

ORGANIZACIÓN DOCENTE del curso 2009-10

1. DATOS GENERALES DE LA ASIGNATURA

NOMBRE	Ingeniería del Software II		PÁGINA WEB	www.ctr.unican.es/asignaturas/is2	
CÓDIGO					
DEPARTAMENTO	Matemáticas, Estadística y Computación				
PLAN DE ESTUDIOS	Ingeniero en Informático		CURSO	2009/2010	
PROFESORADO	<u>Nombre</u>		<u>e-mail</u>		
	Michael González Harbour		mgh@unican.es		
	Pablo Sánchez Barreiro		p.sanchez@unican.es		
	Carlos Blanco Bueno		carlos.blanco@unican.es		
	Juan Hernández Marqués		juan.hernandez@unican.es		
CRÉDITOS ALUMNO	<u>Teóricos</u> (1)	<u>Prac. Problemas</u> (2)	<u>Prac. Laboratorio</u>	<u>Prac. Computador</u>	TOTALES
	4.5	1.5		3.0	9.0
LUGAR DE IMPARTICIÓN	<u>Teóricos</u>	<u>Prac. Problemas</u>	<u>Prac. Laboratorio</u>	<u>Prac. Computador</u>	
	Aula	Aula		Aula PCs	
HORARIO PREVISTO(*)	<u>Teóricos</u>	<u>Prac. Problemas</u>	<u>Prac. Laboratorio</u>	<u>Prac. Computador</u>	
(*) Observaciones:	Consultar el cuadro que se expone en el tablón de anuncios				

(1) Se corresponde con clases magistrales de teoría en aula

(2) Se corresponde con clases prácticas (problemas, experiencias de cátedra,...) en aula

2. PROGRAMA DE LA ASIGNATURA

TEORÍA:

0. Presentación de la Asignatura

Objetivos. Papel en la profesión. Métodos de Trabajo. Evaluación.

Parte A – Procesos de Ingeniería del Software

1. Construcción y Pruebas

Construcción del Software. Estrategias de Prueba. Pruebas de Unidad. Pruebas de Integración. Pruebas de Validación. Diseño de Casos de Prueba. Métodos de Prueba: Nivel de Clase; Nivel Entre-Clases.

2. Mantenimiento

Concepto de Mantenimiento de Software. Tipos de Mantenimiento. Actividades de Mantenimiento – Norma ISO 14764. Técnicas: Reestructuración; Ingeniería Inversa; Rediseño; Reingeniería.

3. Gestión de la Configuración

Problemas organizativos durante el desarrollo de software: Visibilidad; Control de Cambios; Trazabilidad; Responsabilidad; Coordinación. Concepto de Configuración Software. El Proceso de Gestión de la Configuración del Software: Identificación; Control; Contabilidad de Estados; Auditoría; Despliegue y Distribución.

4. Calidad del Software

Concepto y Factores de Calidad del Software – Norma ISO 9126. Niveles de Calidad. Evaluación y Mejora de los Procesos. Medición de los Productos y Procesos Software. Aseguramiento de la Calidad. Verificación y Validación del Software. Técnicas de Verificación y Validación: Revisiones; Inspecciones; Auditorías.

Parte B – Gestión de Proyectos Software

5. Fundamentos de Gestión de Proyectos

Cuerpo de Conocimientos de la Gestión de Proyectos – PMBOK. Conceptos básicos. Grupos de Procesos. Áreas de Trabajo. Interacciones entre procesos.

6. Gestión de la Integración y el Alcance

Planificación de Proyectos. Plan de proyectos Software – IEEE 1058. Control Global de los Cambios. Iniciación de un Proyecto. Análisis de Necesidades. Estudio de Viabilidad. Definición del Alcance: Estructuras de Descomposición de Trabajos; Diagramas de Flujos de Trabajo.

7. Gestión del Tiempo

Definición de Actividades. Secuenciación: Diagramas en Red. Estimación de la duración. Desarrollo del calendario: Método PERT; Método CPM. Representaciones del calendario.

