

Facultad de Ciencias

GUÍA DOCENTE DE LA ASIGNATURA

G658 - Ingeniería del Software I

Grado en Ingeniería Informática
Obligatoria. Curso 3

Curso Académico 2018-2019

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	Facultad de Ciencias. Planta: + 1. DESPACHO PROFEROR (1070)			
Otros profesores	PATRICIA LOPEZ MARTINEZ			

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 organización y planificación.
Capacidad de gestión de la información.
Capacidad de resolución de problemas aplicando técnicas de ingeniería.
Capacidad para argumentar y justificar lógicamente las decisiones tomadas y las opiniones.
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.
Razonamiento crítico.
Capacidad de iniciativa y espíritu emprendedor.
Poseer una capacidad demostrada para la comunicación oral y escrita así como para hacer presentaciones efectivas en público.
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 desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la Ingeniería del Software.
Capacidad para valorar las necesidades del cliente y especificar los requisitos software para satisfacer estas necesidades, reconciliando objetivos en conflicto mediante la búsqueda de compromisos aceptables dentro de las limitaciones derivadas del coste, del tiempo, de la existencia de sistemas ya desarrollados y de las propias organizaciones.
Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.
Capacidad para diseñar soluciones apropiadas en uno o más dominios de aplicación utilizando métodos de la ingeniería del software que integren aspectos éticos, sociales, legales y económicos.
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.
- Capturar, especificar y modelar requisitos de usuario y de un sistema software.
- Construir los modelos de diseño a alto nivel y detallados 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 y modelos que resulten de utilidad para cada tipo de aplicación, de acuerdo con el dominio de la misma.

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)	20
- Prácticas en Aula (PA)	10
- Prácticas de Laboratorio (PL)	30
- Horas Clínicas (CL)	
Subtotal horas de clase	60
ACTIVIDADES DE SEGUIMIENTO (B)	
- Tutorías (TU)	7,5
- Evaluación (EV)	7,5
Subtotal actividades de seguimiento	15
Total actividades presenciales (A+B)	75
ACTIVIDADES NO PRESENCIALES	
Trabajo en grupo (TG)	25
Trabajo autónomo (TA)	50
Tutorías No Presenciales (TU-NP)	
Evaluación No Presencial (EV-NP)	
Total actividades no presenciales	75
HORAS TOTALES	150

6. ORGANIZACIÓN DOCENTE

CONTENIDOS		TE	PA	PL	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	0,00	6,00	0,00	0,50	0,50	4,00	8,00	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.	6,00	3,00	8,00	0,00	2,50	2,50	8,00	15,00	0,00	0,00	2-5
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.	6,00	3,00	8,00	0,00	2,50	2,50	8,00	15,00	0,00	0,00	6-9
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.	6,00	4,00	8,00	0,00	2,00	2,00	5,00	12,00	0,00	0,00	10-14
TOTAL DE HORAS		20,00	10,00	30,00	0,00	7,50	7,50	25,00	50,00	0,00	0,00	

Esta organización tiene carácter orientativo.

TE	Horas de teoría
PA	Horas de prácticas en aula
PL	Horas de prácticas de laboratorio
CL	Horas 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 Teórico	Examen escrito	Sí	Sí	50,00
Calif. mínima	4,50			
Duración	4			
Fecha realización	Al finalizar el cuatrimestre. Fijada por el centro			
Condiciones recuperación	Examen en convocatoria de Septiembre			
Observaciones	<p>Se realizará una prueba escrita al finalizar el cuatrimestre, en el día fijado por el centro. Su duración será aproximadamente de cuatro horas.</p> <p>Consistirá en una parte de preguntas cortas y otra de ejercicios.</p> <p>La parte de preguntas cortas se realizará sin apuntes y su peso será inferior al 50% de la calificación final de la prueba.</p> <p>Para la parte de ejercicios se dejará hacer uso de todo tipo de material escrito, incluyendo apuntes y/o ejercicios. En ningún caso se dejará hacer uso de portátiles, teléfonos móviles u otros dispositivos electrónicos con capacidades de comunicación inalámbricas.</p> <p>La prueba escrita se entenderá como superada cuando la calificación obtenida sea superior o igual 4,5.</p>			
Desarrollo de Proyecto	Trabajo	No	Sí	50,00
Calif. mínima	4,50			
Duración	Durante del cuatrimestre de forma continua			
Fecha realización	A lo largo del cuatrimestre			
Condiciones recuperación	Realización de proyecto similar o prueba en laboratorio			
Observaciones	<p>Los alumnos deberán desarrollar un proyecto integrado donde se pongan en práctica los conceptos aprendidos en las clases teóricas.</p> <p>Se crearán grupos de trabajo y cada uno definirá su proyecto. En su desarrollo habrá que realizar una serie de actividades individuales y en grupo. Por ello, se destinará a la realización del mismo un número determinado de sesiones de laboratorio más un cierto porcentaje de las horas de trabajo autónomo, tanto en grupo como individual.</p> <p>El proyecto se realizará de forma continua durante todo el cuatrimestre. Se realizarán entregas parciales al finalizar cada fase del proyecto y una entrega final del proyecto completo al finalizar el cuatrimestre. Para superar el trabajo habrá que obtener una calificación mínima de 4,5 puntos.</p> <p>Los alumnos que no superen este trabajo podrán recuperarlo en la convocatoria de Septiembre realizando un proyecto de similares características.</p>			
TOTAL				100,00
Observaciones				
Criterios de evaluación para estudiantes a tiempo parcial				
<p>El alumnado a tiempo parcial podrá realizar a distancia todas las actividades teóricas y prácticas que se realicen en el aula. De esta forma, podrá gestionar su tiempo compatibilizándolo con sus factores profesionales o personales.</p> <p>La parte teórica 50% se evaluará de igual forma que el alumnado a tiempo completo, mediante un examen teórico. Todas las actividades no evaluables (como los ejercicios propuestos) podrá realizarlas a distancia y solicitar su corrección enviándolos por correo o mediante tutorías.</p> <p>Para la parte práctica 50% tendrá que desarrollar un proyecto con las mismas entregas que el alumnado a tiempo completo, sin embargo, el proyecto será individual y podrá disponer de más flexibilidad en los plazos de entrega, para de este modo compatibilizar el estudio de la asignatura con sus restricciones de alumno a tiempo parcial.</p>				

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.
IEEE Computer Society, SWEBOK - Guide to the Software Engineering Body of Knowledge, v3, 2014.
A.Cockburn, Writting 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.
Complementaria
Pressman, 2010. Ingeniería del Software: Un Enfoque Práctico. 7ª Edición. McGraw-Hill, 2010.
Pfleeger, 2009. Software Engineering: Theory and Practice. Prentice Hall, 2009.
Object Management Group, 2005: Unified Modelling Language, version 2.0. Disponible en http://www.omg.org/spec/UML/2.0/ .
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.
E.Freeman, E.Robson, B.Bates y K.Sierra. Head First Design Patterns. O'Reilly Media. 2004.
Larman, 2003. UML y Patrones: Introducción al análisis y diseño orientado a objetos, 2ª Edición, Prentice Hall.
M.Piattini et al., 2007. Análisis y diseño de Aplicaciones Informáticas de Gestión. Una perspectiva de Ingeniería del Software. Ra-Ma. 2007.

9. SOFTWARE

PROGRAMA / APLICACIÓN	CENTRO	PLANTA	SALA	HORARIO
Magic Draw (Editor UML)	Facultad de Ciencias			
Microsoft Visio	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	