

**Carrera/ Plan:****INGENIERIA DE SOFTWARE***Ingeniería en Computación Plan 2024***Año 2026****Año:** 3º**Régimen de Cursada:** Semestral**Carácter (Obligatoria/Optativa):** Obligatoria**Correlativas:** Programación II**Profesor/es:** Lic. Patricia Pesado

Dr. Ariel Pasini

**Hs. semanales teoría:** 3 Hs**Hs. semanales practica:** 6 Hs**FUNDAMENTACIÓN**

La asignatura Ingeniería de Software brinda al futuro profesional una visión general del proceso de desarrollo de Software, focalizándose en todas las etapas de proceso. En esta materia el alumno integra los conceptos de Requerimiento, obtención y análisis de requerimientos, Diseño, Implantación, verificación y validación, Mantenimiento, entre otros. Los aportes de Ingeniería de Software se focalizan en los conocimientos y habilidades necesarios para que el futuro profesional contribuya al proceso de desarrollo de software, generando valor agregado al producto y las partes interesadas.

**OBJETIVOS GENERALES**

Introducir al alumno en los conceptos fundamentales de la Ingeniería de Software.

Profundizar las etapas del ciclo de vida (requerimientos, análisis, diseño, codificación, pruebas).

Introducir los conceptos de re-ingeniería e ingeniería inversa.

Estudiar los temas de gestión, planificación y evaluación de proyectos de software, incluyendo el análisis de riesgo. Presentar los conceptos de calidad de software.

El alumno deberá desarrollar sistemas concretos utilizando las metodologías/herramientas estudiadas

**EJES TRANSVERSALES**

03. Especificación, proyecto y Desarrollo de Software y Sistemas Conjuntos de Hardware y Software haciendo uso de conceptos, métodos y herramientas de gestión de proyectos, ingeniería de software, base de datos, experiencia del usuario, elicitación, análisis, especificación y validación de requerimientos. (3-Alto)

05. Sistemas de Gestión de Recursos de Hardware y Software a sistemas generales, de tiempo real, distribuidos, para dispositivos fijos y móviles. (1-Bajo)

06. Proyecto, desarrollo, dirección, control, construcción, operación y mantenimiento de Sistemas de Procesamiento de Señales, Sistemas Embebidos y sus periféricos incluido en software de soporte, Sistemas Computarizados de automatización y control y Sistemas Conjuntos de Hardware y Software. (2-Medio)

07. Certificación del funcionamiento, condición de uso o estados de Sistemas de Procesamiento de Señales, Sistemas Embebidos, Sistemas Computarizados de automatización y control, Sistemas Conjuntos de Hardware y Software. (2-Medio)

08. Proyecto, Dirección y Aseguramiento de la calidad en lo referido a Seguridad Informática. (2-Medio)

09. Identificación, formulación y resolución de problemas de ingeniería en computación. (2-Medio)

10. Concepción, diseño y desarrollo de proyectos de ingeniería en computación. (3-Alto)

11. Gestión, planificación, ejecución y control de proyectos de ingeniería en computación. (3-Alto)

12. Utilización de técnicas y herramientas de aplicación en la ingeniería en computación. (2-Medio)

13. Generación de desarrollos tecnológicos y/o innovaciones tecnológicas. (2-Medio)
14. Fundamentos para el desempeño en equipos de trabajo. (3-Alto)
15. Fundamentos para una comunicación efectiva. (1-Bajo)
16. Fundamentos para una actuación profesional ética y responsable. (1-Bajo)
18. Fundamentos para el aprendizaje continuo. (1-Bajo)
19. Fundamentos para el desarrollo de una actitud profesional emprendedora. (1-Bajo)

### **CONTENIDOS MINIMOS (de acuerdo al Plan de Estudios)**

- El proceso de software. Ciclos de vida de software.
- Herramientas para el proceso de software.
- Ingeniería de requerimientos. Introducción a los métodos formales.
- Metodologías de Análisis.
- 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 calidad de software.

