

**GUÍA DOCENTE**

**2014-2015**

**1. DESCRIPCIÓN DE LA ASIGNATURA**

<b>Grado:</b>	<b>Ingeniería Informática en Sistemas de Información</b>
<b>Doble Grado:</b>	
<b>Asignatura:</b>	<b>Ingeniería del Software 1</b>
<b>Módulo:</b>	<b>M3: Ingeniería del Software, Sistemas de Información y Sistemas Inteligentes</b>
<b>Departamento:</b>	<b>Deporte e Informática</b>
<b>Año académico:</b>	<b>2014-2015</b>
<b>Semestre:</b>	<b>Primer semestre</b>
<b>Créditos totales:</b>	<b>6</b>
<b>Curso:</b>	<b>2º</b>
<b>Carácter:</b>	<b>Obligatoria</b>
<b>Lengua de impartición:</b>	<b>Español</b>

<b>Modelo de docencia:</b>	<b>C1</b>	
<b>a. Enseñanzas Básicas (EB):</b>		<b>50%</b>
<b>b. Enseñanzas de Prácticas y Desarrollo (EPD):</b>		<b>50%</b>
<b>c. Actividades Dirigidas (AD):</b>		

## GUÍA DOCENTE

### 2. RESPONSABLE DE LA ASIGNATURA

<b>Responsable de la asignatura</b>	
<b>Nombre:</b>	Roberto Ruiz Sánchez
<b>Centro:</b>	Escuela Politécnica Superior
<b>Departamento:</b>	Deporte e Informática
<b>Área:</b>	Lenguaje y Sistemas Informáticos
<b>Categoría:</b>	Profesor Titular de Universidad
<b>Horario de tutorías:</b>	Martes 11:30-14:00 y 15:00-17:00; y miércoles 17:00-18:30h
<b>Número de despacho:</b>	11.1.14 (edificio 11, 1ª planta)
<b>E-mail:</b>	robertoruiz@upo.es
<b>Teléfono:</b>	954977591

### 3. UBICACIÓN EN EL PLAN FORMATIVO

#### 3.1. Descripción de los objetivos

La asignatura de Ingeniería del Software 1 (IS1) combina formación sobre los procesos de análisis y de diseño de un sistema software, las tecnologías de programación orientadas a objetos y la notación UML (Lenguaje unificado de modelado). También se pretende que el alumno tenga una visión general del proceso de Ingeniería del software, y de los conceptos de prueba y mantenimiento del software.

#### 3.2. Aportaciones al plan formativo

En las asignaturas de Fundamentos de la Programación y de Programación Orientada a Objetos, los alumnos aprenden rudimentos de la programación. En esta asignatura de Ingeniería del Software 1 se introduce al alumno en las técnicas para realizar los procesos básicos incluidos en las fases de análisis y diseño de software. Esta asignatura se corresponde directamente con una asignatura posterior, Ingeniería del Software 2, donde se incluirán los conceptos relacionados con la arquitectura del software y se revisarán con más detalle los procesos de desarrollo de software.

#### 3.3. Recomendaciones o conocimientos previos requeridos

Los alumnos deberían tener conocimientos de programación, de algoritmos básicos y de estructuras de datos simples.

## GUÍA DOCENTE

### 4. COMPETENCIAS

#### 4.1 Competencias de la Titulación que se desarrollan en la asignatura

4.1.1. Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, de acuerdo con los conocimientos adquiridos, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas. (Competencia G01)

4.1.2. Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas. (Competencia G04)

4.1.3. Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad. (Competencia G05)

4.1.4. Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero en Informática. (Competencia G09)

#### 4.2. Competencias del Módulo que se desarrollan en la asignatura

4.2.1. Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería. (Competencia EB5)

4.2.2. Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados. (Competencia EC08)

4.2.3. Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software. (Competencia EC16)

4.2.4. Capacidad para participar activamente en la especificación, diseño, implementación y mantenimiento de los sistemas de información y comunicación. (Competencia ET3)

#### 4.3. Competencias particulares de la asignatura

4.3.1. Tener una visión general del proceso de desarrollo de software.



## GUÍA DOCENTE

4.3.2. Identificar y analizar los requisitos del sistema.

4.3.3. Enseñar al alumno el papel fundamental que desempeña el diseño dentro del proceso software. Comprender la importancia de aplicar principios y patrones de diseño.

4.3.4. Modelar un sistema de información mediante el lenguaje UML.

4.3.5. Conocer los conceptos básicos de las pruebas del software.

4.3.6. Conocer los conceptos básicos de mantenimiento del software.

## GUÍA DOCENTE

### 5. CONTENIDOS DE LA ASIGNATURA (TEMARIO)

Tema 1: Introducción al proceso de desarrollo de software. Elementos de la ingeniería del software. Ciclo de vida del software.

Tema 2: Análisis de requisitos. Recopilación de requisitos de un sistema software. Especificación de sistemas mediante UML. Modelo de casos de uso. Construcción del modelo de Dominio.

Tema 3: Diseño de sistemas. Introducción al diseño orientado a objetos con UML. Principios y patrones de diseño.

