

# ORGANIZACIÓN DOCENTE del curso 2008-09

## 1. DATOS GENERALES DE LA ASIGNATURA

<b>NOMBRE</b>	Ingeniería del Software I		<b>PÁGINA WEB</b>	http://personales.unican.es/ruizfr/is1/	
<b>CÓDIGO</b>					
<b>DEPARTAMENTO</b>	Matemáticas, Estadística y Computación				
<b>PLAN DE ESTUDIOS</b>	Ingeniero en Informático		<b>CURSO</b>	2008/2009	
<b>PROFESORADO</b>	<b><u>Nombre</u></b>		<b><u>e-mail</u></b>		
	Francisco Ruiz González (teoría, coordinador)		francisco.ruiz@unican.es		
	Maria Rita Sierra Sánchez		mariarita.sierra@unican.es		
<b>CRÉDITOS ALUMNO</b>	<u>Teóricos</u> (1)	<u>Prac. Problemas</u> (2)	<u>Prac. Laboratorio</u>	<u>Prac. Computador</u>	<b>TOTALES</b>
	4.5	1.5		3.0	9.0
<b>LUGAR DE IMPARTICIÓN</b>	<u>Teóricos</u>	<u>Prac. Problemas</u>	<u>Prac. Laboratorio</u>	<u>Prac. Computador</u>	
	Aula	Aula		Aula PCs	
<b>HORARIO PREVISTO(*)</b>	<u>Teóricos</u>	<u>Prac. Problemas</u>	<u>Prac. Laboratorio</u>	<u>Prac. Computador</u>	
<b>(*) Observaciones:</b>	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 – Fundamentos

##### 1. Introducción a la Ingeniería del Software

Concepto y objetivos. Cuerpo de Conocimientos – SWEBOK. Áreas principales.

##### 2. Procesos de Ingeniería del Software.

Ciclo de Vida del Software. Procesos del ciclo de vida – ISO 12207. Ciclos de Vida tradicionales. Ciclos de vida para sistemas orientados a objetos. Metodologías de desarrollo de software: concepto, evolución histórica, tipos.

##### 3. Requisitos

Concepto y características. Tipos de requisitos. Captura. Análisis. Especificación. Validación.

##### 4. Diseño de Software.

Contexto y Aspectos Clave. Arquitectura Software. Patrones de Diseño. Notaciones: Descripciones Estructurales (estática); Descripciones de Comportamiento (dinámica). Estrategias: Estructurada; Orientada a Objetos; Centrada en los Datos; Basada en Componentes.

##### 5. Construcción y Pruebas del Software.

Principios de construcción. Codificación. Reutilización e Integración de código. Fallos, errores y defectos. Niveles y tipos de pruebas. Técnicas de prueba.

##### 6. Estudio de Metodologías.

Proceso Unificado. METRICA 3. Programación Extrema.

#### Parte B – Desarrollo Orientado a Objetos

##### 7. Lenguaje Unificado de Modelado - UML

Características de metodologías orientadas a objetos. Objetivos y características de UML. Usos de UML: Visualizar; Especificar; Construir; Documentar. Tipos de diagramas. Arquitectura del Sistema.

##### 8. Contexto y Requisitos del Sistema

Conceptos básicos de Casos de Uso. Relaciones. Diagramas de casos de uso. Modelado del contexto de un sistema. Modelado de los requisitos de un sistema.

##### 9. Interacciones en el Sistema

Elementos de la Interacción. Diagramas de Secuencia. Diagramas de Colaboración. Modelado de Flujos de Control por Ordenación Temporal. Modelado de Flujos de Control por Organización.

##### 10. Estructura del Sistema

Elementos principales: clases; relaciones. Diagramas de clases. Diagramas de objetos. Características Avanzadas del Modelado Estructural. Interfaces. Mecanismos de Extensión: Notas; Estereotipos; Valores Etiquetados; Restricciones. Técnicas de Modelado.

##### 11. Arquitectura Lógica del Sistema

Paquetes. Relaciones entre Paquetes. Estereotipos y Valores Etiquetados de los Paquetes. Diagramas de Paquetes. Modelado de Grupos de Elementos. Modelado de Vistas Arquitectónicas.

##### 12. Comportamiento del Sistema

Concepto de Evento. Tipos de Eventos. Máquinas de Estados. Diagramas de Transición de Estados. Diagramas de Actividad. Modelado de la Vida de un Objeto. Modelado de un Flujo de Trabajo. Modelado de una Operación.