8. Gestión de Riesgos

Planificación de los riesgos: Identificación; Análisis; Priorización; Planificación de respuestas. Supervisión y control de riesgos.

9. Gestión de Costes

Planificación de recursos. Estimación de costes. Elaboración de presupuestos y control de gastos. Tipos de técnicas para estimación del software. Estimación del tamaño mediante Puntos Función. Método COCOMO para estimación del esfuerzo y el coste.

10. Aspectos Adicionales de los Proyectos Software

Capacidades y Habilidades Requeridas. Roles. Gestión de los Recursos Humanos. Gestión de las Comunicaciones. Gestión de las Adquisiciones. Aspectos legales de los sistemas software.

PRÁCTICAS:

P1. Pruebas

P2. Mantenimiento de un Software Heredado

P3. Gestión de Configuración

P4. Medición de propiedades del software

P5. Planificación de un Proyecto

P6. Seguimiento de un Proyecto

P7. Estimación de Esfuerzo y Costes del Software

Asignaturas que se recomienda al alumno haber cursado o estar cursando

Pre-requisitos:

Ingeniería del software I

Co-requisitos:

-

Se recomienda fuertemente realizar esta asignatura el mismo curso de manera conjunta con su predecesora, Ingeniería del Software I.

3. OBJETIVOS GENERALES DE LA ASIGNATURA

Los objetivos generales son:

Continuar la asignatura de Ingeniería del Software I para comprender y conocer mejor la importancia y papel de la disciplina de Ingeniería del Software, de cara a la correcta aplicación de los principios generales de la ingeniería al problema de hacer software. Saber planificar y gestionar proyectos de desarrollo y/o mantenimiento de software.

4. OBJETIVOS ESPECIFICOS: APTITUDES/DESTREZAS

Los anteriores objetivos generales se concretan en:

CONOCER:

- Métodos para pruebas de sistemas orientados a objetos.
- Las características que diferencian el mantenimiento del desarrollo del software.
- La utilidad y ventajas de realizar gestión de configuración.
- El concepto y la importancia de la calidad del software, en sus dos dimensiones, de producto y de proceso.
- El estándar PMBOK de gestión de proyectos en general.
- Algunas técnicas especialmente útiles para gestión de proyectos software en las áreas de: alcance, integración, tiempo, riesgos y costes.

SABER:

- Hacer pruebas de software orientado a objetos.
- Aplicar técnicas de ingeniería inversa y reingeniería para mantenimiento de software.
- Desplegar versiones de una aplicación software usando alguna herramienta de gestión de configuración del software.
- Preparar un plan de proyecto software siguiendo algún estándar internacional.
- Utilizar una herramienta de gestión de proyectos.
- Realizar un calendario detallado de proyecto.
- Planificar y controlar riesgos.
- Hacer estimaciones de tamaño y esfuerzo del software.
- Trabajar en equipo para llevar a cabo un proyecto.

5. BIBLIOGRAFÍA

Básica

Generales:

- Piattini et al., 2007. Análisis y diseño de Aplicaciones Informáticas de Gestión. Una perspectiva de Ingeniería del Software. Ra-Ma. Junio 2007.
- Pressman, 2005. Ingeniería del Software: Un Enfoque Práctico. 6ª Edición. McGraw-Hill, 2005.
- Pfleeger, 2002. Ingeniería del Software. Teoría y Práctica. Prentice Hall, 2002.
- Sommerville, 2005. Ingeniería del Software. 7ª Edición, Addison-Wesley. Julio 2005.

Complementaria

- Jacobson et al., 2000. El Proceso Unificado de Desarrollo de Software. Addison-Wesley.
- McConnell, 1997. Desarrollo y Gestión de Proyectos Informáticos. McGraw-Hill Interamericana. España.
- Piattini et al., 2000. Mantenimiento del Software: Modelos, técnicas y métodos para la gestión del cambio. Ra-Ma.
- Piattini et al., 2007. Calidad de los Sistemas Informáticos. Ra-Ma.