### **PROGRAMA ANALÍTICO**

#### **1- Conceptos de software e ingeniería de software.**

- Evolución del software. Características. Componentes.
- Definición de Ingeniería de Software. Evolución.
- Software de alta calidad.
- Ingeniería de Sistemas.

#### **2- Procesos del Software.**

- El significado de proceso.
- Modelos de proceso. Modelo de cascada.
- Iteración de procesos. Modelos incrementales. Modelos Evolutivos.
- Prototipación. Metodologías ágiles. Desarrollo basado en componentes.
- Métodos formales. Desarrollo orientado a aspectos. Proceso unificado.
- Actividades del Proceso. Especificación. Diseño. Implementación. Validación. Evolución.
- Herramientas y técnicas para modelado de procesos.
- Ingeniería de Software Asistida por Computadora.

#### **3- Ingeniería de Requerimientos.**

- El proceso de requerimientos.
- Tipos de requerimientos. Requerimientos funcionales, no funcionales, del usuario, del sistema.
- Características de los requerimientos. Obtención y análisis de los requerimientos.
- Técnicas de comunicación. Los problemas de la comunicación. Elicitación de requisitos. Entrevistas, cuestionarios, JAD, brainstorming.
- Validación de requerimientos. Gestión de requerimientos. Medición de requerimientos.
- Documentos de Especificación de requerimientos. STD 830.

#### **4- Modelos del Sistema.**

- Modelo de contexto, de comportamiento, de datos, de objetos.

- Técnicas de especificación de requerimientos: estáticas, dinámicas, relacionales, orientadas a estados, formales.
- Tablas de Decisión, Diagramas de Transición de Estados, Redes de Petri...
- Casos de Uso, Diagramas de Flujos de Datos, Diagramas de Flujos de Control, UML.
- Prototipado de los requerimientos. Técnicas de construcción rápida.

## 5- 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 (COCOMO). Decisión de desarrollar-comprar.
- 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.

## 6- Diseño e Implementación

- 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.
- Diseño a nivel de componentes: notaciones gráficas, notaciones tabulares, lenguajes de diseño.
- Características de un bien diseño. Técnicas para la mejora del diseño. Evaluación y validación del diseño. Documentando el diseño.
- Aspectos de Implementación

## 7- Implementación

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

## 8- 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.

## 9- Entrega

- Entrenamiento
- Documentación

## 10- 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.

## 11- Calidad.

- Conceptos de Calidad y Calidad Total.
- Calidad del proceso y del producto.
- Aseguramiento y estándares de calidad.
- Planeamiento de la calidad.
- Control de la calidad.
- Revisiones Técnicas formales. Inspecciones.
- Modelos de madurez para las organizaciones de desarrollo de software (CMM-CMMI).
- Normas ISO 9000.

## BIBLIOGRAFÍA

- Ingeniería de Software. 9na Edición. Ian Sommerville. Pearson. 2011.
- Software Engineering: Theory and Practice. 4th Edition. Shari Pfleeger. Prentice Hall. 2009. (Edición en castellano: Ingeniería de Software. Teoría y Práctica. Shari Pfleeger. Pearson Education. 2002)
- Ingeniería de Software. Un enfoque práctico. 7ma Edición. Roger Pressman. McGraw-Hill. 2010.
- Sistemas de Información Administrativa. Murdick R. Prentice Hall. 1988
- Systems Analysis and Design, 9/E. Kendall & Kendall. Pearson. 2013. (Edición en castellano: Análisis y diseño de sistemas. 8va Edición. Kendall & Kendall. Pearson. 2011)

## METODOLOGÍA DE ENSEÑANZA

Las clases teóricas se dictan una vez por semana, la asistencia es obligatoria solo para los alumnos que estén inscriptos en el régimen de promoción.

Durante la práctica, los alumnos realizarán un conjunto de actividades individuales y un trabajo práctico en grupo de tres o cuatro alumnos, cada grupo tendrá un docente asignado y un horario de consulta por semana. La composición del grupo debe permanecer a lo largo de toda la cursada.

