

**Proyecto/Guía docente de la asignatura**

Asignatura	INGENIERÍA DE SISTEMAS SOFTWARE		
Materia	FUNDAMENTOS DE SISTEMAS SOFTWARE		
Módulo	MATERIAS BÁSICAS DE TELECOMUNICACIONES		
Titulación	GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN GRADO EN INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS DE TELECOMUNICACIÓN		
Plan	460 (I.T.T.) 512 (I.T.E.T.)	Código	45015 (I.T.T.) 46618 (I.T.E.T.)
Periodo de impartición	2º CUATRIMESTRE	Tipo/Carácter	FORMACIÓN OBLIGATORIA
Nivel/Ciclo	GRADO	Curso	2º
Créditos ECTS	6 ECTS		
Lengua en que se imparte	CASTELLANO		
Profesor/es responsable/s	GUILLERMO VEGA GORGOJO MARÍA ÁNGELES PÉREZ JUÁREZ		
Datos de contacto (E-mail, teléfono...)	TELÉFONO: 983 423000 ext. 5538 / ext. 3709 E-MAIL: guiveg@tel.uva.es, mperez@tel.uva.es		
Departamento	TEORÍA DE LA SEÑAL Y COMUNICACIONES E INGENIERÍA TELEMÁTICA		
Fecha de revisión por el Comité de Título	27/6/2025		

Esta asignatura forma parte de dos planes en extinción. De acuerdo con la memoria de verificación del plan de estudios “727 – Grado en Ingeniería de Tecnologías de Telecomunicación”, para las asignaturas de los planes de estudio a extinguir los alumnos tendrán derecho a 4 convocatorias de examen (dos en el primer curso en el que queda suprimida la docencia y dos en el curso siguiente) y a tutorías durante esos dos cursos académicos. Esta guía determina el temario y el mecanismo de evaluación de los estudiantes en esta situación.

Dicho lo anterior, el nuevo plan de estudios incluye la asignatura “Programación II” con objetivos y contenidos asimilables a esta asignatura. Así, se ofrece a los estudiantes del plan a extinguir que, sin modificar su matrícula, participen íntegramente de la docencia y la evaluación de la asignatura del plan nuevo. Para ello, durante la primera semana del curso deberán expresar su voluntad en este sentido por los medios que los profesores determinen. Para los alumnos que tomen esta opción, los contenidos, métodos docentes y criterios de evaluación serán los expresados en la asignatura “Programación II”.

1. Situación / Sentido de la Asignatura

1.1 Contextualización

Al enfrentarse a la presente asignatura, puede que el alumno tenga aún la idea preconcebida de que hacer software es, fundamentalmente, elaborar código. Un planteamiento inicial de ese tipo suele estar destinado al fracaso en proyectos de cierta envergadura – las recomendaciones de distribución del esfuerzo en el desarrollo del código de un proyecto software suelen ser de un 15-20 % del esfuerzo global.

Tras una profunda crisis en la industria del software en los años 70 y 80, acompañada de un sustancial incremento en la complejidad del software a desarrollar, se hizo evidente la necesidad de disponer de nuevas técnicas que permitieran el paso de la producción de software de manera artesanal a un proceso de ingeniería. Así, la IEEE define la Ingeniería de Software como la aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software. Por tanto, la Ingeniería de Software es la disciplina que se ocupa del software, enfrentándose al mismo como un producto de ingeniería que requiere planificación, análisis, diseño, implementación, pruebas y mantenimiento.

Desarrollar software es, por tanto, mucho más que escribir código, requiriendo un esfuerzo previo para diseñarlo. Por este motivo, en Ingeniería de Sistemas Software el énfasis se pone no tanto en el producto final (el software como un producto), como en su proceso de desarrollo (el software como un proceso). El proceso del software define el enfoque que se aplica cuando el software es tratado utilizando una aproximación ingenieril tal y como hace la ingeniería de software. El proceso que se seguirá en la asignatura será uno de los más extendidos, el denominado Proceso Unificado (*Unified Process* – UP) de desarrollo de software. Además, se empleará como notación el Lenguaje de Modelado Unificado (*Unified Modelling Language* – UML), el lenguaje de facto para documentar proyectos software. En cuanto a las herramientas, se empleará una herramienta CASE (*Computer Aided Software Engineering*) que permita elaborar diagramas en UML, así como un entorno integrado de desarrollo (*Integrated Development Environment* – IDE) para el lenguaje de programación Java.

