

Facultad de Ciencias

GUÍA DOCENTE DE LA ASIGNATURA

G658 - Ingeniería del Software I

Grado en Ingeniería Informática
Obligatoria. Curso 3

Curso Académico 2023-2024

1. DATOS IDENTIFICATIVOS					
Título/s	Grado en Ingeniería Informática			Tipología y Curso	Obligatoria. Curso 3
Centro	Facultad de Ciencias				
Módulo / materia	MATERIA INGENIERÍA DEL SOFTWARE Y SISTEMAS DE INFORMACIÓN MÓDULO OBLIGATORIO				
Código y denominación	G658 - Ingeniería del Software I				
Créditos ECTS	6	Cuatrimestre	Cuatrimestral (1)		
Web					
Idioma de impartición	Español	English friendly	No	Forma de impartición	Presencial

Departamento	DPTO. INGENIERÍA INFORMÁTICA Y ELECTRÓNICA				
Profesor responsable	CARLOS BLANCO BUENO				
E-mail	carlos.blanco@unican.es				
Número despacho					
Otros profesores					

2. CONOCIMIENTOS PREVIOS	
Existen dependencias entre temas de esta materia y temas de las materias de "Sistemas de Información" y 'Bases de Datos'. Por ello, es recomendable seguir la planificación temporal diseñada para el plan de estudios.	

3. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS DEL PLAN DE ESTUDIOS TRABAJADAS

Competencias Genéricas

Capacidad de análisis, síntesis y evaluación.

Capacidad de gestión de la información.

Capacidad de resolución de problemas aplicando técnicas de ingeniería.

Capacidad de trabajo en equipo.

Capacidad de relación interpersonal.

Aprendizaje autónomo.

Adaptación a nuevas situaciones.

Creatividad.

Capacidad de liderazgo.

Tener motivación por la calidad.

(Aplicación) Aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

(Análisis) Reunir e interpretar datos relevantes (dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.

(Comunicación) Transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

(Aprendizaje) Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

Competencias Específicas

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.

Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.

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.

3.1 RESULTADOS DE APRENDIZAJE

- Conocer los principios y fundamentos de la Ingeniería del Software y sus principales conceptos: ciclo de vida, metodología, proceso.

- Conocer y saber aplicar una metodología de desarrollo de software.

- Modelar requisitos funcionales y no funcionales de un sistema software.

- Modelar el diseño arquitectónico de un sistema software.

- Modelar el diseño detallado de un sistema software.

- Saber usar herramientas que dan soporte al modelado de sistemas software.

4. OBJETIVOS

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 aplicar un proceso de desarrollo de software, incluyendo las etapas de requisitos, arquitectura y diseño detallado.

Ser capaz de modelar software utilizando UML.

5. MODALIDADES ORGANIZATIVAS Y MÉTODOS DOCENTES	
ACTIVIDADES	HORAS DE LA ASIGNATURA
ACTIVIDADES PRESENCIALES	
HORAS DE CLASE (A)	
- Teoría (TE)	15
- Prácticas en Aula (PA)	15
- Prácticas de Laboratorio Experimental(PLE)	
- Prácticas de Laboratorio en Ordenador (PLO)	30
- Prácticas Clínicas (CL)	
Subtotal horas de clase	60
ACTIVIDADES DE SEGUIMIENTO (B)	
- Tutorías (TU)	
- Evaluación (EV)	8
Subtotal actividades de seguimiento	8
Total actividades presenciales (A+B)	68
ACTIVIDADES NO PRESENCIALES	
Trabajo en grupo (TG)	22
Trabajo autónomo (TA)	60
Tutorías No Presenciales (TU-NP)	
Evaluación No Presencial (EV-NP)	
Total actividades no presenciales	82
HORAS TOTALES	150

6. ORGANIZACIÓN DOCENTE													
CONTENIDOS		TE	PA	PLE	PLO	CL	TU	EV	TG	TA	TU-NP	EV-NP	Semana
1	FUNDAMENTOS DE INGENIERÍA DEL SOFTWARE. Cuerpo de conocimientos SWEBOK. Áreas principales. Modelado en Ingeniería Software. Lenguajes de modelado: UML. Métodos de desarrollo. Ciclos de vida del Software. Procesos del ciclo de vida.	2,00	2,00	0,00	4,00	0,00	0,00	0,50	2,00	6,00	0,00	0,00	1
2	REQUISITOS DE SISTEMAS SOFTWARE. Concepto y características. Etapas del proceso de especificación de requisitos software: captura, análisis, especificación y validación. Modelado de la fase de requisitos mediante diagramas de casos de uso, plantillas y diagramas de clases de dominio. Requisitos no funcionales.	4,00	4,00	0,00	10,00	0,00	0,00	2,50	8,00	18,00	0,00	0,00	2-6
3	ARQUITECTURA DE SISTEMAS SOFTWARE. Arquitectura Software. Concepto de Componente. Arquitecturas en 3 Capas. Modelado de la arquitectura mediante diagramas de componentes y despliegue. Modelado de escenarios arquitectónicos mediante diagramas de secuencia.	4,00	4,00	0,00	10,00	0,00	0,00	2,50	8,00	18,00	0,00	0,00	7-11
4	DISEÑO DE SISTEMAS SOFTWARE. Descripción estructural. Patrones de Diseño. Modelado mediante diagramas de clases. Comportamiento procedural y modelado mediante diagramas de actividad. Comportamiento reactivo y modelado mediante diagramas de máquinas de estado.	5,00	5,00	0,00	6,00	0,00	0,00	2,50	4,00	18,00	0,00	0,00	12-14
TOTAL DE HORAS		15,00	15,00	0,00	30,00	0,00	0,00	8,00	22,00	60,00	0,00	0,00	
Esta organización tiene carácter orientativo.													