La cátedra se comunicará con los alumnos a través del curso de Ingeniería de Software de la Plataforma MFI.

---

## **EVALUACIÓN**

### **Aprobación de Cursada**

Los alumnos deberán presentar el trabajo práctico grupal, tener un 80% de asistencia a la práctica y aprobar (de forma directa o en su recuperatorio) un examen práctico individual.

### **Aprobación de la Asignatura**

Habiendo aprobado la cursada los alumnos podrán rendir el final (teórico-practico) de la materia durante todo el período de vigencia de la cursada.

### **Régimen de promoción**

Los alumnos podrán optar por el régimen de promoción, para lo cual deberán inscribirse al mismo, aprobar la cursada, cumplir con el 80% de asistencia a las clases teóricas y aprobar los exámenes de promoción que se tomarán en las clases teóricas con promedio de 7 (siete). En caso de no alcanzar un promedio de 7 (siete) podrán acceder a un recuperatorio integrador de los exámenes que han sido tomados. De no aprobar el examen y habiendo aprobado la cursada el alumno podrá aprobar la asignatura rindiendo el final antes mencionado.

La nota final obtenida en la asignatura se compondrá por un 40% de la nota del examen práctico y un 60% de la nota del examen teórico.

## CRONOGRAMA TENTATIVO DE CLASES Y EVALUACIONES

Inicio tentativo: 24 de agosto

Cronograma tentativo:

Semana	Fecha	Teoría	Práctica
1	Lunes, 24 de Agosto	Conceptos de Ingeniería de Software - Proceso del Software	
2	Lunes, 31 de Agosto	Ingeniería de Requerimiento - Técnicas de Elicitación	Introducción. Presentación de enunciado.
	Viernes, 4 de Septiembre		Introducción. Presentación de enunciado.
3	Lunes, 7 de Septiembre	GCS (GIT/SVN) Modelo de Sistema - Tablas - DTE -Petri - CU - HU -SRS	
	Viernes, 11 de Septiembre		Feriado
4	Lunes, 14 de Septiembre	Gestión de Proyectos Planificación - Riesgos	
	Viernes, 18 de Septiembre		
5	Lunes, 21 de Septiembre	Feriado	
	Viernes, 25 de Septiembre		
6	Lunes, 28 de Septiembre	Gestión de Proyectos Métricas – Calidad	
	Viernes, 2 de Octubre		
7	<b>Lunes, 5 de Octubre</b>	<b>CACIC / Examen teórico + Parcialito 1</b>	
	Viernes, 9 de Octubre		
8	Lunes, 12 de Octubre	Feriado	
	Viernes, 16 de Octubre		
9	Lunes, 19 de Octubre	Diseño - Diseño Arquitectónico + Interfaces	
	Viernes, 23 de Octubre		
10	Lunes, 26 de Octubre	Pruebas - Entrega Mantenimiento - Calidad	
	Viernes, 30 de Octubre		
11	Lunes, 2 de Noviembre	Consultas	
	Viernes, 6 de Noviembre		
12	<b>Lunes,9 de Noviembre</b>	<b>Examen teórico + Parcialito 2</b>	Inicio sprint
	Viernes, 13 de Noviembre		
13	Lunes, 16 de Noviembre	Feriado	
	Viernes, 20 de Noviembre	Feriado	
14	Lunes, 23 de Noviembre	Feriado	
	Viernes, 27 de Noviembre		
15	Lunes, 30 de Noviembre		Demo
	Viernes, 4 de Diciembre		Demo

16	Lunes, 7 de Diciembre	Feriado	
	<b>Viernes, 12 de Diciembre</b>		Parcial
17	Lunes, 14 de Diciembre		
	<b>Viernes, 18 de Diciembre</b>		Recuperatorio teórico y parcial

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

**Mail:** [apasini@lidi.info.unlp.edu.ar](mailto:apasini@lidi.info.unlp.edu.ar)

**Plataforma:** MFI.

Firma del/los profesor/es

Ariel Pasini