

# ORGANIZACIÓN DOCENTE del curso 2010-11

## 1. DATOS GENERALES DE LA ASIGNATURA

NOMBRE	Ingeniería del Software II		PÁGINA WEB	Alojada en moodle.unican.es	
CÓDIGO					
DEPARTAMENTO	Matemáticas, Estadística y Computación				
PLAN DE ESTUDIOS	Ingeniero en Informático		CURSO	2010/2011	
PROFESORADO	<b><u>Nombre</u></b>		<b><u>e-mail</u></b>		
	Pablo Sánchez		<a href="mailto:p.sanchez@unican.es">p.sanchez@unican.es</a>		
	Juan Hernández		<a href="mailto:juan.hernandez@unican.es">juan.hernandez@unican.es</a>		
	Rafael Duque		<a href="mailto:rafael.duque@unican.es">rafael.duque@unican.es</a>		
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>	
HORA RIO PREVI STO(*)	<u>Teóricos</u>	<u>Prac. Problemas</u>	<u>Prac. Laboratorio</u>	<u>Prac. Computador</u>	
Observaciones:					

(\*) Lo rellenará la secretaría del centro

(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:

### Parte A – Continuación de Ingeniería del Software

#### 1. Pruebas de Sistemas 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. Modelado de Pruebas: UML Testing Profile. Metodologías de Pruebas: TTCN-3.

#### 2. Calidad de Sistemas y Procesos Software

Concepto y Factores de Calidad del Software – Calidad de Producto: Norma ISO 9126 y Calidad de la Información. Calidad del Proceso de Desarrollo del Software: ISO 90003, CMMI e ISO 15504. Métricas de los Productos y Procesos Software. Proceso GQM. Gestión de la Calidad.: Aseguramiento, Verificación y Validación del Software. Revisiones, Inspecciones y Auditorías.

### Parte B – Gestión de Proyectos Software

#### 3. Fundamentos de Gestión de Proyectos

Concepto de Proyecto, Ciclo de Vida y Procesos de Gestión de Proyectos – PMBOK. Conceptos básicos y Áreas de Trabajo e Interacciones. Métricas de Control de Proyectos: Earned Value Management (EVM).

#### 4. 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. El estándar SPEM.

#### 5. Gestión de Costes

Motivación. Retos de la Estimación de Costes Software. Técnicas Generales de Estimación de Costes. Elaboración de Presupuestos. Estimación del Tamaño mediante Puntos Función. Método COCOMO II para la Estimación del Esfuerzo y el Coste. Control de Costes.

#### 6. Gestión del Tiempo

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

#### 7. Gestión de Riesgos

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

### Parte C – Mantenimiento y Configuración de Sistemas Software.

#### 8. Mantenimiento de Sistemas Software

Introducción. Definiciones. Tipos de Mantenimiento. Principales Retos del Mantenimiento Software. Sistemas Heredados. Técnicas de Gestión del Mantenimiento. Norma ISO 14764. Técnicas de Mantenimiento: Ingeniería Inversa, Refactorización y Métricas. Desarrollo Software Orientado a Aspectos como Técnica de Mejora del Mantenimiento.

#### 9. Gestión de la Configuración de Sistemas Software

Introducción. Definiciones. Trazabilidad. Estándar IEEE 828-1998. Identificación de Versiones. Árboles de Características. Control de Versiones. Gestión de Entregas.

### Parte D – Miscelánea

#### 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. Aspectos éticos. Prevención de Riesgos Laborales. Subcontratación.

**PRÁCTICAS:**

- P1. Diseño y Ejecución de Pruebas de Sistemas Software.
- P2. Medición de propiedades del software con SDMetrics.
- P3. Medición del Proceso de desarrollo software mediante estándar EVM.
- P4. Planificación y seguimiento de un proyecto con Microsoft Project.
- P5. Estimación de Costes en Sistemas Software con COCOMO II.
- P6. Introducción a la Programación Orientada a Aspectos.
- P7. Gestión de Peticiones de Mantenimiento con Bugzilla.
- P8. Control de Versiones con Subversion.
- P9. Creación de Árboles de Características y Ficheros ANT para el Control de Versiones.