El Proceso Unificado de Desarrollo de Software y el Lenguaje de Modelado Unificado serán el hilo conductor de buena parte de las sesiones teóricas y prácticas, ya que en muchas de ellas se abordará algún aspecto concreto como alguna de las Fases y/o Actividades y/o Entregables típicos del Proceso Unificado – en muchos casos correspondientes a un diagrama UML. La estructura secuencial de exposición de contenidos se planteará para permitir abordar gradualmente la realización de un proyecto software de forma completa siguiendo las pautas de la Ingeniería de Software y haciendo uso de la orientación a objetos como paradigma de programación subyacente en el planteamiento propuesto para el proyecto práctico. La orientación a objetos es el paradigma de programación dominante en el panorama actual del desarrollo del software. En la asignatura se hará especial énfasis en la aplicación de patrones software para realizar un diseño orientado a objetos efectivo.

Además, y puesto que la creación de un producto software de cierta relevancia es siempre una tarea en equipo, se trabajará bajo dicha perspectiva. Así, la asignatura está diseñada siguiendo el método de aprendizaje por proyectos, de modo que los alumnos trabajarán en grupo para llevar a cabo un proyecto de un sistema software. Se trata de sentar las bases para abordar proyectos complejos de desarrollo software en equipo, lo cual podrá ser útil al alumno tanto en otras asignaturas de la titulación como en su Trabajo Fin de Grado o en su carrera profesional posterior. En este sentido no debe perderse de vista que el ingeniero de software es un perfil profesional ampliamente demandado por el mercado laboral tanto en España como fuera de nuestro país y que

la industria del software es una industria potente en la que conviven grandes gigantes como Microsoft junto con numerosas microempresas de menos de 10 trabajadores.

1.2 Relación con otras materias

Esta asignatura se apoya en la asignatura *Programación* de la materia *Informática* del *Bloque de Materias Instrumentales* que se imparte en el 1er cuatrimestre del 1er curso.

Por otra parte, dado que *Ingeniería de Sistemas Software* sienta las bases de la ingeniería de software y trata de llevar a cabo un proyecto de un sistema software, se recomienda fuertemente haberla cursado antes de abordar las asignaturas posteriores de los planes de estudios que están relacionadas con la construcción de software – especialmente si se sigue el paradigma de programación orientada a objetos y se utiliza el lenguaje de programación Java, tratados en *Ingeniería de Sistemas Software*. A continuación se describen brevemente las asignaturas posteriores con las que se relaciona:

- *Desarrollo de Aplicaciones Distribuidas* es una asignatura obligatoria de tercer curso de la materia *Ingeniería de Redes, Sistemas y Servicios Telemáticos* dentro del bloque de *Materias Específicas de Tecnologías de Telecomunicación* del *Grado en Ingeniería de Tecnologías de Telecomunicación*. Esta asignatura se centra en el estudio de los paradigmas, lenguajes y *middleware* para el desarrollo de aplicaciones distribuidas, utilizando el lenguaje de programación Java.
- *Laboratorio de Desarrollo de Sistemas Telemáticos* es una asignatura optativa de cuarto curso de la materia *Ingeniería de Redes, Sistemas y Servicios Telemáticos* dentro del bloque de *Materias Específicas de Tecnologías de Telecomunicación* del *Grado en Ingeniería de Tecnologías de Telecomunicación*. Esta asignatura se centra en el estudio de aplicaciones web utilizando los lenguajes de programación JavaScript y PHP.
- *Arquitecturas de Aplicaciones Distribuidas* es una asignatura obligatoria de tercer curso de la materia *Arquitecturas y Tecnologías para Aplicaciones Distribuidas* dentro del bloque de *Materias Específicas* del *Grado en Ingeniería de Tecnologías Específicas de Telecomunicación con Mención en Telemática*. Esta asignatura se centra en el estudio de los paradigmas, lenguajes y *middleware* para el desarrollo de aplicaciones distribuidas. Se aplica el paradigma de programación orientada a objetos (además de la orientación a componentes software y a servicios) y se utiliza el lenguaje de programación Java.
- *Tecnologías para Aplicaciones Web* es una asignatura obligatoria de cuarto curso de la materia *Arquitecturas y Tecnologías para Aplicaciones Distribuidas* dentro del bloque de *Materias Específicas* del *Grado en Ingeniería de Tecnologías Específicas de Telecomunicación con Mención en Telemática*. Esta asignatura se centra en el estudio de aplicaciones web utilizando los lenguajes de programación JavaScript y PHP.
- *Desarrollo de Aplicaciones para Dispositivos Móviles* es una asignatura optativa de cuarto curso de la materia *Dominios Específicos de Aplicación* dentro del bloque de *Materias Específicas* del *Grado en Ingeniería de Tecnologías Específicas de Telecomunicación con Mención en Telemática*. Esta asignatura se centra en el estudio de aplicaciones móviles para plataformas Android e iOS y en ella se utiliza el lenguaje de programación Java.

