

DATOS DE LA ASIGNATURA*

* Asignatura en experiencia piloto de implantación del sistema de créditos ECTS

Nombre:			
Ingeniería del Software de Gestión II			
Denominación en inglés¹:			
Software Engineering II			
Código:	Año del Plan de Estudios:	Tipo:	
450004024	Publicación BOE: 27-07-2004	<input checked="" type="checkbox"/> Troncal <input type="checkbox"/> Obligatoria <input type="checkbox"/> Optativa	
Créditos:			
	Totales:	Teóricos:	Prácticos:
Créditos L.R.U.	6,75	2,25	4,50
Créditos E.C.T.S.	5,4	1,8	3,6
Departamento:			
Tecnologías de la Información Tecnologías de la Información			
Área de Conocimiento:			
Lenguaje y Sistemas Informáticos Lenguaje y Sistemas Informáticos			
Curso:	Cuatrimestre:	Ciclo:	
Tercero	1º Cuatrimestre	Primero	
Web de la asignatura:			
http://moodle.uhu.es			

¹Para su inclusión en el Complemento Europeo al Título

DATOS DE LOS PROFESORES

Nombre:	e-mail:	Teléfono:	Despacho:
Manuel de la Villa Cordero	manuel.villa@dti.uhu.es	959 217639	44

DATOS ESPECÍFICOS DE LA ASIGNATURA

1.1. Descriptores de la asignatura:
Diseño, propiedades y mantenimiento del software. Planificación y gestión de proyectos informáticos. Análisis de aplicaciones de gestión
1.2. Descriptores de la asignatura (en inglés)²:
Design, properties and maintenance of the software. Planning and managing software projects. Analysis of management applications

²Para su inclusión en el Complemento Europeo al Título

2. Situación de la asignatura.
2.1. Prerrequisitos:
<ul style="list-style-type: none">• Ninguno en los actuales Planes de Estudio• Es recomendable el conocimiento de materias fundamentales en la titulación, como Metodología de la Programación, pues se basa en el paradigma de Orientación a Objetos y para los ejemplos se usará el lenguaje Java.
2.2. Contexto dentro de la titulación:
Por sus contenidos y de acuerdo con los descriptores del BOE, esta materia está interrelacionada con la práctica totalidad de las que componen el título y se basa en las habilidades adquiridas para llevar a la práctica los paradigmas de desarrollo y el análisis y diseño de sistemas software. Es una continuación de la asignatura 'Ingeniería del Software de Gestión I', centrándose este segundo curso en una visión mucho más práctica e instrumental de la materia, mediante la inmersión en un proyecto de desarrollo software.
2.3. Recomendaciones:
Si bien no es obligatorio, se recomienda encarecidamente que los alumnos tengan aprobadas las siguientes asignaturas: <ul style="list-style-type: none">• Ingeniería del Software de Gestión I (imprescindible)• Metodología de la Programación I.• Metodología de la Programación II.• Estructuras de Datos I.• Estructuras de Datos II.• Bases de Datos I.

3. Competencias a adquirir por los estudiantes.

3.1. Competencias transversales o genéricas.

3.1.1. Competencias instrumentales:

<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad de análisis y síntesis.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad de organización y planificación.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Comunicación oral y escrita en lengua nativa.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Conocimiento de una lengua extranjera.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad de gestión de la información.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Resolución de problemas.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Toma de decisiones.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.

3.1.2. Competencias personales:

<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Trabajo en equipo.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Trabajo en un equipo de carácter interdisciplinar.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Trabajo en un contexto internacional.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Habilidades en las relaciones interpersonales.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad para comunicarse con expertos de otras áreas.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Reconocimiento a la diversidad y la multiculturalidad.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Razonamiento crítico.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Compromiso ético.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.

3.1.3. Competencias sistémicas:

<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Aprendizaje autónomo.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Adaptación a nuevas situaciones.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad de aplicar los conocimientos en la práctica.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Habilidad para trabajar de forma autónoma.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Creatividad.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Liderazgo.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Conocimiento de otras culturas y costumbres.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Iniciativa y espíritu emprendedor.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Motivación por la calidad.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Sensibilidad hacia temas medioambientales.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.

