GUÍA DOCENTE

INGENIERÍA DE SOFTWARE

Coordinación: SENDÍN VELOSO, MONTSERRAT

Año académico 2017-18

INGENIERÍA DE SOFTWARE 2017-18

Información general de la asignatura

<table>
<tr><td>Denominación</td><td>INGENIERÍA DE SOFTWARE</td></tr>
<tr><td>Código</td><td>102018</td></tr>
<tr><td>Semestre de impartición</td><td>1R Q(SEMESTRE) EVALUACIÓN CONTINUADA</td></tr>
<tr><td>Carácter</td><td>
<table>
<tr><th>Grado/Máster</th><th>Curso</th><th>Carácter</th><th>Modalidad</th></tr>
<tr><td>Doble titulación: Grado en Ingeniería Informática y Grado en Administración y Dirección de Empresas</td><td>3</td><td>OBLIGATORIA</td><td>Presencial</td></tr>
<tr><td>Grado en Ingeniería Informática</td><td>3</td><td>OBLIGATORIA</td><td>Presencial</td></tr>
<tr><td>Máster Universitario en Ingeniería Informática</td><td></td><td>COMPLEMENTOS DE FORMACIÓN</td><td>Presencial</td></tr>
</table>
</td></tr>
<tr><td>Número de créditos ECTS</td><td>6</td></tr>
<tr><td>Grupos</td><td>1GG,2GM</td></tr>
<tr><td>Créditos teóricos</td><td>3</td></tr>
<tr><td>Créditos prácticos</td><td>3</td></tr>
<tr><td>Coordinación</td><td>SENDÍN VELOSO, MONTSERRAT</td></tr>
<tr><td>Departamento/s</td><td>INFORMATICA I ENGINYERIA INDUSTRIAL</td></tr>
<tr><td>Distribución carga docente entre la clase presencial y el trabajo autónomo del estudiante</td><td>6 ECTS = (10 h de Clase presencial + 15 h de Trabajo autónomo del estudiante) x 6 = 150 h de trabajo

40% Presencial (equivalente a 60 h)
60% Trabajo autónomo (equivalente a 90 h)</td></tr>
<tr><td>Información importante sobre tratamiento de datos</td><td>Consulte este enlace para obtener más información.</td></tr>
<tr><td>Idioma/es de impartición</td><td>Preferentemente en Catalán (Castellano si algún estudiante muestra dificultades con el Catalán).</td></tr>
<tr><td>Distribución de créditos</td><td>Juan Manuel Gimeno Illa 4.5
Montserrat Sendin Veloso 4.5</td></tr>
<tr><td>Horario de tutoría/lugar</td><td>Juan Manuel Gimeno (3.20 EPS, miércoles a las 13h; otras con cita previa)
Montserrat Sendín (3.20 EPS, con cita previa)</td></tr>
</table>

INGENIERÍA DE SOFTWARE 2017-18

Profesor/a (es/as)	Dirección electrónica profesor/a (es/as)	Créditos impartidos por el profesorado	Horario de tutoría/lugar
GIMENO ILLA, JUAN MANUEL	jmgimeno@diei.udl.cat	3	
SENDÍN VELOSO, MONTSERRAT	msendin@diei.udl.cat	6	

Información complementaria de la asignatura

Asignatura que se imparte durante el 2 º semestre del 2 º curso de la titulación Grado en Ingeniería Informática.

Corresponde a la Materia "Informática" dentro del Módulo de "Formación Básica".

Se asumen los conocimientos sobre programación orientada a objetos y estructuras de datos correspondientes a las asignaturas de Programaciín II y Estrcuturas de Datos.

Objetivos académicos de la asignatura

- Conocer las bases conceptuales y los diferentes aspectos de la disciplina, entre ellos los modelos de proceso del ciclo de vida del software.