##### 13. Arquitectura Física del Sistema

Concepto de Componente. Relaciones entre Componentes. Tipos de Componentes. Técnicas de Modelado de Componentes. Diagramas de Componentes. Despliegue: nodos, tipos y conexiones. Modelado de procesadores y dispositivos. Modelado de la distribución de componentes. Diagramas de Despliegue.

##### 14. Pruebas de Sistemas Orientados a Objetos

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.

### PRÁCTICAS (con una herramienta CASE):

P1. Especificación de requisitos software.

P2. Diseño de un software orientado a objetos.

P3. Pruebas.

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

Pre-requisitos:

Bases de Datos, Estructuras de Datos y Algoritmos, Programación II

Co-requisitos:

-

## **2. OBJETIVOS GENERALES DE LA ASIGNATURA**

Los objetivos generales son:

Comprender y conocer 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 desarrollar software mediante la aplicación de alguna metodología orientada a objetos.

## **3. OBJETIVOS ESPECIFICOS: APTITUDES/DESTREZAS**

Los anteriores objetivos generales se concretan en:

CONOCER:

- Las diversas áreas que conforman la disciplina de Ingeniería del Software.
- Los principales ciclos de vida y metodologías existentes.
- Los fundamentos del análisis (requisitos), diseño, construcción y pruebas del software.
- Métodos y técnicas precisos para el desarrollo de sistemas orientados a objetos.

SABER:

- Utilizar UML como lenguaje para el modelado de sistemas software orientado a objetos.
- Especificar requisitos.
- Diseñar sistemas software siguiendo alguna metodología orientada a objetos.
- Aplicar las principales técnicas de modelado orientado a objetos.
- Hacer pruebas de software orientado a objetos.

## **4. BIBLIOGRAFÍA**

### **Básica**

- 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**

- Booch et al., 2006. El Lenguaje Unificado de Modelado. Guía del Usuario, 2da edic. Pearson Educación, 2006.
- Rumbaugh et al., 2007. El Lenguaje Unificado de Modelado. Manual de Referencia, 2da edic. Pearson Educación, 2007.
- Jacobson et al., 2000. El Proceso Unificado de Desarrollo de Software. Addison-Wesley, 2000.
- Miles y Hamilton, 2006. Learning UML 2.0. O'Reilly, 2006.
- Rumbaugh et al., 1996. Modelado y diseño orientados a objetos, Madrid, Prentice Hall, 1996.
- Larman, 2003. UML y Patrones: Introducción al análisis y diseño orientado a objetos, 2ª Edición, Prentice Hall.

ESTANDARES:

- IEEE Computer Society, 2004. Guide to the Software Engineering Body of Knowledge (SWEBOK), 2004 version. Disponible en <http://www.swebok.org>.
- ISO/IEC, 2005: ISO/IEC 12207: Information Technology - Software life cycle processes. Versión en español disponible en <http://www.bvindicopi.gob.pe/normas/isoiec12207.pdf>.
- Object Management Group, 2005: Unified Modelling Language, version 2.0. Disponible en <http://www.omg.org/spec/UML/2.0/>.

## 5. 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 6 y defensa en la clase.  
 Prácticas de laboratorio individuales con ordenador.  
 Pruebas de evaluación parciales.  
 Tutorías.  
 Análisis y debate de artículos.

## 6. 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 65% de teoría y el 35% de laboratorio.

La nota de teoría en evaluación continua vendrá determinada por:  
 70%: nota de las pruebas parciales (temas 1-5), que consistirán en preguntas cortas (30%) y en ejercicios (70%). Se dejan apuntes. Se deberá obtener al menos una media ponderada de 5'0 en esta parte.  
 10%: trabajo en grupo (tema 6) y su defensa en clase.  
 15%: 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.

### **Descripción del examen final (duración, se pueden llevar apuntes o no, tiene partes diferenciadas o no, se promedian teoría y problemas o no, etc).**

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 65% de teoría y el 35% de laboratorio.  
 La nota del examen final de teoría supondrá el 70% 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).

## 7. OBSERVACIONES

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

Es una asignatura nueva, por lo que este primer año está sujeta a posibles cambios y ajustes. En su caso, serán comunicados y acordados con los estudiantes.