3.2. Competencias específicas.

3.2.1. Competencias cognitivas (saber):

- Elección de paradigmas de desarrollo.
- Técnicas de especificación y análisis de requisitos.
- Técnicas de análisis y modelado orientado a objetos.
- Técnicas de implementación basadas en el uso de patrones.
- Concepto de persistencia de objetos

3.2.2. Competencias procedimentales e instrumentales (saber hacer):

- Planificar y especificar las necesidades de un sistema software.
- Extracción y especificación de requisitos de un sistema software.
- Modelado de un sistema software.
- Paso del modelo a la implementación.
- Manejo de mecanismos de persistencia de objetos

3.2.2. Competencias actitudinales (ser):

- Extracción de información, especificación y análisis razonado y crítico.
- Análisis y diseño de un sistema, solucionando problemas y trabajando en equipo

4. Objetivos:	
Podemos considerar los siguientes objetivos de carácter general:	
1. Conocer las propiedades del software y su importancia en la sociedad.	
2. Conocer los aspectos relacionados con el proceso de producción del software.	
3. Tomar conciencia de la necesidad de considerar la producción del software como un proceso de ingeniería.	
4. Conocer los distintos aspectos del proceso de producción del software.	
5. Ser capaz de construir un sistema software aplicando un enfoque de ingeniería.	

5. Metodología (en horas de trabajo del estudiante):			
	Primer Cuatrimestre	Segundo Cuatrimestre	
	Presenciales		
Clases de teoría	10,0	0,0	
Clases de problemas	3,0	0,0	
Clases prácticas	36,0	0,0	
Actividades académicas dirigidas	7,5	0,0	
Exámenes	6,0	0,0	
	No presenciales		
Estudio de clases teóricas (factor de trabajo: 1,10)	11,0	0,0	
Estudio de clases de problemas y prácticas (factor de trabajo: 1,40)	54,6	0,0	
Preparación de actividades académicamente dirigidas y otras actividades	16,0	0,0	
Total:	144,1	0,0	
Trabajo total del estudiante: 144,1 horas.			
Horas presenciales:	56,5	Horas no presenciales:	81,6
		Exámenes:	6,0

6. Técnicas docentes.	
6.1. Técnicas docentes utilizadas:	
<input checked="" type="checkbox"/> Sesiones académicas de teoría <input type="checkbox"/> Sesiones académicas de problemas <input checked="" type="checkbox"/> Sesiones prácticas en laboratorio <input checked="" type="checkbox"/> Seminarios, exposiciones y debates <input checked="" type="checkbox"/> Trabajo en grupos reducidos <input checked="" type="checkbox"/> Resolución y entrega de problemas/prácticas <input type="checkbox"/> Realización de pruebas parciales evaluables <input checked="" type="checkbox"/> Otras: Participación en foros, uso de blogs y wikis. <input type="checkbox"/> Otras: Especificar	
6.2. Desarrollo y justificación:	
<ul style="list-style-type: none"> • Sesiones académicas teóricas La asignatura se desarrolla a través de sesiones teóricas mediante el uso de lecciones magistrales para las horas de teoría, impartidas con el apoyo de presentaciones con cañón retroproyector, publicando previamente las diapositivas. • Sesiones académicas prácticas <ul style="list-style-type: none"> ○ Se utilizarán 4 horas de las dedicadas a las prácticas para la exposición del problema que deberán desarrollar los alumnos en las secciones prácticas, durante esta sección se expondrá el dominio de aplicación del problema a resolver desde el punto de vista del usuario y se le expondrán también las habilidades que deberá de poseer dicha aplicación para cubrir los requisitos de usuario. ○ Así mismo, se dedicarán otras 4 horas para conocer el funcionamiento básico de las herramientas a usar (herram. CASE y de Programación). 	

- El resto de las horas de prácticas, se utilizarán para que los alumnos, en grupos planifiquen y realicen todas las fases necesarias para obtener el sistema requerido. Durante estas secciones el profesor actuará en todo momento como fuente de información para aclarar todas las posibles dudas o problemas que se les puedan plantear.

- Exposiciones de trabajos y debates

A lo largo del cuatrimestre se presentarán una lista de propuestas de trabajos (también los alumnos pueden proponer trabajos), que los alumnos deberán elaborar y presentar de manera oral delante de sus compañeros. También se dedicará alguna sesión al debate sobre temas controvertidos (software libre vs. propietario, desarrollo ágil vs. modelos de procesos, etc.)

- Sesiones conjuntas para resolución de problemas individualmente por parte de los alumnos

Al final de cada bloque temático o en ciertos temas se dedicarán sesiones a la resolución de problemas prácticos (habitualmente incluidos en exámenes de cursos anteriores). Se intenta primar la participación del alumno, actuando el profesor como coordinador del esfuerzo y portavoz del grupo.

