

**Guía docente de la asignatura**

Asignatura	FUNDAMENTOS DE INGENIERÍA DE SOFTWARE		
Materia	ENTORNO SOFTWARE		
Módulo			
Titulación	Grado en INGENIERÍA INFORMÁTICA Grado en INGENIERÍA INFORMÁTICA DE SISTEMAS		
Plan	463 y 464	Código	45188 y 45248
Periodo de impartición	S2	Tipo/Carácter	OB
Nivel/Ciclo	Grado	Curso	1
Créditos ECTS	6		
Lengua en que se imparte	Español		
Profesor/es responsable/s	Félix Prieto (Coordinador de la parte teórica), Yania Crespo (Coordinador de la parte práctica), Esperanza Manso		
Datos de contacto (E-mail, teléfono...)	felix[at]infor.uva.es, yania[at]infor.uva.es, manso[at]infor.uva.es		
Horario de tutorías	Véase www.uva.es → Centros → Campus de Valladolid → Escuela Técnica Superior de Ingeniería Informática → Tutorías		
Departamento	Informática		

1. Situación / Sentido de la Asignatura**1.1 Contextualización**

Esta asignatura, al estar integrada en el primer curso de ingeniería informática, pretende hacer énfasis en el enfoque de ingeniería de software. La asignatura presentará la ingeniería de software como un medio para resolver problemas de los clientes, garantizando la calidad del producto entregado. El reto será motivar a los estudiantes con el aprendizaje de técnicas básicas de ingeniería de software antes de saber mucho acerca de programación.

1.2 Relación con otras materias

La asignatura está planteada como una visión panorámica de las diferentes disciplinas que componen la ingeniería de software y que se desarrollan en asignaturas de cursos posteriores. En lugar de dar una visión superficial de muchas técnicas se propone una visión parcial pero no trivial de los aspectos fundamentales de las disciplinas de análisis, diseño, implementación y pruebas.

1.3 Prerrequisitos

Aunque no se han establecido prerrequisitos, es recomendable que el alumno posea conocimientos básicos de programación. En particular haber cursado la asignatura de Fundamentos de Programación. Se requiere una



buena disposición para el trabajo en equipo. Es recomendable disponer de un nivel de inglés que permita al estudiante leer bibliografía de consulta.

2. Competencias

La asignatura recogerá los aspectos principios generales de la ingeniería de software: La resolución de problemas, la abstracción, la división del sistema en componentes, la reutilización, conceptos de diseño, pruebas, principios de evolución del software Se abarcará todo el ciclo de vida de un sistema software sencillo: Requisitos, diseño, implementación con interfaces de usuario gráficas, revisión y verificación de software. Se contemplarán cambios en los requisitos, informes y seguimiento de problemas, etc. Se utilizará UML como lenguaje de modelado, utilizando modelos simples: Diagramas de casos de uso, de clases, de paquetes y de comportamiento.

2.1 Generales

Código	Descripción
G02	Conocimientos básicos de la profesión
G03	Capacidad de análisis y síntesis
G04	Capacidad de organizar y planificar
G05	Comunicación oral y escrita en la lengua propia
G08	Habilidades de gestión de la información
G09	Resolución de problemas
G10	Toma de decisiones
G11	Capacidad crítica y autocrítica
G12	Trabajo en equipo
G14	Responsabilidad y compromiso ético
G15	Liderazgo
G16	Capacidad de aplicar los conocimientos en la práctica
G18	Capacidad de aprender
G19	Capacidad de adaptarse a nuevas situaciones
G20	Capacidad de generar nuevas ideas
G21	Habilidad para trabajar de forma autónoma

2.2 Específicas

Código	Descripción
CI1	Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente
CI8	Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
CI16	Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.