- Aplicar la técnica de los casos de uso.
- Especificar textualmente las necesidades funcionales y no funcionales de un determinado sistema software planteado a través de un enunciado (y/u otras informaciones procedentes del usuario).
- Desarrollar el diagrama de clases de un determinado sistema software siguiendo los principios del Modelado Orientado a Objetos.
- Utilizar una herramienta de modelado basada en UML
- Comprender el concepto de código como algo que evoluciona en el tiempo.
- Ser capaz de programar pruebas unitarias básicas.
- Comprender los principios fundamentales del diseño orientado a objetos.
- Reconocer el concepto de responsabilidad como fundamental para plantear un diseño orientado a objetos.

Competencias

Competencias transversales

- **EPS-11:** Capacidad de comprender las necesidades del usuario expresadas en un lenguaje no técnico.

Competències específicas

- **GII-CRI2:** 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.
- **GII-CRI12:** Conocimiento y aplicación de las características, funcionalidades y estructura de las bases de datos, que permitan su adecuado uso, y el diseño y el análisis e implementación de aplicaciones basadas

en ellos.
- **GII-CRI13:** Conocimiento y aplicación de las herramientas necesarias para el almacenamiento, procesamiento y acceso a los Sistemas de información, incluidos los basados en web.
- **GII-CRI16:** Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.
- **GII-CRI17:** Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.

Contenidos fundamentales de la asignatura

Tema I - *Aspectos introductorios*

1.1. Cuestiones iniciales sobre la Ingeniería del Software

1.2. Un poco de historia

1.3. Proceso de desarrollo de software

1.4. Modelos de proceso de software

1.5. Conclusiones

Tema II - *Análisis de requisitos*

2.1. Especificación de requisitos

2.2. La técnica de los Casos de uso

2.3. Un paso más en la especificación: Los Diagramas de Secuencia del Sistema

2.4. Conclusiones

Tema III - *Análisis del Dominio*

3.1. Diagramas de Clases del Análisis

3.2. Un paso más en el análisis del dominio: los Contratos de las operaciones

3.3. Conclusiones

Tema IV - *Introducción al diseño y pruebas unitarias*

4.1. El código como algo que varía en el tiempo.

4.2. El framework JUnit para pruebas unitarias

Tema V - *Los principios SOLID*

5.1. Principio de responsabilitat única

5.2. Principio abierto-cerrado

5.3. Principio de substitución de Liskov

5.4. Principio de segregación de interficies

5.5. Principio de inversión de dependencias

Tema VI - *Diseño basado en responsabilidades*

6.1. El concepto de responsabilidad

6.2. Patrones GRASP de assignación de responsabilidades

Ejes metodológicos de la asignatura

Grupos Grandes: Clases Teoría (3 créditos)

- Parte teórica: clases soportadas con transparencias y/o apuntes
- Parte de aplicación práctica: se trabaja siempre con ejemplos. Se dispone de una **colecció de problemas**, y en clase se trabaja la resolución de problemas concretos. Se van proporcionando las soluciones a lo largo del cuatrimestre.

Grupos Medianos: Clases Laboratorio (3 créditos)

- Clases dirigidas y seguimiento personalizado por grupos de prácticas
- Uso de la herramienta de Modelado en UML: **ArgoUML** y/o **Visual Paradigm**
- Herramientas de control de versiones y framework de testing
- Trabajo continuado alrededor de un cierto **enunciado de práctica**, que simulará el desarrollo de un proyecto de software

Trabajo Autónomo (no presencial):

- La práctica se completará en horas no presenciales
- **Se recomienda** que el alumno resuelva por cuenta propia los problemas de la **colección de problemas**, con objeto de obtener feedback por parte del profesor

El **sistema de evaluación** (detallado en el apartado correspondiente) consta de: 1) pruebas escritas (los 2 exámenes parciales); y 2) prácticas (a desarrollar individualmente o en equipo dependiendo de cada caso).

Plan de desarrollo de la asignatura

Semana	Teoría (GG)	Laboratorio (GM)	Trabajo autónomo
1	Presentación Assignatura T1: Aspectos introductorios	T1: Aspectos introductorios	Estudio
2	T1: Aspectos introductorios	T1: Aspectos introductorios	Estudio
3	T2: Análisis de requisitos Especificación de requisitos	Aplicación Análisis de requisitos en enunciado de práctica	Estudio y resolución problemas (Colección de problemas)
4	T2: Análisis de requisitos La tècnica de Casos de Uso. Problemas	Uso herramienta modelado UML, aplicación tècnica de Casos de Uso en enunciado de práctica	Estudio y resolución problemas (Colección de problemas)
5	T2: Análisis de requisitos Diagramas de Secuencia del Sistema. Problemas	Aplicación técnica de Casos de Uso en enunciado práctica	Estudio, resolución problemas (Colección de problemas) y desarrollo práctica

6	T3: Análisis del Dominio Tècnica de Modelado Orientado a Objetos	Uso herramienta modelado UML Aplicación DSS en enunciat de pràctica	Estudio, resolució problemes (Colección de problemas) y desarrollo pràctica
7	T3: Análisis del Dominio Problemas	Uso herramienta modelado UML Aplicación Modelo del Dominio en enunciat de pràctica	Estudio, resolució problemes (Colección de problemas) y desarrollo pràctica
8	T3: Análisis del Dominio Contratos de las operaciones	Aplicación Contratos en enunciado de práctica	Estudio, resolució problemes (Colección de problemas) y desarrollo práctica
9	Primer parcial		
10	T4: Introducció al disseny Concepte de proves	Uso de Git	Estudio
11	T4: Junit Objectes substituts	Uso de Git	Estudio y resolución de problemas
12	T5: Principios SOLID Intro, OCP & LSP	Problemas simples testing	Estudio y práctica testing
13	T5: Principios SOLID SRP, ISP & DIP	T6: Patrones GRASP Concepto de responsabilidad	Estudio, resolución de problemas y práctica testing
14	T6: Patrones GRASP Experto, creador, bajo acoplamiento	Testing con sustituciones	Estudio, resolución de problemas y práctica testing
15	T6: Patrones GRASP Alta cohesión, controlador	Testing con sustituciones	Estudio, resolución de problemas i práctica testing
16	Semana segundo parcial		
17	Semana segundo parcial		
18	Tutorías		
19	Recuperación		

Sistema de evaluación

Activd.	Descripción	Ponderación	Nota mínima	En grupo	Presencial	Obligatoria	Recuperable
Parc1	Primer parcial	25%	3,0	No	Sí	Sí	Sí
Parc2	Segundo parcial	25%	3,0	No	Sí	Sí	Sí
Actv1	Análisis de Requisitos	20%	No	Sí	No	Sí	No
Actv2	Modelo del Dominio y Contratos	10%	No	Sí	No	Sí	No
Actv3	Pruebas unitarias	20%	No	No	No	Sí	No

Nota final = 0,25 * Parc1 + 0,25 * Parc2 + 0,20 * Actv1 + 0,10 * Actv2 + 0,20 * Actv3

- La asignatura se aprueba si la **nota final** es superior a **5** y se llega a las notas mínimas en los parciales.

Otras consideraciones y criterios:

- Tipología de los exámenes parciales: fijación de conceptos y resolución de problemas.
- Para todas las actividades evaluables: Entregas programadas, fechas no prorrogables.
- Caso de no llegar a la nota mínima en alguno de los exámenes parciales, la nota final será un 4,5 como máximo.
- Al examen de RECuperación se debe examinar, como mínimo, del parcial con nota más baja.

-

Bibliografía y recursos de información

Bibliografía básica

- C. Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Prentice-Hall, 2005 (3ª ed.)
- P. Tahchiev et al.: Junit in Action (2nd edition). Manning, 2011

Bibliografía complementaria

- G. Kotonya, I. Sommerville: Requirements Engineering: Processes and Techniques. Wiley, 1998
- R. C. Martin: Agile Software Development: Principles, Patterns, and Practices, Prentice-Hall, 2002