

Guía docente de la asignatura

Asignatura	FUNDAMENTOS DE INGENIERÍA DE SOFTWARE		
Materia	ENTORNO SOFTWARE		
Módulo	COMUNES A LA INFORMÁTICA		
Titulación	Grado en INGENIERÍA INFORMÁTICA		
Plan	545	Código	46919
Periodo de impartición	2º CUATRIMESTRE	Tipo/Carácter	COMPLEMENTOS DE INFORMÁTICA
Nivel/Ciclo	Grado	Curso	2º
Créditos ECTS	6		
Lengua en que se imparte	Español		
Profesor/es responsable/s	Félix Prieto (Coordinador de la parte teórica), Esperanza Manso (Coordinadora de la parte práctica)		
Datos de contacto (E-mail, teléfono...)	felix@infor.uva.es, manso@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, 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 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.

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 y Programación Orientada a Objetos. 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, , 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
CG1	Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.
CG2	Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática.
CG3	Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas, servicios y aplicaciones informáticas, así como de la información que gestionan.
CG5	Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería de software como instrumento para el aseguramiento de su calidad.
CG6	Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes.
CG10	Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática.

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
CI2	Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.
CI3	Capacidad para comprender la importancia de la negociación, los hábitos de trabajo efectivos, el liderazgo y las habilidades de comunicación en todos los entornos de desarrollo de software.
CI16	Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.

3. Objetivos

Código	Descripción
RA1	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.
RA2	Conocer los métodos, técnicas y herramientas que pueden aplicarse al desarrollo y antenimiento del software.
RA3	Entender los principios de calidad y conocer las técnicas específicas de validación y verificación
RA4	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: Fundamentos de Ingeniería del Software

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación

Tras una introducción general a la materia, se estudian los aspectos fundamentales de los requisitos de software, haciendo énfasis en su comprensión y clasificación.

Posteriormente se estudian los elementos del modelado estático y dinámico de sistemas software.

Finalmente se estudia proceso de desarrollo de sistemas software, introduciendo los elementos básicos de diseño y pruebas de sistemas software.

b. Objetivos de aprendizaje

Código	Descripción
RA1	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.
RA2	Conocer los métodos, técnicas y herramientas que pueden aplicarse al desarrollo y mantenimiento del software.
RA3	Entender los principios de calidad y conocer las técnicas específicas de validación y verificación
RA4	Aplicar estos fundamentos a ejemplos de caso de uso sencillos.

c. Contenidos

- 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
- 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.
5. EL PROCESO DE DESARROLLO
 - Métodos de desarrollo: alternativas
 - Elementos del proceso en cascada
 - Bases del diseño
 - Elementos de pruebas
 - El Proceso Unificado de Desarrollo
 - Métodos ágiles

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

El plan de trabajo será publicado en el aula virtual de la escuela una vez coordinado con el resto de las asignaturas del curso

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

6. Temporalización (por bloques temáticos)

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1	6	Semanas 1..15

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	5%	
Entrega práctica (requisitos)	12%	
Cuestionario de la unidad 3	5%	
Entrega práctica (modelo de dominio)	14%	
Cuestionario de la unidad 4	5%	
Entrega práctica (modelo dinámico y modelo de dominio modificado)	14%	
Cuestionario de la unidad 5	5%	
Examen sobre un supuesto práctico, resolución de ejercicios del tipo de los realizados en aula y laboratorio	40%	

Criterios de calificación

- **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.
 - Si $\text{nota}(\text{examen}) \geq 4$, $\text{Nota final} = \text{Suma ponderada}$
 - Si $\text{nota}(\text{examen}) < 4$, $\text{Nota final} = \text{mínimo}(\text{Suma ponderada}; 4,5)$
- **Calificación de la convocatoria extraordinaria:** se utilizará la misma fórmula de cálculo de la nota final,
 - Obligatoria, 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
 - Si no se ha entregado alguna de las prácticas de la asignatura se podrá optar a una entrega extraordinaria en las mismas condiciones de la convocatoria ordinaria.