

**Carrera/ Plan:****INGENIERIA DE SOFTWARE***Ingeniería en Computación Plan 2008/Plan 2011***Año 2019****Año:** 3º**Régimen de Cursada:** *Semestral***Carácter (Obligatoria/Optativa):** Obligatoria**Correlativas:****Profesor/es:** Lic. Patricia Pesado

Mg Ariel Pasini

**Hs. semanales:** 9 Hs**FUNDAMENTACIÓN**

*Explicar brevemente la importancia de la asignatura para la formación del futuro profesional y el tipo de aporte específico que realizará la misma.*

**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

**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 buen 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 trabajo práctico, con múltiples entregas, en grupo de tres o cuatro alumnos, cada grupo tendrá un docente asignado y un horario de consulta fijo por semana con asistencia obligatoria para el grupo y un horario por semana de consulta adicional. 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 IDEAD y/o el blog de la cátedra.

## **EVALUACIÓN**

### **Aprobación de Cursada**

Los alumnos deberán aprobar 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 promedio obtenida de los exámenes, o la nota obtenida en el recuperatorio integrador será la nota final que el alumno habrá obtenido en la asignatura.

## CRONOGRAMA TENTATIVO DE CLASES Y EVALUACIONES

Semana	Fecha	Teoría	Explicación	Práctica
	miércoles, agosto 21, 2019	Conceptos de Ingeniería de Software - Proceso del Software		
1	lunes, agosto 26, 2019	Ingeniería de Requerimiento - Tecnicas de Elicitación		
	miércoles, agosto 28, 2019		Introducción + Elicitación + Instalación Rails	Brainstorming
	viernes, agosto 30, 2019		Entrevistas + Creación Repo	Entrevistas
2	lunes, septiembre 02, 2019	GCS (GIT/SVN) Modelo de Sistema - Tablas - DTE -Petri - SRS		
	miércoles, septiembre 04, 2019			
	viernes, septiembre 06, 2019		Entrevista	Entrevista Proyecto
3	lunes, septiembre 09, 2019	Modelo de Sistema - CU - US- DFD	Entrevista	Entrevista Proyecto
	miércoles, septiembre 11, 2019	Feriado		
	viernes, septiembre 13, 2019		Primera App y CRUD	Laboratorio 1
4	lunes, septiembre 16, 2019	Gestión de Proyectos Planificación - Riesgos		
	miércoles, septiembre 18, 2019		Modelos	Laboratorio 2
	viernes, septiembre 20, 2019		Vistas	Laboratorio 3
5	lunes, septiembre 23, 2019	Examen Teórico		
	miércoles, septiembre 25, 2019		Planificación	
	viernes, septiembre 27, 2019		Scrum + Pivotal Tracker	
6	lunes, septiembre 30, 2019	Gestión de Proyectos Metricas - Calidad		
	miércoles, octubre 02, 2019		Controllers	Laboratorio 4
	viernes, octubre 04, 2019		Rutas + Migraciones	Laboratorio 5
7	lunes, octubre 07, 2019	Diseño - Diseño Arquitectónico + Interfaces		
	miércoles, octubre 09, 2019		GIT	
	viernes, octubre 11, 2019		Heroku	
8	lunes, octubre 14, 2019	FERIADO		
	miércoles, octubre		Repaso Scrum	Competencia

	16, 2019			(juego)
	viernes, octubre 18, 2019		Formularios, Partials, Helpers	Laboratorio 7
9	lunes, octubre 21, 2019	Pruebas - Entrega		
	miércoles, octubre 23, 2019		Actividad Scrum	SCRUMIA
	viernes, octubre 25, 2019			Planificación Sprint 1
<b>INICIO PROYECTO</b>				
10	lunes, octubre 28, 2019	Mantenimiento - Calidad		
	miércoles, octubre 30, 2019			Planificación Sprint 1
	viernes, noviembre 01, 2019			Consulta
11	lunes, noviembre 04, 2019	Consulta		
	miércoles, noviembre 06, 2019			Consulta
	viernes, noviembre 08, 2019			Demo 1 + Planificación 2
12	lunes, noviembre 11, 2019	Segundo Examen Teórico		
	miércoles, noviembre 13, 2019			Demo 1 + Planificación 2
	viernes, noviembre 15, 2019			Consulta
13	miércoles, noviembre 20, 2019			Consulta
	viernes, noviembre 22, 2019			Demo 2 + Planificación 3
	miércoles, noviembre 27, 2019			Demo 2 + Planificación 3
14	viernes, noviembre 29, 2019			Consulta
	lunes, diciembre 02, 2019	Examen Teórico Integrador		
	miércoles, diciembre 04, 2019			Consulta
15	viernes, diciembre 06, 2019			Demo 3
	viernes, diciembre 13, 2019			Examen compensatorio

Evaluaciones previstas	Fecha
Examen Teórico	lunes, septiembre 23, 2019

Entrega - Entrevistas + SRS + Épicas	miércoles, septiembre 25, 2019
Entrega - Planificación + Pila + DER	miércoles, octubre 16, 2019
Demo 1	viernes, noviembre 08, 2019
Segundo Examen Teórico	lunes, noviembre 11, 2019
Demo 2	viernes, noviembre 22, 2019
Examen Teórico Integrador (Rec)	lunes, diciembre 02, 2019
Demo 3	viernes, diciembre 06, 2019

**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)

La cátedra se comunicará con los alumnos a través del curso de Ingeniería de Software de la Plataforma IDEAD y/o el blog de la cátedra.

Firma del/los profesor/es