- *Desarrollo de Aplicaciones Telemáticas* es una asignatura obligatoria de tercer curso de la materia *Desarrollo de Aplicaciones Telemáticas* dentro del bloque de *Materias Específicas* del *Grado en Ingeniería de Tecnologías Específicas de Telecomunicación con Mención en Sistemas de Telecomunicación*. Esta asignatura se centra en el estudio de los paradigmas, lenguajes y *middleware* para el desarrollo de aplicaciones distribuidas, así como en el estudio de aplicaciones web. Se aplica el paradigma de programación orientada a objetos utilizando varios lenguajes como JavaScript, PHP y Java.

Es de destacar que en esta asignatura se presentan los conceptos de Orientación a Objetos y se introduce el lenguaje de programación Java, sirviendo como base para una buena parte de las asignaturas posteriores mencionadas anteriormente.

1.3 Prerrequisitos

Aunque no existen requisitos previos para matricularse en *Ingeniería de Sistemas Software*, se recomienda haber cursado previamente la asignatura *Programación* de la materia *Informática* del bloque de *Materias Instrumentales* que se imparte en primer curso de las titulaciones de grado. En dicha asignatura se introducen conceptos básicos de programación y algorítmica utilizando el lenguaje de programación C.

2. Competencias

2.1 Generales

- GB1. Capacidad de razonamiento, análisis y síntesis.
- GB2. Capacidad para relacionar conceptos y adquirir una visión integrada, evitando enfoques fragmentarios.
- GB3. Capacidad de toma de decisiones en la resolución de problemas básicos de ingeniería de telecomunicación, así como identificación y formulación de los mismos.
- GB4. Capacidad para trabajar en grupo, participando de forma activa, colaborando con sus compañeros y trabajando de forma orientada al resultado conjunto, y en un entorno multilingüe.
- GB5. Conocimiento de materias básicas, científicas y tecnologías, que le capacite para el aprendizaje de nuevos métodos y tecnologías.
- GB6. Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos en el ámbito de la ingeniería técnica de Telecomunicación.
- GBE1. Capacidad para manejar especificaciones, reglamentos y normas de obligado cumplimiento.
- GBE3. Capacidad para resolver problemas con iniciativa, creatividad y razonamiento crítico.
- GC1. Capacidad de organización, planificación y gestión del tiempo.
- GC2. Capacidad para comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas relacionadas con las telecomunicaciones y la electrónica.
- GC3. Capacidad para trabajar en cualquier contexto, individual o en grupo, de aprendizaje o profesional, local o internacional, desde el respeto a los derechos fundamentales, de igualdad de sexo, raza o religión y los principios de accesibilidad universal, así como la cultura de paz.

2.2 Específicas

- T1. Capacidad para aprender de manera autónoma nuevos conocimientos y técnicas adecuados para la concepción, el desarrollo o la explotación de sistemas y servicios de telecomunicación.
- T2. Capacidad de utilizar aplicaciones informáticas (ofimáticas, bases de datos, cálculo avanzado, gestión de proyectos, visualización, etc.) para apoyar el desarrollo y explotación de redes, servicios y aplicaciones de telecomunicación y electrónica.
- T6. Capacidad de concebir, desplegar, organizar y gestionar redes, sistemas, servicios e infraestructuras de telecomunicación en contextos residenciales (hogar, ciudad y comunidades digitales), empresariales o institucionales responsabilizándose de su puesta en marcha y mejora continua, así como conocer su impacto económico y social.
- T7. Conocimiento y utilización de los fundamentos de la programación en redes, sistemas y servicios de telecomunicación.