3. Objetivos

Código	Descripción
CI1.1	Diferenciar los requisitos funcionales y no funcionales de un sistema software. Escribir documentos simples de requisitos y desarrollar, concisa y claramente, los requisitos de sistema nuevo o existente, basado en las necesidades de los usuarios y otros interesados.
CI1.2	Dibujar diagramas simples de diseños de software. Identificar y describir los conceptos de un dominio y las relaciones entre ellos.
CI1.3	Comprender las diferencias entre los distintos modelos dinámicos de UML. Dibujar diagramas de secuencia sencillos asociados a un caso de uso.
CI8.1	Conocer las técnicas y herramientas básicas de diseño y construcción del software
CI8.2	Entender los principios de calidad y conocer las técnicas específicas de validación y verificación.
CI8.3	Realizar inspecciones de los programas de tamaño medio, probar y depurar sistemáticamente programas pequeños
CI16.1	Describir las características diferenciales del software y las diferencias entre los distintos tipos de software. Comprender la importancia de los estándares.
CI16.2	Describir el ciclo de vida del software. Entender el proceso de ingeniería aplicado a la construcción del software, incluyendo sus etapas y objetivos, así como los procedimientos para determinar el nivel de consecución de los mismos. Aplicar estos fundamentos a ejemplos de caso de uso sencillos.

4. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	28	Estudio y trabajo autónomo individual	60
Clases prácticas de aula (A)		Estudio y trabajo autónomo grupal	30
Laboratorios (L)	30		
Prácticas externas, clínicas o de campo			
Seminarios (S)			
Tutorías grupales (TG)			
Evaluación	2		
Total presencial	60	Total no presencial	90

5. Bloques temáticos

Bloque 1: Especificación del sistema

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación



Después de una introducción general se estudian los aspectos fundamentales de requisitos y modelado estático y dinámico de sistemas software.

b. Objetivos de aprendizaje

Código	Descripción
CI1.1	Diferenciar los requisitos funcionales y no funcionales de un sistema software. Escribir documentos simples de requisitos y desarrollar, concisa y claramente, los requisitos de sistema nuevo o existente, basado en las necesidades de los usuarios y otros interesados.
CI1.2	Dibujar diagramas simples de diseños de software. Identificar y describir los conceptos de un dominio y las relaciones entre ellos.
CI1.3	Comprender las diferencias entre los distintos modelos dinámicos de UML. Dibujar diagramas de secuencia sencillos asociados a un caso de uso.
CI16.1	Describir las características diferenciales del software y las diferencias entre los distintos tipos de software. Comprender la importancia de los estándares.

c. Contenidos

1. INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE
 - Software
 - Conceptos básicos de la Ingeniería del Software
 - Ciclo de vida software
 - Estándares en Ingeniería del Software: La notación UML
2. REQUISITOS
 - Requisitos funcionales y no funcionales.
 - Técnicas de comunicación.
 - Casos de uso.
 - El documento de requisitos del sistema.
 - Herramientas de modelado
3. MODELADO DEL DOMINIO
 - El modelo Objeto
 - Clases y objetos en UML
 - Análisis: Identificación de las clases
 - Diagramas de clases
 - Modularidad: Paquetes y subsistemas
4. MODELADO DE LA INTERACCIÓN
 - Diagramas de interacción entre objetos
 - Realización de los casos de uso
 - Diagramas de estados
 - Estados, eventos y transiciones.

d. Métodos docentes

Actividad	Metodología
Clase de teoría	<ul style="list-style-type: none"> • Clase magistral participativa • Estudio de casos en aula • Resolución de problemas (identificación de requisitos, construcción de modelos, etc.) en pequeños grupos y discusión)
Clase práctica	<ul style="list-style-type: none"> • Se utilizará un método basado en la realización de un proyecto, siguiendo un esquema paralelo al de los casos de estudio presentados en el aula y siempre guiado por el profesor, que encargará y controlará el trabajo no presencial que se realizará en grupos (2/3 alumnos), siguiendo un enfoque colaborativo.
Seminarios Tutorías	<ul style="list-style-type: none"> • Talleres de aprendizaje del manejo de herramientas en sesiones específicas.



	<ul style="list-style-type: none"> • Presentación y discusión de trabajos en grupo • Seguimiento de las prácticas desarrolladas en grupo.
--	---

e. Plan de trabajo

Ver cronograma al final

f. Evaluación

Ver tabla y criterios en el punto 7 de esta guía.

g. Bibliografía básica

[Larman] Larman, C. "UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado". Prentice Hall, 2002. (2ª ed.)

