

Guía docente de la asignatura

**Fundamentos de Ingeniería del Software**

Fecha última actualización: 14/06/2021

Fecha de aprobación: 14/06/2021

<b>Grado</b>	Grado en Ingeniería Informática (Ceuta)	<b>Rama</b>	Ingeniería y Arquitectura				
<b>Módulo</b>	Formación Específica de Rama	<b>Materia</b>	Programación e Ingeniería del Software				
<b>Curso</b>	2º	<b>Semestre</b>	2º	<b>Créditos</b>	6	<b>Tipo</b>	Obligatoria

**PRERREQUISITOS Y/O RECOMENDACIONES**

Los alumnos no tendrán que tener asignaturas, materias o módulos aprobados como requisito indispensable para cursar el módulo. No obstante se recomienda la superación de los contenidos y adquisición de competencias de las materias de formación básica, teniendo especial importancia la superación de las materias de “Fundamentos de Programación”, “Metodología de la Programación” y Programación y Diseño Orientado a Objetos.

**BREVE DESCRIPCIÓN DE CONTENIDOS (Según memoria de verificación del Grado)**

Concepto de Ingeniería del Software. El producto Software, propiedades y ciclo de vida. El proceso de desarrollo. Ingeniería de requisitos. Diseño e implementación de software. Planificación y gestión de proyectos. Validación y verificación de software. Mantenimiento de software.

**COMPETENCIAS ASOCIADAS A MATERIA/ASIGNATURA****COMPETENCIAS GENERALES**

- CG01 - 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.
- CG02 - Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática de acuerdo con los conocimientos adquiridos.
- CG04 - Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.
- CG05 - 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.



- CG06 - Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes.
- CG09 - 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.
- CG10 - Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática.
- CG12 - 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, de acuerdo con los conocimientos adquiridos.

### COMPETENCIAS ESPECÍFICAS

- CE07 - 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.
- CE08 - 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.
- CE09 - 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.
- CE10 - Capacidad para elaborar el pliego de condiciones técnicas de una instalación informática que cumpla los estándares y normativas vigentes.
- CE11 - Conocimiento, administración y mantenimiento sistemas, servicios y aplicaciones informáticas.
- 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.

### RESULTADOS DE APRENDIZAJE (Objetivos)

- Conocer el origen y significado del término “Ingeniería del Software”, su evolución histórica y los desafíos actuales, y ser consciente de la responsabilidad ética y profesional de un ingeniero de software.
- Comprender el concepto de sistema de software, sus propiedades y sus relaciones con el tipo de organización para la que se desarrolla.
- Conocer el concepto de ciclo de vida y algunos de sus tipos, siendo capaz de diferenciar entre modelos secuenciales y modelos iterativos, incrementales y evolutivos.
- Comprender el proceso de desarrollo de software y conocer las principales actividades realizadas y los productos obtenidos en el mismo.
- Comprender la necesidad del modelado, la abstracción y la transformación en el desarrollo de software.
- Conocer los principales métodos de desarrollo de software y aprender a utilizar alguno de ellos.
- Aprender técnicas concretas de especificación de requisitos y comprender la importancia



de los requisitos como base del desarrollo de software.

- Conocer los principios de diseño y la importancia de la arquitectura del software y saber aplicarlos en casos concretos.
- Conocer diferentes actividades de transformación entre diseño e implementación y saber aplicar algunas de ellas.
- Diferenciar entre planificación y gestión de proyectos, conociendo las características principales a tratar en cada caso, y aprender a realizar la planificación inicial de supuestos prácticos.
- Distinguir entre validación y verificación de software, conocer su relación con el proceso de desarrollo y su importancia en la garantía de calidad del software.
- Comprender las principales dificultades del mantenimiento de software, conocer una guía de medidas a seguir para facilitar el mantenimiento y hacer mantenimiento de software ajeno.

