

## ORGANIZACIÓN DOCENTE del curso 2010-11

### 1. DATOS GENERALES DE LA ASIGNATURA

<b>NOMBRE</b>	Ingeniería del Software I		<b>PÁGINA WEB</b>		
<b>CÓDIGO</b>					
<b>DEPARTAMENTO</b>	Matemáticas, Estadística y Computación				
<b>PLAN DE ESTUDIOS</b>	Ingeniero en Informática		<b>CURSO</b>	2010/2011	
<b>PROFESORADO</b>	<b><u>Nombre</u></b>		<b><u>e-mail</u></b>		
	Carlos Blanco		<a href="mailto:Carlos.Blanco@unican.es">Carlos.Blanco@unican.es</a>		
	Patricia López		<a href="mailto:lopezpa@unican.es">lopezpa@unican.es</a>		
	Juan Hernández		<a href="mailto:Juan.Hernandez@unican.es">Juan.Hernandez@unican.es</a>		
<b>CRÉDITOS ALUMNO</b>	<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
<b>LUGAR DE IMPARTICIÓN(*)</b>	<u>Teóricos</u>	<u>Prac. Problemas</u>	<u>Prac. Laboratorio</u>	<u>Prac. Computador</u>	
<b>HORARIO PREVISTO(*)</b>	<u>Teóricos</u>	<u>Prac. Problemas</u>	<u>Prac. Laboratorio</u>	<u>Prac. Computador</u>	

(\*) 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:

#### 0. Presentación de la Asignatura

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

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

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

#### 2. 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.

#### 3. 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.

#### 4. Requisitos

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

#### 5. 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.

#### 6. 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.

#### 7. Interacciones en el Sistema

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

#### 8. 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.

#### 9. 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.

#### 10. 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.

#### 11. 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.

#### 12. Estudio de Metodologías.

Proceso Unificado. METRICA 3. Programación Extrema.

### PRÁCTICAS (con una herramienta CASE):

P0. Especificación inicial de Requisitos

P1. Trabajando con Visual Paradigm

P2. Especificación de Requisitos

P3. Modelado de Requisitos

P4. Modelo de Análisis

P5. Modelo de Diseño

## **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:**

-

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

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

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 orientado a objetos.

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

**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) y diseño 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.

### **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.

**UML y Proceso Unificado:**

- Booch et al., 2006. El Lenguaje Unificado de Modelado. Guía del Usuario, 2da edic. Pearson Educación, 2006.
- Miles y Hamilton, 2006. Learning UML 2.0. O'Reilly, 2006.

#### **Complementaria**

- Larman, 2003. UML y Patrones: Introducción al análisis y diseño orientado a objetos, 2ª Edición, Prentice Hall.
- 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.

**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/2.3/>

## 6. ACTIVIDADES A DESARROLLAR EN LA ASIGNATURA

Clases magistrales para los temas de teoría.  
 Clases de ejercicios y problemas.  
 Realización y entrega de ejercicios voluntarios.  
 Trabajos en grupos de 3/4 alumnos para el tema de teoría 12 y defensa en clase.  
 Prácticas de laboratorio individuales con ordenador.  
 Trabajos de laboratorio en grupos de 3/4 alumnos.  
 Pruebas de evaluación parciales.  
 Tutorías.

## 7. MÉTODO DE EVALUACIÓN

### **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 en evaluación continua vendrá determinada por:

25%: nota del primer examen parcial de teoría (temas 1-6).

25%: nota del segundo examen parcial de teoría (temas 7-11).

(\*) Se deberá obtener un mínimo de 3,5 en cada examen parcial y una media de 5 puntos entre los dos.

Las pruebas consistirán en:

- Una parte de preguntas cortas sin apuntes valorada hasta 3 puntos. Habrá que obtener un mínimo de 1 punto en esta parte.
- Una parte de ejercicios con apuntes valorada hasta 7 puntos.

10%: trabajo en grupo (tema 12) y su defensa en clase.

10%: entrega de ejercicios voluntarios y participación en clase.

30%: nota de prácticas en evaluación continua determinada por la evaluación de las memorias de las prácticas.

Los trabajos en grupo (tanto de teoría como de prácticas) 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.

Para superar la asignatura se deberán superar por separado las partes de teoría y prácticas de laboratorio, obteniendo al menos 5 puntos en cada uno de los exámenes.

- El examen final de teoría supondrá el 50% de la nota final (igual que las notas de las dos pruebas parciales).

- El examen final de prácticas supondrá el 30% de la nota final de la asignatura.

- El 20% de la nota restante corresponde a la evaluación continua realizada durante el curso (10% del trabajo en grupo y 10% la entrega de ejercicios voluntarios y participación en clase).

## 8. OBSERVACIONES

Para las prácticas de laboratorio se utilizará Visual Paradigm for UML 7.2 (<http://www.visual-paradigm.com/product/vpuml/>).