[Pressman] Pressman, Roger S. "Ingeniería del software: un enfoque práctico MacGraw-Hill", 2010 (7ª ed.)

h. Bibliografía complementaria

[Sommerville] Sommerville, I. "Ingeniería del software" Pearson, 2005 (7ª ed.)

[Booch] Booch, G., Jacobson, I., Rumbaugh, J. "El Lenguaje Unificado de Modelado. Guía del usuario". Addison-Wesley/Diaz de Santos, 2ª edición, 2005

i. Recursos necesarios

Herramientas de ingeniería de software proporcionadas a través de aula virtual

Bloque 2: Construcción del sistema

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación

El segundo bloque presenta una visión de conjunto de las disciplinas que se utilizan para construir el software: diseño, implementación y validación.

b. Objetivos de aprendizaje

Código	Descripción
CI8.1	Conocer las técnicas y herramientas básicas de diseño y construcción del software
CI8.2	Entender los principios de calidad y conocer las técnicas específicas de validación y verificación.
CI8.3	Realizar inspecciones de los programas de tamaño medio, probar y depurar sistemáticamente programas pequeños



c. Contenidos

5. DISEÑO E IMPLEMENTACIÓN
 - Diseño del sistema
 - Tipos de Arquitecturas.
 - La interacción Hombre-Máquina.
 - Diseño de la interacción
 - Traducción de diseño a código
6. VERIFICACIÓN Y VALIDACIÓN
 - Validación: aceptación de requisitos
 - Verificación: Tipos de pruebas
 - Herramientas de pruebas

d. Métodos docentes

Actividad	Metodología
Clase de teoría	<ul style="list-style-type: none">• Clase magistral participativa• Estudio de casos en aula• Resolución de problemas (identificación de requisitos, construcción de modelos, etc.) en pequeños grupos y discusión)
Clase práctica	<ul style="list-style-type: none">• Se utilizará un método basado en la realización de un proyecto, siguiendo un esquema paralelo al de los casos de estudio presentados en el aula y siempre guiado por el profesor, que encargará y controlará el trabajo no presencial que se realizará en grupos (2/3 alumnos), siguiendo un enfoque colaborativo.
Seminarios Tutorías	<ul style="list-style-type: none">• Talleres de aprendizaje del manejo de herramientas en sesiones específicas.• Presentación y discusión de trabajos en grupo• Seguimiento de las prácticas desarrolladas en grupo.

e. Plan de trabajo

Ver cronograma al final

f. Evaluación

Ver tabla y criterios en el punto 7 de esta guía.

g. Bibliografía básica

[Larman] Larman, C. "UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado". Prentice Hall, 2002. (2ª ed.)

[Pressman] Pressman, Roger S. "Ingeniería del software: un enfoque práctico MacGraw-Hill", 2010 (7ª ed.)

h. Bibliografía complementaria

[Sommerville] Sommerville, I. "Ingeniería del software" Pearson, 2005 (7ª ed.)

[Booch] Booch, G., Jacobson, I., Rumbaugh, J. "El Lenguaje Unificado de Modelado. Guía del usuario". Addison-Wesley/Díaz de Santos, 2ª edición, 2005



i. Recursos necesarios

Herramientas de ingeniería de software proporcionadas a través de aula virtual

Bloque 3: Proceso de desarrollo

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación

El tercer bloque está dedicado al estudio del proceso de desarrollo de sistemas software, una vez que el alumno dispone de una visión general de las distintas técnicas y disciplinas de desarrollo.

b. Objetivos de aprendizaje

Código	Descripción
CI16.2	Describir el ciclo de vida del software. Entender el proceso de ingeniería aplicado a la construcción del software, incluyendo sus etapas y objetivos, así como los procedimientos para determinar el nivel de consecución de los mismos. Aplicar estos fundamentos a ejemplos de caso de uso sencillos.

c. Contenidos

- 7. EL PROCESO DE DESARROLLO
 - Métodos de desarrollo: alternativas
 - El Proceso Unificado de Desarrollo
 - Métodos ágiles
 - Herramientas de planificación y gestión

d. Métodos docentes

Actividad	Metodología
Clase de teoría	<ul style="list-style-type: none">Clase magistral participativaEstudio de casos en aulaResolución de problemas (identificación de requisitos, construcción de modelos, etc.) en pequeños grupos y discusión)

e. Plan de trabajo

Ver cronograma al final

f. Evaluación

Ver tabla y criterios en el punto 7 de esta guía.

g. Bibliografía básica



[Larman] Larman, C. "UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado". Prentice Hall, 2002. (2ª ed.)

