



Universitat de Lleida

GUÍA DOCENTE
**ESPECIFICACIÓN Y ANÁLISIS
DE SISTEMAS INTERACTIVOS**

Coordinación: SAYAGO BARRANTES, SERGIO

Año académico 2022-23

Información general de la asignatura

Denominación	ESPECIFICACIÓN Y ANÁLISIS DE SISTEMAS INTERACTIVOS			
Código	102382			
Semestre de impartición	2o Q(SEMESTRE) EVALUACIÓN CONTINUADA			
Carácter	Grado/Máster	Curso	Carácter	Modalidad
	Grado en Técnicas de Interacción Digital y de Computación	2	OBLIGATORIA	Presencial
Número de créditos de la asignatura (ECTS)	6			
Tipo de actividad, créditos y grupos	Tipo de actividad	PRALAB	TEORIA	
	Número de créditos	3	3	
	Número de grupos	1	1	
Coordinación	SAYAGO BARRANTES, SERGIO			
Departamento/s	INFORMATICA E INGENIERIA INDUSTRIAL			
Distribución carga docente entre la clase presencial y el trabajo autónomo del estudiante	<p>Según el marco académico de grados de la EPS:</p> <ul style="list-style-type: none"> - 1 ECTS = 25 horas; 6 ECTS = 150 horas - 40% (60h) de trabajo presencial y 60% (90h) de trabajo autónomo <p>La distribución de horas en la asignatura es:</p> <ul style="list-style-type: none"> - Trabajo presencial: teoría (28h) + prácticas (28h) = 56h + 4 exámenes = 60h - Trabajo autónomo: estudio (45h) + proyecto (45h) = 90h 			
Información importante sobre tratamiento de datos	Consulte este enlace para obtener más información.			
Idioma/es de impartición	Castellano (apuntes). Catalán / Castellano (en el aula)			
Distribución de créditos	Véase sección tipo de actividad, créditos y grupos			

Profesor/a (es/as)	Dirección electrónica\nprofesor/a (es/as)	Créditos impartidos por el profesorado	Horario de tutoría/lugar
SAYAGO BARRANTES, SERGIO	sergio.sayago@udl.cat	6	Solicitar cita por correo-e. Despacho 12 del Campus Igualada (Pla de la Massa) / Virtual via videoconferencia

Información complementaria de la asignatura

El software domina el mundo y el desarrollo de software profesional es una actividad formal. Especificación y Análisis de Sistemas Interactivos es una asignatura de 6 ECTS, de formación específica, y de carácter obligatorio, que se imparte en el segundo semestre del segundo año del Grado en Técnicas de Interacción Digital y de Computación (GTIDIC).

El GTIDIC tiene como objetivo formar a profesionales del mundo de la computación con una vertiente muy práctica, prestando especial atención al diseño e implementación de aplicaciones interactivas. Los graduados en el GTIDIC estarán plenamente preparados para ejercer de profesionales en el mundo de las TIC, centrándose en el diseño y el desarrollo de aplicaciones interactivas.

La asignatura de Especificación y Análisis de Sistemas Interactivos tiene como principal objetivo proporcionar a los estudiantes una introducción práctica a la Ingeniería del Software, prestando especial atención al diseño (especificación y análisis), para que acaben el grado con unos conocimientos y habilidades que les permitan integrarse en equipos de desarrollo software muy rápidamente, y con calidad.

Para alcanzar dicho objetivo, los estudiantes desarrollaran un proyecto de ingeniería de software en el contexto de un proyecto común que se realiza en 4 asignaturas: Innovación, Desarrollo Aplicaciones Móviles, Experiencia de Usuario, y este curso. En el proyecto, se diseña y desarrolla una aplicación móvil innovadora y centrada en el usuario en un contexto de desarrollo ágil de software.