3. Objetivos

El objetivo general de la asignatura puede formularse de la siguiente manera:

OG Comprender los principios, conceptos y métodos de ingeniería de software, y en particular del Proceso Unificado, para organizar, diseñar, desarrollar y documentar un proyecto de un sistema software orientado a objetos de tamaño mediano.

Tras definir el objetivo general de *Ingeniería de Sistemas Software*, se especifican los siguientes objetivos específicos:

OE1 Aplicar la metodología del Proceso Unificado para planificar y organizar proyectos de software.

OE2 Analizar y formalizar los requisitos de software de un proyecto.

OE3 Realizar el análisis y el diseño orientado a objetos de un sistema software.

OE4 Aplicar patrones de software como instrumento principal en el diseño orientado a objetos.

OE5 Programar en el lenguaje Java un sistema software a partir de un diseño software orientado a objetos.

OE6 Documentar un sistema software utilizando el lenguaje UML como notación fundamental.

OE7 Trabajar en equipo de manera efectiva para la construcción de un sistema software.

4. Contenidos y/o bloques temáticos



Bloque 1: Ingeniería de Sistemas Software

Carga de trabajo en créditos ECTS:

6

a. Contextualización y justificación

Véase la contextualización y justificación de la asignatura.

b. Objetivos de aprendizaje

Véanse los objetivos de la asignatura.

c. Contenidos

TEMA 1: Introducción a la Ingeniería de Software

- 1.1 Conceptos generales
- 1.2 El proceso del software
- 1.3 El Proceso Unificado de Desarrollo de Software
- 1.4 El Lenguaje de Modelado Unificado (UML)
- 1.5 Herramientas CASE

TEMA 2: Visión, Requisitos y Glosario

- 2.1 Visión: Descripción inicial del Sistema y Prototipo de la Interfaz de Usuario
- 2.2 Especificación de Requisitos
- 2.3 Glosario del proyecto

TEMA 3: Los Casos de Uso

- 3.1 Motivación de los Casos de Uso
- 3.2 Notación en UML de los Casos de Uso
- 3.3 Buenas prácticas en la elaboración de los Casos de Uso

TEMA 4: Modelos de Dominio

- 4.1 Introducción a la Orientación a Objetos
- 4.2 Notación en UML de los Diagramas de Clases
- 4.3 Motivación de los Modelos de Dominio
- 4.4 Buenas prácticas en la elaboración de los Modelos de Dominio

TEMA 5: Los Diagramas de Secuencia del Sistema y los Contratos

- 5.1 Diagramas de Secuencia del Sistema
- 5.2 Notación en UML de los Diagramas de Secuencia del Sistema
- 5.3 Buenas prácticas en la elaboración de los Diagramas de Secuencia del Sistema
- 5.4 Contratos de las Operaciones del Sistema
- 5.5 Buenas prácticas en la elaboración de los Contratos de las Operaciones del Sistema

TEMA 6: Los Diagramas de Interacción

- 6.1 Diagramas de Interacción

6.2 Notación en UML de los Diagramas de Interacción

6.3 Buenas prácticas en la elaboración de los Diagramas de Interacción

TEMA 7: El Diseño

7.1 Consideraciones en el Diseño

7.2 Patrones de Diseño

TEMA 8: La Implementación

8.1 Consideraciones en la Implementación

8.2 Introducción al lenguaje de programación Java

TEMA 9: Aspectos complementarios de Ingeniería de Software

9.1 Introducción a los Métodos Ágiles

9.2 Otros aspectos de la Ingeniería del Software

Nota: Los contenidos de este tema pueden adaptarse dinámicamente en función del tiempo disponible y de la posibilidad de contar con expertos de empresas de software que puedan compartir su experiencia.

d. Métodos docentes

Este curso esta asignatura es sin docencia. Los alumnos pueden utilizar los materiales proporcionados en el último curso con docencia, así como asistir a tutorías concertadas con los profesores.

e. Plan de trabajo

No procede en una situación sin docencia.

f. Evaluación

La evaluación de la adquisición de competencias se llevará a cabo con un examen final de problemas.

g. Material docente

g.1 Bibliografía básica

- Documentación básica de los diferentes temas de la asignatura proporcionada por los profesores de la asignatura.
- Otros documentos complementarios también proporcionados por los profesores de la asignatura: enunciados de ejercicios y prácticas, etc.
- C. Larman, *Applying UML and Patterns*, Prentice Hall, 3ª edición, Upper Saddle River, NJ, USA, 2005.