7. Bloques temáticos:

Bloque I: Planificación de proyectos software.

Bloque II: Análisis de sistemas de información.

Bloque III: Construcción de sistemas de información.

8. Temario desarrollado:

Bloque I: Planificación de proyectos software.

Tema 1 UML y el Proceso Unificado

1.1 Proceso de desarrollo

1.2 UML 2.0

1.3 Proceso Unificado vs Metodologías Ágiles

1.4 Agile UP

Tema 2 Agile UP. Tareas y Fases

2.1 Fases

2.2 Tareas o disciplinas

Bloque II: Análisis de sistemas de información.

Tema 3 Agile UP a fondo (I). Inicio

3.1 Fase de inicio.

3.2 Requisitos y arquitectura

3.3 Interfaces de usuario

3.4 Desarrollo de prototipos

3.5 Planificación de las iteraciones

3.6 Gestión de la configuración

Tema 4 Agile UP a fondo (II). Elaboración

Bloque III: Construcción de sistemas de información

Tema 5 Agile UP a fondo (III). Construcción

5.0 Del análisis y diseño al código

5.1 Ingeniería inversa y directa

5.2 Realización de los casos de uso

5.3 Implementación en Java de los Diagramas de Clase

5.4 Implementación en Java de los Diagramas de Secuencia

5.5 Reusabilidad en el diseño: Los patrones de diseño

5.6 Separación Modelo-Vista: El patrón MVC (Observer/Observable)

Tema 6 Agile UP a fondo (IV). Transición.

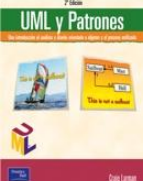
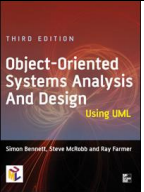

6.1 Introducción a la persistencia

6.2 Pruebas



6.3 Despliegue

9. Bibliografía.

9.1. Bibliografía general:

	<p>UML y patrones Autor: Craig Larman Pearson Prentice Hall</p> <p>Uno de los mejores libros para la asignatura, eminentemente práctico, recorre todo un ciclo de vida RUP, iterativamente.</p>
	<p>Object Oriented Systems Analysis and Design using UML, 3/e Autores: Simon Bennett, Steve McRobb, Ray Farmer McGraw-Hill</p> <p>Muy buen libro, ya existe versión en castellano, recorre todo el ciclo de vida con especial hincapié en los aspectos de implementación</p>
	<p>Análisis y diseño orientado a objetos con UML y el proceso unificado Autor: Schach, Stephen McGraw-Hill</p> <p>Muy buen libro, con muchos ejemplos y ejercicios, abarca todo el ciclo de vida O.O.</p>

9.2. Bibliografía específica:

	<p>UML para Programadores en Java Autor: Robert Cecil Martín Pearson Addison Wesley</p> <p>Libro que conecta Java y UML, muy centrado en la implementación de todas las estructuras UML</p>
	<p>Ingeniería de Software orientada a objetos con UML, JAVA e Internet Autor: Alfredo Weitzenfeld Thomson</p> <p>Libro eminentemente práctico, realiza un caso de estudio completo</p>

10. Técnicas de evaluación.

10.1. Técnicas de evaluación utilizadas:

<input checked="" type="checkbox"/> Examen teórico-práctico
<input checked="" type="checkbox"/> Trabajos desarrollados durante el curso
<input type="checkbox"/> Participación activa en las sesiones académicas
<input type="checkbox"/> Controles periódicos de adquisición de conocimientos
<input type="checkbox"/> Examen práctico en aula de informática
<input checked="" type="checkbox"/> Otras: Seguimiento y documentación de un proyecto
<input checked="" type="checkbox"/> Otras: Trabajo expositivo sobre Patrones de Diseño

10.2. Criterios de evaluación y calificación:

- Examen teórico y práctico.

Al final del cuatrimestre, el alumno habrá de someterse a una prueba teórica escrita en la que se evalúe su aprovechamiento de la materia, mediante cuestiones teóricas y problemas de análisis y diseño a resolver, incluso supuestos prácticos.

Con todos los resultados obtenidos por el alumno en cada una de estas técnicas, con una ponderación que dependerá del grado y compromiso de participación, se calcula el resultado tanto de teoría, como de prácticas y el final de la asignatura.

- Realización de trabajo en grupo durante las prácticas.