ESTANDARES:

- Project Management Institute, 2004. A Guide to the Project Management Body of Knowledge, 3th edition. PMI Communications, USA 2004.
- ANSI/IEEE, 1998. Std. 1058: IEEE Standard for Software Project Management Plans. IEEE Computer Society, USA. Existe una versión en castellano en: http://alarcos.inf-cr.uclm.es/doc/pgsi/doc/especiales.htm#IEEE_1058.

- ISO/IEC, 2001: ISO/IEC 9126-1:2001 Software Engineering - Product Quality -- Part 1: Quality Model.
- ISO/IEC, 2006: ISO/IEC 14764:2006 Software Engineering - Software Life Cycle Processes – Maintenance.

6. ACTIVIDADES A DESARROLLAR EN LA ASIGNATURA

Los tipos de actividades a desarrollar serán los siguientes:

- Clases magistrales para los temas de teoría.
- Clases de ejercicios y problemas.
- Realización y entrega de ejercicios voluntarios.
- Trabajos en grupo de 3/4 alumnos para el tema de teoría 10 y defensa en la clase.
- Prácticas de laboratorio en grupo de 3/4 con ordenador.
- Pruebas de evaluación parciales.
- Tutorías.
- Análisis y debate de artículos.

7. MÉTODO DE EVALUACIÓN

Establecer en cada caso el peso en porcentaje que tiene en la evaluación de la asignatura la parte de la evaluación continua, y la correspondiente a la prueba del examen final.

Descripción de la evaluación continua: actividades que debe desarrollar el alumno y su valoración

Para superar la asignatura se deberán superar por separado dos partes: teoría y prácticas de laboratorio. La nota final será la media ponderada del 60% de teoría y el 40% de laboratorio.

- La nota de teoría en evaluación continua vendrá determinada por:
- 65%: nota de las pruebas parciales (temas 1-4 y 5-9), que consistirán en preguntas cortas (30%) y en ejercicios (70%). Se dejan apuntes en los ejercicios. Se deberá obtener al menos una media ponderada de 5'0 en esta parte.
 - 15%: trabajo en grupo (tema 10) y su defensa en clase.
 - 20%: entrega de ejercicios voluntarios y participación en debates y análisis de artículos.

- La nota de prácticas en evaluación continua vendrá determinada por:
- 80%: memorias de las prácticas.
 - 20%: participación en las clases de laboratorio.

Los trabajos en grupo (teoría y laboratorio) tendrán un reparto interno de puntos decidido por el propio grupo de alumnos.

Descripción del examen final.

- Los alumnos que no superen por evaluación continua la asignatura, podrán optar al examen final, bien de teoría, de prácticas o de ambos. En el examen final se dejarán apuntes.
- Para superar la asignatura se deberán superar por separado dos partes: teoría y prácticas de laboratorio. La nota final será la media ponderada del 60% de teoría y el 40% de laboratorio.
- La nota del examen final de teoría supondrá el 65% de la nota final de teoría (igual que los parciales en evaluación continua).
- La nota del examen final de prácticas supondrá el 80% de la nota final de prácticas (igual que las notas de las memorias en evaluación continua).

8. OBSERVACIONES

Indicar en este apartado si se prevé el uso de algún tipo de Software (nombre y versión), además de cualquier otra observación que se desee.

Dada la naturaleza de la asignatura, en la que se abordan diversos tipos de esfuerzos y habilidades, se emplearán diversas herramientas:

- Visual Studio 2005 y lenguaje C# para mantener una aplicación heredada.
- Visual Paradigm for UML para hacer ingeniería inversa de un software ya construido.

- Junit en plataforma Eclipse
- Subversion en plataforma ECLIPSE para gestionar la configuración.
- SDMetrics para medir propiedades de un diseño UML.
- Microsoft Project 2003 para planificar y hacer seguimiento de proyectos.
- USC COCOMO II para estimar el esfuerzo y coste del desarrollo de un software.