**Asignaturas que se recomienda al alumno haber cursado o estar cursando**

**Pre-requisitos:**

Ingeniería del software I, Programación

**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**

- (1) Continuar la asignatura de Ingeniería del Software I, explicando elementos de la Ingeniería de Sistemas Software que por razones de tiempo no pudieron abordarse en Ingeniería del Software I, tales como Pruebas de Sistemas Software o Control de la Calidad.
- (2) Adquirir conocimientos y destrezas básicas sobre Gestión de Proyectos Software, tales como Gestión de Tiempo o Costes.
- (3) 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:**

- (1) Métodos de pruebas de sistemas software.
- (2) El concepto y la importancia de la calidad del software, en sus dos dimensiones, de producto y de proceso.
- (3) Los diferentes elementos que forma parte de la gestión de un proyecto.
- (4) Métodos de planificación de sistemas software.
- (5) Métodos de control temporal de la planificación de un proyecto.
- (6) Métodos de estimación de esfuerzo y coste.
- (7) Métodos para la gestión de riesgos en proyectos software.
- (8) Importancia el mantenimiento del desarrollo del software.
- (9) Métodos para la gestión y la mejora del mantenimiento software.
- (10) Métodos para la gestión de la configuración de sistemas software.

**SABER:**

- (1) Diseñar y ejecutar pruebas de sistemas software.
- (2) Desarrollar y ejecutar planes de calidad, tanto de productos como de procesos software.
- (3) Diseñar un plan de proyecto software.
- (4) Utilizar una herramienta de gestión de proyectos.
- (5) Diseñar un calendario detallado de proyecto e identificar y calcular aspectos relevantes de dicho calendario.
- (6) Calcular estimaciones de esfuerzo y coste para proyectos software.
- (7) Diseñar un plan de riesgos.
- (8) Aplicar técnicas de ingeniería inversa y refactorización para mantenimiento de software.
- (9) Aplicar métricas para evaluar la mantenibilidad y estabilidad de diferentes diseños software.
- (10) Diseñar árboles de características para el control de la configuración.
- (11) Controlar versiones de sistemas software usando alguna herramienta de gestión de configuración del software.

## 5. BIBLIOGRAFÍA

### Básica

- (1) Sommerville, 2005. "Ingeniería del Software". 7ª Edición, Addison-Wesley. Julio 2005.

#### ESTÁNDARES

- (1) Project Management Institute, 2004. "A Guide to the Project Management Body of Knowledge (PMBOK)", 3rd edition.  
 (2) Pierre Bourque and Robert Dupuis (Editores). "Guide to the Software Engineering Body of Knowledge". IEEE (The Institute of Electrical and Electronics Engineers), 2004.

### Complementaria

- (1) Colin Willcock, Thomas Deiß, Stephan Tobies, Stefan Keil, Federico Engler, Stephan Schulz. "An Introduction to TTCN-3". Wiley 2005.  
 (2) Paul Baker, Zhen Ru Dai, Jens Grabowski, Øystein Haugen, Ina Schieferdecker, Clay Williams. "Model-Driven Testing: Using the UML Testing Profile". Springer. 2007.  
 (3) David Garmus, David Herron. "Function Point Analysis: Measurement Practices for Successful Software Projects". Addison-Wesley, 2002.  
 (4) Caper Jones. "Estimating Software Costs: Bringing Realism to Estimating". McGraw-Hill, 2007.  
 (5) Barry W. Boehm, Chris Abts, A. Winsor Brown, Sunita Chulani, Bradford K. Clark, Ellis Horowitz, Ray Madachy, Donald J. Reifer, Bert Steece. "Software Cost Estimation with COCOMO II". 2009.  
 (6) Penny Grubb, Armstrong A. Takang. "Software Maintenance: Concepts and Practice". World Scientific Publishing, 2003.  
 (7) Frederick P. Brooks. "The Mythical Man-Month: Essays on Software Engineering" Anniversary Edition. 1995.  
 (8) Martin Fowler. "Refactoring: Improving the Design of Existing Code". Addison-Wesley 1999.  
 (9) C. Michael Pilato, Ben Collins-Sussman, Brian W. Fitzpatrick. "Version Control with Subversion". O'Reilly 2004.  
 (10) Alexis Leon. "Software configuration management handbook". Artech Print.  
 (11) Anne Mette Jonassen Hass. "Configuration Management Principles and Practice." Addison-Wesley

#### ESTÁNDARES

- (3) ANSI/IEEE, 1998. Std. 1058: IEEE Standard for Software Project Management Plans. IEEE Computer Society, USA.  
 (4) ISO/IEC, 2001: ISO/IEC 9126-1:2001 Software Engineering - Product Quality -- Part 1: Quality Model.  
 (5) ISO/IEC/IEEE. Software engineering – software life cycle processes – maintenance. Technical Report International Standard ISO/IEC 14764:2006, IEEE Std 14764-2006, September 2006.  
 (6) IEEE (The Institute of Electrical and Electronics Engineers). Standard for Software Maintenance. Technical Report IEEE Std 1219-1993, December 1992.  
 (7) IEEE (The Institute of Electrical and Electronics Engineers). Standard for Software Configuration Management Plans. Technical Report IEEE Std 828-2005, August 2005.