El software que se utiliza en la asignatura está disponible de manera gratuita en Internet, como editores online UML (Visual Paradigm Online), e IDEs, como IntelliJ IDEA

Objetivos académicos de la asignatura

1. Conocer las bases conceptuales y los diferentes aspectos de la disciplina, entre ellos los modelos de proceso del ciclo de vida del software.
2. Aplicar la técnica de los casos de uso.
3. 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).
4. Desarrollar el diagrama de clases de un determinado sistema software siguiendo los principios del Modelado Orientado a Objetos.
5. Utilizar una herramienta de modelado basada en UML
6. Comprender el concepto de código como algo que evoluciona en el tiempo.
7. Ser capaz de programar pruebas unitarias básicas.
8. Comprender los principios fundamentales del diseño orientado a objetos.
9. Reconocer el concepto de responsabilidad como fundamental para plantear un diseño orientado a objetos

Competencias

Según la tabla de competencias del GTIDIC, disponible en su página web (<https://ja.cat/zvyK4>):

Competencias significativas:

Básicas

CB3. Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética

Transversales

CT3. Adquirir capacitación en el uso de las nuevas tecnologías y de las tecnologías de la información y la comunicación

Generales

CG1. Capacidad para concebir, planificar y desarrollar proyectos en el ámbito de las TIC

CG2. Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas informáticos.

CG4. Capacidad para emplear los métodos de la ingeniería del software en el desarrollo de aplicaciones informáticas interactivas.

CG9. Capacidad de análisis y síntesis

Específicas

CE3. Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos y bases de datos útiles para el desarrollo de aplicaciones informáticas interactivas.

CE6. Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad.

CE10. Capacidad para analizar, diseñar, construir y mantener aplicaciones digitales interactivas de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

CE13. 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 interactivas basadas en ello.

CE15. Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.

CE16. Capacidad para diseñar y evaluar interfaces persona computador que garanticen la usabilidad de los sistemas, servicios y aplicaciones informáticas.

CE25. Ser capaz de analizar, organizar, etiquetar y visualizar la estructura que define la interacción con los contenidos digitales, mediante la aplicación de métodos, técnicas y herramientas de arquitectura de información que faciliten la accesibilidad.

CE26. Saber aplicar los principios y estándares de accesibilidad y diseño universal de los principales productos y servicios digitales para diseñar experiencias que garanticen la igualdad de oportunidades entre sus usuarios.

Contenidos fundamentales de la asignatura

- Conceptos básicos de Ingeniería de Software
- Análisis de requisitos y especificación
- Análisis del dominio
- Introducción a las pruebas de software
- Principios de diseño de software

Ejes metodológicos de la asignatura

Metodología	Teoría	Práctica	Trabajo autónomo
Clases magistrales participativas	X		
Resolución de prácticas en el laboratorio (proyecto común)		X	
Resolución autónoma de prácticas (proyecto común)			X
Estudio			X

Proyecto común

Las prácticas se realizan dentro del contexto de un proyecto común. Este proyecto se desarrollará en Innovación a las TIC, Experiencia de Usuario, Aplicaciones para Dispositivos Móviles, y este curso.

El proyecto tiene como objetivo animar y fomentar en los estudiantes a enfrentarse en un escenario ágil de desarrollo software, que pretende consolidar una start-up innovadora basada en una aplicación para dispositivos móviles. El proyecto también tiene como objetivo desarrollar competencias relacionadas con la organización, comunión y relaciones sociales.

Plan de desarrollo de la asignatura

Semana	Fechas	Teoría	Práctica	Observaciones
1	06 Febrero - 09 Febrero	Presentación T1. Intro. Ing. Software	P1. Aspectos introductorios	
2	13 Febrero - 16 Febrero	T2. Intro. Ing. Requisitos	P1 (cont)	Entrega P1 hasta 19 Feb 23.55
3	20 Febrero - 23 Febrero	T3. Modelaje	P2. Requisitos iniciales	
4	27 Febrero - 2 Marzo	T4. Modelos de proceso	P2 (cont.)	
5	6 Marzo - 09 Marzo	T5. Principios de diseño software	P3. Análisis del dominio	Entrega P2 hasta 12 Marzo 23.55
6	13 Marzo - 16 Marzo	Simulacro examen	P3 (cont.)	
7	20 Marzo - 23 Marzo	Corrección simulacro P3 (cont)	FESTIVO	Entrega P3 hasta 26 Marzo 23.55
8	27 Marzo - 31 Marzo	PARCIALES	PARCIALES	
9	10 Abril - 13 Abril	SEMANA SANTA	P4 .Sprint I	
10	17 Abril - 20 Abril	T6. Gestión y planificación P4 (cont)	P4 (cont)	
11	24 Abril - 27 Abril	T7. Calidad	FESTIVO	Entrega P4 hasta 30 Abril 23.55
12	01 Mayo - 04 Mayo	FESTIVO	P5. Sprint II	
13	8 Mayo - 11 Mayo	P5 (cont.)	P5 (cont)	Entrega P5 hasta 14 Mayo 23.55
14	15 Mayo - 18 Mayo	P6. Sprint III	P6 (cont.)	