Puede accederse a la [plataforma Leganto](#) para comprobar la disponibilidad de la bibliografía básica.

g.2 Bibliografía complementaria

- Ariadne Training, *UML Applied. Object Oriented Analysis and Design*, 2nd ed. Ariadne Training, 2005.

- Mike Cohn. Succeeding with agile: software development using Scrum. Addison-Wesley, Boston, MA, EEUU, 2009.
- Benjamin J. Evans y David Flanagan. Java in a Nutshell (sexta edición). O'Reilly, Sebastopol, CA, EEUU, 2014.
- Martin Fowler. UML distilled: a brief guide to the standard object modeling language (tercera edición). Addison-Wesley, Boston, MA, EEUU, 2003.
- Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Design patterns: elements of reusable object-oriented software. Prentice Hall, Upper Saddle River, NJ, EEUU, 1995.
- Robert Nystrom. Game programming patterns. Genever Benning, Exeter, Reino Unido, 2014.
- Roger S. Pressman, *Software engineering: a practitioner's approach (octava edición)*. McGraw-Hill, Columbus, OH, EEUU, 2014.
- J. Rumbaugh, I. Jacobson, G. Booch, *El Proceso Unificado de Desarrollo de Software*, Addison Wesley, 1999.

Puede accederse a la [plataforma Leganto](#) para comprobar la disponibilidad de la bibliografía básica.

g.3 Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), ...)

- Materiales de la asignatura preparados por los profesores y publicados en el Campus Virtual.
- Tutoriales de Java ofrecidos por Oracle (<https://docs.oracle.com/javase/tutorial/java/>).
- MOOC Introduction to Java Programming: Starting to code in Java de la Universidad Carlos III (<https://www.edx.org/course/introduction-to-java-programming-starting-to-code>).
- Foro de preguntas para programadores StackOverflow (<https://stackoverflow.com/>).
- Web con libros escritos por miembros de StackOverflow sobre temas relacionados con la asignatura: Java, Git, algoritmos..., (<https://books.goalkicker.com/>).

Debe también tenerse en cuenta que, sobre los temas tratados en la asignatura, existen multitud de libros y manuales, a mayores de los recomendados, algunos de los cuales están disponibles para su consulta en la biblioteca del Campus. Estos materiales pueden proporcionar explicaciones alternativas (que quizá le resulten más claras al alumno) y más ejemplos. También es posible encontrar tutoriales, cursos, artículos, etc., sobre los temas abordados en la asignatura en Internet. En particular, en relación con el lenguaje de programación Java, se recomienda a los alumnos consultar la documentación oficial proporcionada por Oracle, consistente en tutoriales y una descripción detallada de la API estándar.

h. Recursos necesarios

Serán necesarios los siguientes recursos, todos ellos facilitados por el Centro y/o por los profesores de la asignatura:

- Entorno de trabajo en la plataforma Moodle ubicada en el Campus Virtual de la Universidad de Valladolid u otra plataforma virtual alternativa.
- Documentación de apoyo.
- Acceso al material bibliográfico complementario recomendado en la biblioteca del Campus o mediante otras vías como Internet.

i. Temporalización

No procede en una situación sin docencia.

5. Métodos docentes y principios metodológicos

Véase el apartado 4.d.

6. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES o PRESENCIALES A DISTANCIA ⁽¹⁾	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
		Estudio y trabajo autónomo	150
Total presencial		Total no presencial	150
TOTAL presencial + no presencial			150

(1) Actividad presencial a distancia es cuando un grupo sigue una videoconferencia de forma síncrona a la clase impartida por el profesor.

7. Sistema y características de la evaluación

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Examen de problemas	100%	Es condición necesaria para superar la asignatura alcanzar una calificación igual o superior a 5 puntos sobre 10 en el examen.

CRITERIOS DE CALIFICACIÓN

Convocatoria ordinaria:

La calificación de la convocatoria será íntegramente la calificación obtenida en el examen.

Convocatoria extraordinaria(*):

La calificación de la convocatoria será íntegramente la calificación obtenida en el examen.

(*) Se entiende por convocatoria extraordinaria la segunda convocatoria.

8. Consideraciones finales