## PROGRAMA DE CONTENIDOS TEÓRICOS Y PRÁCTICOS

### TEÓRICO

Tema 1. Introducción a la Ingeniería del Software.

1. El producto Software, propiedades y ciclo de vida.
2. Concepto de Ingeniería del Software.
3. El proceso de desarrollo de software.

Tema 2. Ingeniería de requisitos.

1. Introducción a la ingeniería de requisitos.
2. Obtención de requisitos.
3. Modelado de casos de uso.
4. Especificación y análisis.

Tema 3. Diseño del software.

1. Conceptos y principios de diseño.
2. Diseño de los casos de uso.
3. Diseño de la estructura de objetos.
4. Arquitectura del software.

Tema 4. Otros aspectos de la Ingeniería del Software.

1. Planificación y gestión de proyectos software.
2. Validación y verificación de software.

Mantenimiento de software.

### PRÁCTICO

Seminarios/Talleres

Seminario práctico 1: Introducción a las herramientas y técnicas de ingeniería de requisitos a usar en prácticas.

Seminario práctico 2: Introducción a las herramientas y técnicas de diseño e implementación a



usar en prácticas.

### Prácticas de Laboratorio

Partiendo de la especificación informal de un pequeño sistema software, se abordarán progresivamente diferentes fases del proceso de desarrollo y mantenimiento de software, así como la necesaria planificación del proyecto, de forma coordinada con la teoría.

Prácticas con pesos en % en la calificación de prácticas:

1. Introducción a Herramienta CASE. (Voluntaria puede añadir hasta un 5% a la calificación)
2. Ingeniería de requisitos: Lista inicial de requisitos. (10 %)
3. Ingeniería de requisitos: Modelo de casos de uso. (30%)
4. Ingeniería de requisitos: Análisis y especificación de requisitos. (30%)
5. Diseño. (30%)

## BIBLIOGRAFÍA

### BIBLIOGRAFÍA FUNDAMENTAL

- I. Sommerville. Software engineering. Addison Wesley, 2011.
- J. Arlow, I Neustad. UML 2. ANAYA Multimedia, 2006.
- R. Pressman. Ingeniería del Software. McGraw Hill, 2013.
- S. L. Pfleeger. Ingeniería de Software: teoría y práctica. Prentice Hall, 2002.
- S. Sanchez, M. Sicilia, D. Rodriguez, Ingeniería del Software. Un enfoque desde la guía SWEBOK. Garceta. 2011
- C. Larman. UML y Patrones. Introducción al analisis y al diseño orientado a objetos. Prentice Hall, 2003.

### BIBLIOGRAFÍA COMPLEMENTARIA

- G. Booch, J. Rumbaugh, I. Jacobson. El Lenguaje Unificado de Modelado. Guía de Usuario. Pearson Educación, 2006.
- B. Bruegge, A. H. Dutoit. Object-Oriented Software Engineering. Using UML, Patterns, and Java. Pearson Educación, 2004.
- T. C. Lethbridge, R. Laganière. Object-Oriented Software Engineering. Practical Software Development using UML and Java. Mc Graw Hill, 2005.

## ENLACES RECOMENDADOS

Definición del estándar [Unified Modeling Language](#)  
[Recursos relacionados con la ingeniería del Software del libro de R. Pressman](#)  
Guide to the Software Engineering Body of Knowledge ([SWEBOK](#))

## METODOLOGÍA DOCENTE



- MD01 Lección Magistral (Clases Teóricas-Expositivas)
- MD02 Actividades Prácticas (Resolución de Problemas, Resolución de Casos Prácticos, Desarrollo de Proyectos, Prácticas en Laboratorio, Taller de Programación, Aula de Informática, Prácticas de Campo).
- MD03 Seminarios (Debates, Demos, Exposición de Trabajos Tutelados, Conferencias, Visitas Guiadas, Monografías).
- MD04 Actividades no presenciales Individuales.
- MD05 Actividades no presenciales Grupales.
- MD06 Tutorías Académicas.