15	22 Mayo - 25 Mayo	P6 (cont.)	P6 (cont.)	Entrega P6 hasta 28 Mayo 23.55
16	29 Mayo - 30 Junio	EXÁMENES + RECUPERACIONES + TUTORIAS		

El plan de desarrollo del proyecto común se encuentra en el enunciado del proyecto en el Campus Virtual.

Sistema de evaluación

Los instrumentos de evaluación y su relación con los objetivos de aprendizaje y competencias específicas son:

Instrumento	Objetivos aprendizaje	Competencias específicas
Proyecto común	Todos	6-10-15-16
Pruebas escritas	1-2-3-6-8-9	6-10

El marco de evaluación está definido por el plan de estudios del GTIDIC, el marco académico de grados de la EPS (<https://www.eps.udl.cat/ca/informacio-academica/normatives/marc-academic-eps/>), y la normativa UdL de evaluación y calificación de la docencia en grados (<http://www.udl.cat/ca/udl/norma/>). La evaluación de la asignatura es:

Evaluación continua
<p>Nota Final (NF) = Nota_de_Prácticas (NP) * 0.5 + Nota_Teoría (NT) * 0.5, NF >= 5</p> <ul style="list-style-type: none"> • NT (50% de la NF) >= 5 <ul style="list-style-type: none"> ◦ (30% de la NF) <u>Primer parcial</u>. 2h. Sin apuntes. En papel <ul style="list-style-type: none"> ▪ Observación: Nota primer parcial = MAX (Simulacro_1, primer parcial) ◦ (20% de la NF) <u>Segundo parcial</u>. 2h. Sin apuntes. En papel. NO hay simulacro • NP (50% de la NF) >= 5 <ul style="list-style-type: none"> ◦ (30% de la NF) Prácticas <ul style="list-style-type: none"> ▪ (P1 + P2 + P3 + P4 + P5 + P6) / 6 ◦ (20% de la NF) Proyecto común <ul style="list-style-type: none"> ▪ <u>Presentación</u> (10%) ▪ <u>Seguimiento y gestión</u> (10%)
<p>Recuperaciones - nota mínima = 5, nota máxima = 7.5</p> <ul style="list-style-type: none"> • No es para subir nota • Si NT < 5: <u>Examen escrito final</u>. 2 horas de duración máx. Sin apuntes. • Si NP < 5: <u>Entrega de todas las prácticas</u>

Proyecto común

Tiene un peso de 20% de la NF. Esta nota es común a las 4 asignaturas en la que se realiza y se calcula mediante la valoración por parte del profesorado, de los compañeros, de las aportaciones que realiza cada estudiante en el desarrollo del proyecto, de la constancia y organización en el desarrollo de las tareas. Se divide en dos partes:

- **Presentación.** Defensa pública frente a un jurado en un formato parecido a una ronda de inversión. Tiene un peso del 10% de la NF.

- **Seguimiento y Gestión.** En este apartado se realizará, por parte del profesorado, un seguimiento intensivo en reuniones de seguimiento. Tiene un peso del 10% de la NF.

Al final del curso, los estudiantes recibirán un informe resumen de los objetivos, valoraciones del profesorado, del

miembro del jurado, y de los compañeros. También se incluirían datos descriptivos para ver la evolución del equipo.

Nota_1: Se puede realizar el proyecto común sin estar matriculado en las 4 asignaturas. Los estudiantes que no están matriculados en las 4 asignaturas tendrán la misma evaluación, con la excepción de que serán forzados a liderar durante la realización del proyecto y la presentación final la parte de las asignaturas en la que han estado matriculados. Este aspecto también simula una realidad cada vez más significativa como la introducción de la figura del freelance.

Nota_2: Si un grupo se divide, se mantiene la nota hasta la última entrega evaluada y las dos partes vuelan solas partiendo de la base común.

Bibliografía y recursos de información

[Applying UML and patterns : an introduction to object-oriented analysis and design and iterative development / Craig Larman](#)

[Requirements engineering / Jeremy Dick, Elizabeth Hull, Ken Jackson](#)

[Software engineering / Ian Sommerville](#)

[Software quality assurance : from theory to implementation / Daniel Galin](#)