[Pressman] Pressman, Roger S. "Ingeniería del software: un enfoque práctico MacGraw-Hill", 2010 (7ª ed.)

h. Bibliografía complementaria

[Sommerville] Sommerville, I. "Ingeniería del software" Pearson, 2005 (7ª ed.)

i. Recursos necesarios

Herramientas de ingeniería de software proporcionadas a través de aula virtual

6. Temporalización (por bloques temáticos)

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1	4,4	Semanas 1..11
Bloque 2	1,2	Semanas 12..14
Bloque 3	0.4	Semana 15

Ver detalle en el cronograma de actividades

7. Tabla resumen de los instrumentos, procedimientos y sistemas de evaluación/calificación

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Cuestionario de los primeras 2 unidades (15 minutos)	4	
Entrega práctica (requisitos)	10	
Cuestionario de la unidad 3	4	
Entrega práctica (modelo de dominio)	12	
Cuestionario de la unidad 4	4	
Entrega práctica (modelo dinámico y modelo de dominio modificado)	14	
Cuestionario de las unidades 5 y 6	4	
Cuestionario de la unidad 7	4	
Entrega práctica (pruebas)	4	
Examen sobre un supuesto práctico, resolución de ejercicios del tipo de los realizados en aula y laboratorio	40	



Criterios de calificación	
<ul style="list-style-type: none"> • Calificación final: Suma ponderada de los cuestionarios (20%), prácticas en parejas (40%) y examen (40%), debiendo obtener una suma igual o mayor a 5. Será necesaria una calificación mínima de 4/10 en el examen. <ul style="list-style-type: none"> ○ Si nota(examen) \geq 4, Nota final= Suma ponderada ○ Si nota(examen) $<$ 4, Nota final= mínimo(Suma ponderada; 4,5) • Calificación de la convocatoria extraordinaria: se utilizará la misma fórmula de cálculo de la nota final, <ul style="list-style-type: none"> ○ Obligatoriamente, se realizará un examen con el mismo formato que en la convocatoria ordinaria ○ Opcionalmente, el alumno podrá realizar un cuestionario global que sustituye la nota acumulada de los cuestionarios del curso ○ Opcionalmente, el alumno podrá volver a presentar cada una de las entregas prácticas que sustituyen a las entregadas durante el curso 	

8. Consideraciones finales

9. Cronograma de actividades previstas

Semana	Fecha	Teoría	Prácticas	Entrega Trabajos	Evaluación
1	11/02/2013	Tema 1			
2	18/02/2013	Tema 2	Introducción		
3	25/02/2013	Tema 2	Requisitos		
4	04/03/2013	Tema 3	Requisitos		Test 1..2
5	11/03/2013	Tema 3	Requisitos		
6	18/03/2013	Tema 3	Modelo Dominio	Entrega 1	
7	25/03/2013	Tema 4	Modelo Dominio		Test 3
8	08/04/2013	Tema 4	Modelo Dominio		
9	15/04/2013	Tema 4	Modelo Dominio		
10	22/04/2013	Tema 4	Modelo Dinámico		
11	29/04/2013	Tema 4	Modelo Dinámico	Entrega 2	
12	06/05/2013	Tema 5	Modelo Dinámico		Test 4
13	13/05/2013	Tema 6	Modelo Dinámico	Entrega 3	
14	20/05/2013	Tema 6	Pruebas		Test 5..6
15	27/05/2013	Tema 7	Pruebas		
16	03/06/2013		(Seminario)	Entrega 4	Test 7

Nota: Las fechas indicadas en esta tabla para pruebas y entregas son aproximadas. Tanto pruebas como entregas serán convocadas con la suficiente antelación mediante la plataforma Moodle de la escuela.