A lo largo del cuatrimestre, durante las sesiones prácticas, el alumno habrá de realizar trabajos individuales/colectivos que podrán ser evaluados (bien APTO/NO APTO, bien con valoración numérica). Habrá de realizar una práctica en grupo que se entregará (y defenderá en su caso) al final del periodo docente.

- Elaboración y defensa de trabajo

Los alumnos obtendrán una nota por la realización de un trabajo propuesto por el profesor. Los trabajos consistirán en el resumen y análisis de un tema, así como su posterior defensa pública. Este curso serán los Patrones de Diseño, cada grupo estudiará, presentará y elaborará un ejemplo de uso de un patrón de diseño. Se valorará especialmente la claridad y estructuración del trabajo, así como la opinión personal del alumno.

11. Organización docente semanal (en horas presenciales del alumno)

11.1. Primer cuatrimestre:

Semana	Horas de clases de teoría	Horas de clases de problemas	Horas de clases prácticas	Actividades Académicas Dirigidas		Horas de exámenes	Temas del temario a tratar
				Actividad	Horas		
1ª	1,0	0,0	0,0		0,0	0,0	Presentacion
2ª	1,5	0,0	3,0		0,0	0,0	1
3ª	0,0	0,0	3,0		1,5	0,0	2
4ª	1,5	0,0	3,0		0,0	0,0	3
5ª	0,0	0,0	3,0	resolucion de problemas	1,5	0,0	
6ª	1,5	0,0	3,0		0,0	0,0	4
7ª	0,0	1,5	3,0		0,0	0,0	5
8ª	1,5	0,0	3,0		0,0	0,0	5
9ª	0,0	0,0	3,0	exposicion de trabajos	1,5	0,0	
10ª	0,0	0,0	3,0	exposicion de trabajos	1,5	0,0	
11ª	0,0	0,0	3,0	conferencia	1,5	0,0	
12ª	1,5	0,0	3,0		0,0	0,0	6
13ª	0,0	1,5	3,0		0,0	0,0	
14ª	1,5	0,0	0,0		0,0	0,0	7-8
15ª	0,0	0,0	0,0		0,0	0,0	
Periodo de exámenes						6,0	
Totales	10,0	3,0	36,0		7,5	6,0	

11.2. Segundo cuatrimestre:

Semana	Horas de clases de teoría	Horas de clases de problemas	Horas de clases prácticas	Actividades Académicas Dirigidas		Horas de exámenes	Temas del temario a tratar
				Actividad	Horas		
1ª	0,0	0,0	0,0		0,0	0,0	
2ª	0,0	0,0	0,0		0,0	0,0	
3ª	0,0	0,0	0,0		0,0	0,0	
4ª	0,0	0,0	0,0		0,0	0,0	
5ª	0,0	0,0	0,0		0,0	0,0	
6ª	0,0	0,0	0,0		0,0	0,0	
7ª	0,0	0,0	0,0		0,0	0,0	
8ª	0,0	0,0	0,0		0,0	0,0	
9ª	0,0	0,0	0,0		0,0	0,0	
10ª	0,0	0,0	0,0		0,0	0,0	
11ª	0,0	0,0	0,0		0,0	0,0	
12ª	0,0	0,0	0,0		0,0	0,0	
13ª	0,0	0,0	0,0		0,0	0,0	
14ª	0,0	0,0	0,0		0,0	0,0	
15ª	0,0	0,0	0,0		0,0	0,0	

Período de exámenes						0,0	
Totales	0,0	0,0	0,0		0,0	0,0	

12. Mecanismos de control y seguimiento:

- En cuanto al método y técnicas docentes empleadas, se valora mucho la percepción que el alumno tiene de ellas, por lo que se establecen mecanismos de retroalimentación.
- Al inicio del curso se hace una encuesta rápida, en la que se pretende captar la formación y situación académica previa, algunos datos personales y, sobre todo, la actitud e ideas preconcebidas que tiene el alumno con respecto a la asignatura.
- Al final del curso se realiza otra encuesta (normalmente el día del examen, para garantizar al máximo la participación) donde se hace hincapié en la valoración personal del alumno de cada una de las técnicas docentes empleadas, aquellos aspectos a mejorar, críticas y sugerencias.
- Anualmente, al final de cada curso, se realizan reuniones docentes entre los compañeros que imparten las distintas asignaturas de 'ingeniería del software' para comentar aspectos mejorables del curso y de los distintos temarios.
- Todo ello se complementa con las evaluaciones a alumnos oficiales que realiza la propia universidad.