Tema 4: Introducción al proceso de pruebas y mantenimiento del software.

### 6. METODOLOGÍA Y RECURSOS

Los conceptos principales se explican y debaten en clase de Enseñanzas Básicas cuya documentación está disponible con suficiente antelación en la página Web de la asignatura. Se trata de clases dinámicas donde los estudiantes y profesores interactúan debatiendo aspectos importantes de los ejemplos y ejercicios que en estas clases se proponen.

Además el estudiante dispone con antelación en la página Web de la asignatura de la documentación de las Enseñanzas Práctica y de Desarrollo correspondiente que se llevará a cabo para una mejor comprensión de los conceptos desarrollados en la clase de Enseñanzas Básicas. En dicha documentación se introduce al alumno en la resolución de problemas mediante unos experimentos guiados, y se enumeran una serie de ejercicios y problemas que el alumno debe resolver en clase y en su casa. Esta actividad permite el trabajo en pequeños grupos de dos o tres alumnos, o de manera individual. Se realizan con grupos inferiores a 15 alumnos, por lo que el seguimiento del profesor es cercano al trabajo realizado por el alumno, pudiendo comprobar la evolución del trabajo realizado por cada uno.

En general, siempre se pretende la integración de los procedimientos y modalidades didácticas de la enseñanza presencial con los de la enseñanza virtual. La asignatura dispone de una página Web de comunicación ágil y amigable que favorece los procesos de intercambio de información y la comunicación de experiencias. Además del espacio para colgar el material necesario para las clases (transparencias de teoría, boletines de prácticas, soluciones de los ejercicios, notas, etc.) y los anuncios de los distintos eventos que acontezcan, esta página dispone de espacios reservados para el foro de la



## GUÍA DOCENTE

asignatura. En el foro se debaten algunos temas que los alumnos desean y otros propuestos por los profesores. Se fomenta el uso de las tutorías a través del foro y correo electrónico y se fomenta la plataforma WebCT disponible en nuestra Universidad.

## GUÍA DOCENTE

### 7. EVALUACIÓN

La evaluación de la asignatura se acogerá al modelo de evaluación continua y se basará principalmente en los conocimientos adquiridos tanto en clase de teoría como en el laboratorio de informática. La participación también podrá ser evaluada.

La evaluación de la asignatura se dividirá en dos evaluaciones parciales, la primera incluirá los temas 1 y 2, mientras que los temas 3 y 4 se valorarán en la segunda.

Cada evaluación parcial constará de pruebas obligatorias que podrán ser individuales y en grupo. Para dichos ejercicios se podrá exigir la resolución de parte de los mismos de manera autónoma no presencial.

Una evaluación parcial se considerará aprobada cuando la suma ponderada de los ejercicios correspondientes sea mayor o igual a 5 puntos en una escala de 0 a 10.

Para aprobar la asignatura por curso es necesario aprobar ambas evaluaciones parciales, en cuyo caso la calificación final será la media aritmética de ambas.

#### CONVOCATORIA DE RECUPERACIÓN (JUNIO/JULIO)

Los estudiantes que no superen alguna de las dos partes de la asignatura en la convocatoria de curso anteriormente descrita, dispondrán de una convocatoria de recuperación de curso en el mes de junio/julio para evaluarse de la parte no superada.

La convocatoria de recuperación seguirá la misma estructura que la evaluación continua, es decir, la asignatura se evaluará mediante dos pruebas de evaluación independientes, una por cada parte anteriormente mencionada, con un peso del 50% por cada prueba sobre la calificación final, siendo necesario aprobar ambas evaluaciones parciales para aprobar la asignatura, en cuyo caso la calificación final será la media aritmética de ambas. Estas pruebas podrán ser realizadas mediante examen escrito o en aulas de informática.

Para más información, consúltese la Normativa de Evaluación de los Estudiantes de Grado de la UPO (BUPO 7/2014 del 3 de junio de 2014).

### 8. BIBLIOGRAFÍA GENERAL





## GUÍA DOCENTE

Ingeniería del Software (Edición 7 traducida). Ian Sommerville. Editorial Pearson, Addison Wesley, 2005. ISBN: 8478290745

Ingeniería del Software. Una perspectiva orientada a objetos. Eric J. Braude. Adaptado por Macario Polo Usaola. Editorial Ra-Ma, 2003. ISBN 970-15-0851-3

Ingeniería del Software. Un enfoque práctico (Edición 6 Traducido), 2005. Roger S. Pressman. Editorial Mac Graw Hill. ISBN: 9701054733

UML y Patrones (2ª edición traducida). Craig Larman. Editorial Pearson, Prentice Hall, 2003. ISBN: 8420534382

UML para programadores Java. R.C. Martin. Pearson Education, 2004. ISBN: 8420541095.

UML distilled third edition. Martin Fowler. Addison-Wesley, 2004. ISBN: 0321193687

UML 2. Iniciación, ejemplos y ejercicios corregidos. L. Debrauwer. Ed. Eni, 2009. ISBN: 978-2-7460-4741-9