## EVALUACIÓN (instrumentos de evaluación, criterios de evaluación y porcentaje sobre la calificación final)

### EVALUACIÓN ORDINARIA

El sistema que se seguirá para la evaluación de esta asignatura es bajo la modalidad de evaluación continua, tal como se describe a continuación.

El sistema de calificación establece que el 100% de la nota final se reparte en: 50% de teoría y 50% prácticas. Para aprobar la asignatura es necesario tener una calificación numérica superior o igual a 5 puntos (sobre 10). La suma de las calificaciones de ambas partes sólo se realizará en caso de tener en cada parte, una puntuación superior o igual a 5 puntos (sobre 10). La calificación final es la suma de las calificaciones obtenidas en teoría y prácticas. Si sólo se supera una parte (teoría o prácticas), la calificación de la parte superada se mantiene hasta la siguiente convocatoria extraordinaria del curso académico actual.

Con respecto a la calificación detallada de cada una de las partes, se establece lo siguiente:

- Los 5 puntos de la teoría se reparten de la siguiente forma:
- 30% corresponden a pruebas objetivas individuales realizadas en el aula. 30% correspondiente a trabajo individual o en grupo para la resolución de ejercicios, problemas, resolución de cuestionarios online previos a las clases y/o trabajos, así como la participación en clase. 40% correspondiente a un examen final.
- Los 5 puntos de prácticas se obtienen de la evaluación ponderada de cada una de las prácticas realizadas a lo largo del curso.

Todo lo relativo a la evaluación se regirá por la [Normativa de evaluación y calificación de los estudiantes](#) vigente en la Universidad de Granada.

El sistema de calificaciones se expresará mediante calificación numérica de acuerdo con lo establecido en el art. 5 del R. D 1125/2003, de 5 de septiembre, por el que se establece el sistema europeo de créditos y el sistema de calificaciones en las titulaciones universitarias de carácter oficial y validez en el territorio nacional.

### EVALUACIÓN EXTRAORDINARIA

El método de evaluación en las convocatorias extraordinarias constará de un examen, incluyendo tanto los aspectos teóricos como prácticos impartidos a lo largo del curso académico.

### EVALUACIÓN ÚNICA FINAL



Se realizará una evaluación única final para aquellos estudiantes que no puedan acogerse a la evaluación continua y hayan solicitado dicho examen único final, alegando y acreditando las razones que le asisten para no poder seguir el sistema de evaluación continua; siempre de acuerdo a lo que se especifica en el Artículo 8 de la “Normativa de Evaluación y Calificación de los Estudiantes de la Universidad de Granada”.

La evaluación única final constará de un examen, incluyendo tanto los aspectos teóricos como prácticos impartidos a lo largo del curso académico.

### INFORMACIÓN ADICIONAL

El título de Graduado/a en Ingeniería Informática de la Universidad de Granada ha obtenido, con fecha 5 de junio de 2019, el sello Euro-Inf, otorgado por ANECA en colaboración con el Consejo General de Colegios Profesionales de Ingeniería en Informática (CCII) y con el Consejo General de Colegios Oficiales de Ingeniería Técnica en Informática (CONCITI). Esta acreditación garantiza el cumplimiento de criterios y estándares reconocidos por los empleadores españoles y del resto de Europa, de acuerdo con los principios de calidad, relevancia, transparencia, reconocimiento y movilidad contemplados en el Espacio Europeo de Educación Superior.

#### RÉGIMEN DE ASISTENCIA A CLASE:

- La asistencia a clase de teoría no será obligatoria, aunque la participación activa en clase y la entrega de ejercicios planteados por el profesor se tendrá en cuenta dentro del sistema de evaluación continua de la asignatura.
- La asistencia a las clases prácticas será obligatoria. Sólo se podrá faltar a tres clases de prácticas.