TE	Horas de teoría
PA	Horas de prácticas en aula
PLE	Horas de prácticas de laboratorio experimental
PLO	Horas de prácticas de laboratorio en ordenador
CL	Horas de prácticas clínicas
TU	Horas de tutoría
EV	Horas de evaluación
TG	Horas de trabajo en grupo
TA	Horas de trabajo autónomo
TU-NP	Tutorías No Presenciales
EV-NP	Evaluación No Presencial

7. MÉTODOS DE LA EVALUACIÓN

Descripción	Tipología	Eval. Final	Recuper.	%
Examen	Examen escrito	Sí	Sí	60,00
Calif. mínima	5,00			
Duración	4 horas			
Fecha realización	Las establecidas por el centro			
Condiciones recuperación				
Observaciones	Consiste en un examen escrito compuesto por preguntas teórico prácticas.			
Desarrollo de Proyecto	Trabajo	No	Sí	40,00
Calif. mínima	5,00			
Duración	Durante del cuatrimestre de forma continua			
Fecha realización	A lo largo del cuatrimestre			
Condiciones recuperación	Realización de proyecto similar			
Observaciones	<p>Los alumnos deberán desarrollar los modelos software correspondientes a un proyecto. Se realizará a modo de portafolio en el que a medida que se vayan estudiando los distintos artefactos software en clases teóricas, se vayan desarrollando e incorporando en el porfolio del proyecto.</p> <p>Dicho proyecto se realizará en grupo, y para cada fase se definirán una serie de tareas individuales a realizar (la mayoría) y otras a realizar en grupo.</p> <p>Los alumnos que no superen este trabajo podrán recuperarlo en la convocatoria extraordinaria realizando un proyecto de similares características.</p>			
TOTAL				100,00
Observaciones				
Criterios de evaluación para estudiantes a tiempo parcial				
El alumnado a tiempo parcial podrá gestionar su tiempo compatibilizándolo con sus factores profesionales o personales. Dispondrá del material de clase en Moodle y podrá realizar a distancia el desarrollo de proyecto práctico evaluable. El examen final deberá ser realizado de forma presencial en la fecha marcada por el centro.				

8. BIBLIOGRAFÍA Y MATERIALES DIDÁCTICOS

BÁSICA
Sommerville, 2012. Ingeniería del Software. 9ª Edición, Addison-Wesley. 2012.
J.Arlow e I.Neustadt. UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design. Addison-Wesley. 2005.
D.Pilone. UML 2.0 in a Nutshell. O'Reilly. 2005.
Complementaria
IEEE Computer Society, SWEBOK - Guide to the Software Engineering Body of Knowledge, v3, 2014.
A.Cockburn, Writing Effective Use Cases. Addison-Wesley, 2000.
R.N.Taylor, N.Medvidovic y E.M.Dashofy, Software Architecture: Foundations, Theory, and Practice. 2009.
E.Gamma, Patrones de Diseño. Addison-Wesley. 2002.
Pressman, 2010. Ingeniería del Software: Un Enfoque Práctico. 7ª Edición. McGraw-Hill, 2010.
Pfleeger, 2009. Software Engineering: Theory and Practice. Prentice Hall, 2009.
Jacobson et al., 2000. El Proceso Unificado de Desarrollo de Software. Addison-Wesley, 2000.
Rumbaugh et al., 2007. El Lenguaje Unificado de Modelado. Manual de Referencia, 2da edic. Pearson Educación, 2007.
Booch et al., 2006. El Lenguaje Unificado de Modelado. Guía del Usuario, 2da edic. Pearson Educación, 2006.

9. SOFTWARE				
PROGRAMA / APLICACIÓN	CENTRO	PLANTA	SALA	HORARIO
Editor UML (Magic Draw, StarUML, etc.)	Facultad de Ciencias			
Paquete de Ofimática Básico	Facultad de Ciencias			

10. COMPETENCIAS LINGÜÍSTICAS	
<input type="checkbox"/> Comprensión escrita	<input type="checkbox"/> Comprensión oral
<input type="checkbox"/> Expresión escrita	<input type="checkbox"/> Expresión oral
<input type="checkbox"/> Asignatura íntegramente desarrollada en inglés	
Observaciones	