueban con nota 6 o superior (con una recuperación).
- Deben rendir y aprobar un coloquio (de la práctica).

Alumnos regulares:

Deben rendir y aprobar un examen final.

CRONOGRAMA DE CLASES Y EVALUACIONES (Tentativo)

Semana	Teoría	Práctica
09-mar	Presentación de la materia Repaso de Elicitación Requerimientos Documentos de Especificación de Sistema y de Requerimientos	
16-mar	GCS-Planificación Organizativa	Entrevista 1
23-mar	Feriado	Entrevista 2
30-mar	Feriado	Entrega 1: Entrevistas + Cuestionario + Épicas
6-abr	Riesgos	Consulta Entrega 2
13-abr	Planificación Temporal-Diseño	Consulta Entrega 2
20-abr	Diseño de Interfaces	Entrega 2: Documentación + Pila de producto
27-abr	Feriado	Revisión pila + Planificación Sprint 1
04-may	Entrega TP Teoría Diseño Arquitectónico	Scrum diario 1
11-may	Métricas	Scrum diario 1
18-may	Pruebas 1	Scrum diario 1 + Predemo
25-may	Pruebas 2	Demo 1
01-jun	Mantenimiento	Planificación Sprint 2
08-jun	Recuperatorio TP Teoría Auditoría	Scrum diario 2
15-jun	Consultas	Scrum diario 2
22-jun	Examen Teórico	Scrum diario 2 + Predemo
29-jun	Corrección teoría	Demo 2
06-jul	Muestra y consulta Martes y Miércoles	
13-jul	Recuperatorio Examen Teórico	
20-jul	Receso Invernal	
27-jul	Receso Invernal	
03-ago	Muestra recuperatorio	Coloquio integrador

Evaluaciones Previstas	Semana
Entrega TP Teoría	04-may
Demo 1	25-may
Recuperatorio TP Teoría	08-jun
Examen Teórico	22-jun
Demo 2	29-jun
Recuperatorio Examen Teórico	13-jul
Coloquio integrador	03-ago

Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):

Los alumnos pueden enviar consultas a la cuenta **marcosb@lidi.info.unlp.edu.ar**.
Se utiliza como plataforma virtual/ *MOODLE* <https://asignaturas.info.unlp.edu.ar/>
(mensajería, página y gestión del curso).

Firma del/los profesor/es

Alejandro H. Gonzalez

Marcos Boracchia

Rocío Muñoz