



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

INGENIERÍA DE SOFTWARE II

Año 2015

Carrera/Plan:

Licenciatura en Sistemas, Planes 2003-2007-2011-2015

Licenciatura en Informática, Planes 2003-2007-2011-2015

Analista Programador Universitario, Planes 2007-2011-2015

Año: 3º

Régimen de Cursada: Semestral

Carácter: Obligatoria

Correlativas: Ingeniería de Software I

Profesores: Lic. Patricia Pesado

Mg. Silvia Esponda

Mg. Ariel Pasini

Mg. Alejandro González

Lic. Marcos Boracchia

Hs. semanales: 6 hs.

OBJETIVOS GENERALES:

Continuar con los temas desarrollados en Ingeniería de Software I, a partir del diseño de sistemas de software. 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. El alumno deberá desarrollar sistemas concretos utilizando las metodologías/herramientas estudiadas.

CONTENIDOS MÍNIMOS:

- 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 (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.

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.
- 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.



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



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

- 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

METODOLOGÍA DE ENSEÑANZA

El curso consta de clases teóricas, explicaciones de práctica y clases prácticas.

La asignatura utiliza la plataforma WebUNLP para interactuar con los alumnos del curso.

Durante la cursada, grupos de 3 / 4 alumnos desarrollan un proyecto, 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 fijadas en el calendario.

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 rendir y aprobar dos evaluaciones teóricas con nota 6 o superior (cada una de ellas con una recuperación).

Deben rendir y aprobar un coloquio.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Alumnos regulares:

Deben rendir y aprobar un examen final.

BIBLIOGRAFÍA OBLIGATORIA

- ✓ 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.
- ✓ 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.



CRONOGRAMA TENTATIVO DE CLASES Y EVALUACIONES 2015

Semana	Teoría	Práctica	Demo
9-mar	Presentación de la materia Repaso de Elicitación Requerimientos Documentos de Especificación de Sistema (1362) y de Requerimientos (830)	Consulta Entrega 1	
16-mar	Planificación Temporal	Consulta Entrega 1	
23-mar		Consulta Entrega 1	
30-mar	Riesgos	Consulta Entrega 1	
6-abr	Interfaces	Entrega 1 + Consulta Entrega 2	
13-abr	Métricas	Consulta Entrega 2	
20-abr	Pruebas	Entrega 2 + Consulta Entrega 3	
27-abr	Pruebas	Consulta Entrega 3	
4-may		Entrega 3 + Consulta Entrega 4	
11-may	Examen Teórico 1	Consulta Entrega 4	
18-may	Diseño	Entrega + Planificación Sprint 1	
25-may	Diseño Arquitectónico	Scrum diario 1	
1-jun	GCS-Mantenimiento	Scrum diario 1	Demo 1
8-jun		Planificación Sprint 2	
15-jun	Examen Teórico 2	Scrum diario 2	
22-jun		Scrum diario 2	Demo 2
29-jun		Planificación Sprint 3	
6-jul		Scrum diario 3	
13-jul		Scrum diario 3	Demo 3
20-jul	Receso invernial		
27-jul	Receso invernial		
3-ago			Coloquio integrador + Manual de usuario
10-ago	Recuperatorio Examen Teórico 1		
24-ago	Recuperatorio Examen Teórico 2		

**CONTACTO DE LA CÁTEDRA (MAIL, PÁGINA, PLATAFORMA VIRTUAL DE
GESTIÓN DE CURSOS):**



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

Los alumnos pueden enviar consultas a la cuenta is2@info.unlp.edu.ar.
Se utiliza como plataforma virtual la plataforma provista por la Universidad: WebUNLP
(mensajería, página y gestión del curso) y Blogs (<http://blogs.unlp.edu.ar/ingenieria2/>).

FIRMA/S DEL/LOS PROFESORES RESPONSABLE/S: