DISEÑO TECNOLÓGICO DE SISTEMAS DE INFORMACIÓN
GUÍA DOCENTE CURSO 2019-20

Titulación:	Grado en Ingeniería Informática				**801G**
Asignatura:	Diseño tecnológico de sistemas de información				**444**
Materia:	Ingeniería del software y sistemas de información				
Módulo:	Ingeniería del software y sistemas de información				
Modalidad de enseñanza de la titulación:	Presencial		**Carácter:**	Obligatoria	
Curso:	3	**Créditos ECTS:** 6,00	**Duración:**	Semestral (Primer Semestre)	
Horas presenciales:	60,00		**Horas estimadas de trabajo autónomo:**	90,00	

Idiomas en que se imparte la asignatura:	Español
Idiomas del material de lectura o audiovisual:	Inglés, Español

DEPARTAMENTOS RESPONSABLES DE LA DOCENCIA

MATEMÁTICAS Y COMPUTACIÓN					**R111**
Dirección:	C/ Madre de Dios, 53			**Código postal:**	26006
Localidad:	Logroño		**Provincia:**	La Rioja	
Teléfono:	941299452	**Fax:** 941299460	**Correo electrónico:**	dpto.dmc@unirioja.es	

PROFESORADO PREVISTO

Profesor:	Pérez Valle, Beatriz			**Responsable de la asignatura**	
Teléfono:	941299608	**Correo electrónico:**	beatriz.perez@unirioja.es		
Despacho:	3235	**Edificio:**	CENTRO CIENTÍFICO TECNOLÓGICO	**Tutorías:**	Consultar

Profesor:	Rubio García, Ángel Luis				
Teléfono:	941299449	**Correo electrónico:**	arubio@unirioja.es		
Despacho:	3222	**Edificio:**	CENTRO CIENTÍFICO TECNOLÓGICO	**Tutorías:**	Consultar

DESCRIPCIÓN DE LOS CONTENIDOS

- Lenguaje estándar para la especificación, visualización y documentación de sistemas.
- Diseño y evaluación de interfaces persona-computador. Accesibilidad y usabilidad de los sistemas informáticos.
- Paradigma de Programación orientada a aspectos (POA).
- Puesta en práctica (mediante miniproyectos) de las primeras fases del desarrollo de sistemas informáticos.

REQUISITOS PREVIOS DE CONOCIMIENTOS Y COMPETENCIAS PARA PODER CURSAR CON ÉXITO LA ASIGNATURA

Recomendados para poder superar la asignatura.

- Comprender el tratamiento de los conceptos de programación orientada a objetos en distintas plataformas y lenguajes de desarrollo.
- Conocer los conceptos básicos (metodologías, ciclos de vida, pruebas, etc.) de la Ingeniería del Software.
- Conocer a nivel básico un lenguaje estándar para la modelización (fundamentalmente orientada a objetos) de sistemas software

Asignaturas que proporcionan los conocimientos y competencias:

- Ingeniería del Software
- Programación orientada a objetos

CONTEXTO

Esta asignatura se entiende como una continuación natural de *Ingeniería de Software* (2º curso/2º semestre). En *Ingeniería de Software* se adquieren los conocimientos y competencias básicas correspondientes a esta disciplina (nociones generales sobre metodologías y ciclos de vida para el desarrollo de sistemas software, estudio del lenguaje estándar para el modelización, especificación, visualización y documentación de sistemas software, etc.)
En *Diseño Tecnológico de Sistemas de Información* se pretende afianzar los conocimientos, destrezas y habilidades adquiridos en *Ingeniería del Software*, en particular estudiando con un cierto nivel de detalle algunas metodologías concretas, y profundizando en el conocimiento del lenguaje estándar de modelización.
Además esta asignatura se presenta con un enfoque eminentemente práctico, para lo que se propone la realización de

pequeños proyectos que abarquen las primeras fases de cualquier metodología. Los proyectos serán elaborados en equipo (grupo pequeño) bajo la supervisión/tutorización de un profesor de la asignatura. Algunos roles de dichos equipos requerirán diferentes capacidades y niveles de compromiso por parte de los alumnos, por lo que se establecerán turnos rotatorios de roles. Existen distintas posibilidades para el desarrollo concreto de los proyectos (distintos proyectos para distintos equipos; mismo proyecto para todos/algunos equipos utilizando distintas metodologías; división de los proyectos en fases de forma que un equipo realice una fase y otro otra, etc…).
Por otra parte, en la asignatura se pretende que se adquieran los conocimientos más básicos sobre el diseño y evaluación de interfaces de usuario. En particular se pretende que los alumnos experimenten con algunas librerías para el desarrollo de GUIs en Java.

COMPETENCIAS

Competencias generales

CG1-Estar capacitado para analizar, razonar y evaluar de modo crítico, lógico y, en caso necesario, formal, sobre problemas que se planteen en su entorno.
CG2-Estar capacitado para, utilizando el nivel adecuado de abstracción, establecer y evaluar modelos que representen situaciones reales.
CG3-Estar capacitado para encontrar, relacionar, estructurar e interpretar datos, información y conocimiento provenientes de diversas fuentes.
CG4-Estar capacitado para transmitir información, ideas, planteamiento de problemas y soluciones, tanto a otros profesionales tecnológicos y científicos, como a personas ajenas a esas disciplinas.
CG5-Estar capacitado tanto para trabajar autónomamente, como para integrarse de modo eficaz en equipos de trabajo.
CG7-Haber desarrollado aquellas habilidades de aprendizaje necesarias para continuar su formación.
CG8-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.
CG9-Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática.
CG11-Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.
CG12-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.
CG15-Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.
CG16-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 Técnico en Informática.
CG17-Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática.
CG19-Conocimiento y aplicación de elementos básicos de economía y de gestión de recursos humanos, organización y planificación de proyectos, así como la legislación, regulación y normalización en el ámbito de los proyectos informáticos.

Competencias específicas

CE7-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.
CE14-Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
CE22-Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.
CE23-Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.
CE25-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.
CE26-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.
CE28-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.
CE30-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.

RESULTADOS DEL APRENDIZAJE

- Saber aplicar conceptos básicos (metodologías, fases de ciclos de vida, etc.) de la Ingeniería del Software.
- Saber aplicar métodos, técnicas, lenguajes y herramientas para realizar las tareas propias (análisis, diseño, construcción, pruebas, etc.) de la Ingeniería del Software.

- Saber analizar, identificar y definir los requisitos que debe cumplir un sistema informático.
- Profundizar en el conocimiento de un lenguaje estándar para la modelización (fundamentalmente orientada a objetos) de sistemas software.
- Desarrollar habilidades y destrezas para modelar e implementar soluciones ajustadas al paradigma de programación POA.
- Ser capaz de diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad de los sistemas informáticos.
- Ser capaz de evaluar las necesidades del cliente y extraer los requisitos software para satisfacerlas.
- Saber actuar como consultor interno, ofreciendo asesoramiento y orientación.
- Definir y evaluar y seleccionar plataformas HW/SW. Poder diseñar integrando HW/SW y redes.
- Saber dirigir el diseño de soluciones con los productos HW/SW disponibles.
- Ser capaz de llevar a cabo proyectos de tamaño pequeño o mediano, que incluyan la realización, en equipo o de manera individual, de las tareas propias de la Ingeniería del Software.

TEMARIO

Tema 1. Introducción/Presentación: Análisis y Diseño de Sistemas de Información
Tema 2. Introducción a la Interacción Persona/Ordenador.
Tema 3. Programación Orientada a Aspectos.
Tema 4. UML: Repaso de Diagramas de Casos de Uso y de Actividad
Tema 5. UML: Repaso y aspectos avanzados de Diagramas de Clases.
Tema 6. UML: Diagramas de Interacción
Tema 7. UML: Diagramas de Estados
Tema 8. UML: Otros Diagramas

BIBLIOGRAFÍA

Tipo:	Título
Básica	El lenguaje unificado de modelado UML Absys Biba
Básica	El proceso unificado de desarrollo de software Absys Biba
Básica	El lenguaje unificado de modelado. Manual de referencia Absys Biba
Básica	Unified modeling language user guide Absys Biba
Básica	UML @ Classroom Absys Biba
Básica	The essential guide to user interface design: an introduction to guide design principles and techniques. Absys Biba
Básica	Diseño de interfaces de usuario: estrategias para una interacción persona-computadora efectiva Absys Biba
Básica	AspectJ in action: enterprise AOP with Spring applications Absys Biba
Básica	Mastering AspectJ: Aspect-Oriented Programming in Java Absys Biba
Básica	Unified modeling language user guide
Complementaria	Human-computer interaction Absys Biba
Complementaria	Interaction design: beyond human-computer interaction Absys Biba
Complementaria	Interface design: the art of developing easy-to-use software. Absys Biba
Complementaria	Análisis y diseño orientado a objetos. Absys Biba
Complementaria	Object-Oriented Analysis and Design Absys Biba
Complementaria	Head first object-oriented analysis and design: [a brain-friendly guide to OOA&D] Absys Biba
Complementaria	UML gota a gota Absys Biba
Complementaria	Modeling with UML Absys Biba
Complementaria	Eclipse Aspectj: Aspect-Oriented programming with Aspectj and the Eclipse Aspectj development tools Absys Biba
Complementaria	Aspect-oriented software development Absys Biba
Complementaria	Aspect-oriented analysis and design: the theme approach Absys Biba
Complementaria	Beginning Java 8 APIs, Extensions and Libraries Absys Biba
Complementaria	Professional Java native interfaces with Swt/JFace Absys Biba

Recursos en Internet
Web oficial UML http://www.uml.org
AspectJ Project

https://www.eclipse.org/aspectj/

The AspectJ Programming Guide
http://www.eclipse.org/aspectj/doc/released/progguide/index.html

METODOLOGÍA

Modalidades organizativas

Clases teóricas
Clases prácticas
Estudio y trabajo autónomo individual

Métodos de enseñanza

Método expositivo - Lección magistral
Resolución de ejercicios y problemas
Aprendizaje basado en problemas
Aprendizaje orientado a proyectos
Aprendizaje cooperativo

ORGANIZACIÓN

Actividades presenciales	Tamaño de grupo	Horas
Clases prácticas de aula	Grande	5,00
Clases prácticas de laboratorio o aula informática	Informática	28,00
Clases teóricas	Grande	18,00
Trabajo en grupo	Grande	9,00
Total de horas presenciales		60,00

Trabajo autónomo del estudiante	Horas
Estudio autónomo individual o en grupo	35,00
Preparación de las prácticas y elaboración de cuaderno de prácticas	30,00
Preparación en grupo de trabajos, presentaciones (orales, debates,...), actividades en biblioteca	5,00
Resolución individual de ejercicios, cuestiones u otros trabajos, actividades en biblioteca o simi	20,00
Total de horas de trabajo autónomo	90,00
Total de horas	150,00

EVALUACIÓN

Sistemas de evaluación	Recuperable	No Recup.
Pruebas escritas	60%	
Trabajos y proyectos		20%
Técnicas de observación		5%
Portafolio	15%	
Total	100%	

Comentarios

Se consideran actividades de **evaluación continua** las correspondientes a:
- la **asistencia a prácticas de laboratorio** (indicado en el apartado de evaluación como “Técnicas de observación”, con un peso del **5%** de la evaluación y con carácter no recuperable),
- la **entrega de prácticas de laboratorio** a través del Campus virtual (indicado en el apartado de evaluación como parte del sistema de evaluación “Portafolio”, con un peso del **15%** de la evaluación y con carácter recuperable),
- la **entrega del trabajo en grupo** (indicado en el apartado de evaluación como “Trabajos y proyectos”, con un peso del **20%** de la evaluación y con carácter no recuperable).

Tal y como se ha indicado anteriormente, el **trabajo en grupo** supone un **20%** de la evaluación y tiene carácter **no recuperable**. Esto significa que el alumno deberá integrarse de manera activa en un grupo de trabajo para optar a la máxima calificación de la asignatura. Del mismo modo, la **asistencia continuada a las prácticas de laboratorio** (cuantificadas en un 80% de asistencia salvo causas justificadas a juicio del profesor) supondrá un **5%** de la evaluación.

La evaluación continua (**40 %**) se realizará mediante los sistemas de evaluación de técnicas de observación, realización de trabajos y proyectos, y entrega de portafolios.

Debido a las condiciones excepcionales causadas por la alerta sanitaria, la prueba final de la convocatoria extraordinaria, la cual contará con un peso del 60%, consistirá en tres pruebas más breves, correspondientes a los conocimientos 1)

teórico-prácticos, 2) de programación orientada a aspectos, y 3) de UML. La exigencia en estas pruebas se adecuará a la excepcional situación docente causada por la alerta sanitaria.
Tras la realización del examen final la profesora responsable podrá solicitar además una reunión por videoconferencia con aquellos estudiantes que considere oportuno. Los estudiantes deberán ser capaces de explicar y responder a preguntas sobre los ejercicios realizados de forma que, si no lo hacen adecuadamente, podrá modificarse la nota del examen.
El material didáctico (ejercicios prácticos, cuestiones, actividades, etc.) se encontrará disponible en el aula virtual para todos los alumnos matriculados en esta asignatura.
Para los estudiantes a tiempo parcial (reconocidos como tales por la Universidad), el profesor responsable de la asignatura podrá sustituir las actividades de evaluación no recuperable por otras a especificar en cada caso. Esta posibilidad se habilitará siempre y cuando la causa que le impida la realización de la actividad de evaluación programada sea la que ha llevado al reconocimiento de la dedicación a tiempo parcial.

Criterios críticos para superar la asignatura

Para **promediar las notas** de las diferentes partes, es necesario obtener en la prueba escrita una **nota superior o igual a un 2,4 sobre 6** (ya que la prueba escrita tiene un peso del 60% de la nota final), o lo que es lo mismo, **un 4 sobre 10** . Si la nota de dicha prueba es inferior a un 4 sobre 10, la **calificación final de la asignatura** será la nota de la prueba escrita, evaluada sobre 10.