## 6. ACTIVIDADES A DESARROLLAR EN LA ASIGNATURA

Los tipos de actividades a desarrollar serán los siguientes:

- (1) Clases magistrales para los temas de teoría.
- (2) Clases de ejercicios y problemas.
- (3) Trabajos en grupo de 3/4 alumnos para el tema de teoría 10 y defensa en la clase. Los trabajos se propondrán por parte del profesorado a comienzo de curso. Los alumnos formarán grupos y elegirán trabajo en la primera semana de clase. A mitad de cuatrimestre habrá una sesión de control, donde los alumnos informarán del estado del trabajo al profesorado, de forma que éste pueda evaluar el estado del mismo y proponer las medidas correctivas oportunas. Los trabajos se expondrán en la última semana de curso y se deberán entregar en la semana previa.
- (4) Prácticas guiadas de laboratorio con ordenador. Son aquellas en las cuales el profesor, con apoyo de computadores, realizará una explicación de ciertos aspectos prácticos de la asignatura. Al final de estas prácticas, el alumno no tiene que entregar ningún material evaluable, pero lo explicado durante la sesión de prácticas puede ser preguntado en las pruebas evaluables sobre contenidos teóricos, y/o en la prueba evaluable final sobre contenidos prácticos.
- (5) Prácticas no guiadas. Son aquellas en las que el profesor propone un caso práctico al alumno que debe resolver mediante las herramientas disponibles en los laboratorios. El alumno deberá entregar los resultados obtenidos para su posterior evaluación por parte del profesorado.
- (6) Pruebas de evaluación parciales.
- (7) Tutorías.
- (8) Seminarios.
- (9) Visitas a empresas del sector donde trabajadores de dichas empresas expliquen a los alumnos en qué aspectos de su trabajo diario se reflejan los contenidos teóricos explicados en clase y como lo llevan a la práctica. Dichas visitas son sustituibles por seminarios impartidos dentro del aula por trabajadores de dichas empresas.

## 7. MÉTODO DE EVALUACIÓN

### EVALUACIÓN CONTINUA

Para superar la asignatura se deberán superar por separado dos partes: teoría y prácticas. La nota final será la media ponderada del 70% de teoría y el 30% de prácticas.

La nota de teoría en evaluación continua vendrá determinada por:

85%: Nota de las pruebas parciales que consistirán en preguntas cortas y en ejercicios. La fecha de los exámenes parciales se fijarán en la primera semana de clases y a partir de ahí serán inamovibles. Los exámenes parciales se realizarán siempre en horarios de clase y en ningún caso durarán menos de dos horas. El reparto del peso entre preguntas cortas y ejercicios se ponderará de acuerdo con la carga teórica y práctica de cada bloque. Para la parte de ejercicios se dejará hacer uso de todo tipo de material escrito, incluyendo apuntes y/o ejercicios, así como calculadoras básicas. En ningún caso se dejará hacer uso de portátiles, agendas personales, teléfonos móviles inteligentes u otros dispositivos electrónicos parecidos. Especialmente, queda totalmente prohibido el uso de dispositivos con capacidades de comunicación inalámbricas. Se considerarán ejercicios supuestos prácticos que describas una situación real de la vida de un Ingeniero Informático, y donde, con los datos aportados, el alumno tenga que tomar una cierta decisión. Se valorará justificación de la respuesta, capacidad de síntesis, claridad de la redacción y precisión en el uso del lenguaje técnico. Se deberá obtener al menos una media ponderada de 4.50 en las pruebas parciales teóricas para poder superar la asignatura.

15%: Trabajo en grupo (tema 10) y su defensa en clase.

La nota de prácticas en evaluación continua vendrá determinada por la media ponderada del material entregado como resultado de la realización de las prácticas de laboratorio no guiadas.

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

### EVALUACIÓN FINAL

Los alumnos que no superen por evaluación continua la asignatura, podrán optar a una prueba evaluable final, bien de teoría, de prácticas o de ambos. La prueba evaluable final se registrará por la misma normativa que las pruebas parciales, a excepción de que se realizará en periodo de exámenes en la fecha fijada por el centro. La prueba evaluable final contará un 85% de la nota, quedando el 15% restante asignado al trabajo realizado sobre el Tema 10.

## 8. OBSERVACIONES

Para que los alumnos alcancen los objetivos planteados, se hará uso de las siguientes herramientas:

- (1) Herramienta de modelado UML: Visual Paradigm y/o IBM Rational. Esta última depende de que prospere la inclusión de al menos de uno de los profesores en el programa IBM Academic, lo que permitiría obtener IBM Rational a coste cero.
- (2) Eclipse con JUnit, Subversion, AspectJ, ANT e Hydra. (todas estas herramientas son *open-source*).
- (3) SDMetrics
- (4) Bugzilla (*open source*).
- (5) USC COCOMO II (*open source*).
- (6) Microsoft Project.