

**Guía docente de la asignatura**

<b>Asignatura</b>	INGENIERÍA DE SISTEMAS SOFTWARE		
<b>Materia</b>	FUNDAMENTOS DE SISTEMAS SOFTWARE		
<b>Módulo</b>	MATERIAS BÁSICAS DE TELECOMUNICACIONES		
<b>Titulación</b>	GRADO EN INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS DE TELECOMUNICACIÓN		
<b>Plan</b>	460 (I.T.T.)	<b>Código</b>	45015 (I.T.T.)
	512 (I.T.E.T.)		46618 (I.T.E.T.)
<b>Periodo de impartición</b>	2º CUATRIMESTRE	<b>Tipo/Carácter</b>	FORMACIÓN OBLIGATORIA
<b>Nivel/Ciclo</b>	GRADO	<b>Curso</b>	2º
<b>Créditos ECTS</b>	6 ECTS		
<b>Lengua en que se imparte</b>	CASTELLANO		
<b>Profesor/es responsable/s</b>	EDUARDO GÓMEZ SÁNCHEZ MARÍA ÁNGELES PÉREZ JUÁREZ		
<b>Datos de contacto (E-mail, teléfono...)</b>	TELÉFONO: 983 423000 ext. 5532 / ext. 3709 E-MAIL: <a href="mailto:edugom@tel.uva.es">edugom@tel.uva.es</a> , <a href="mailto:mperez@tel.uva.es">mperez@tel.uva.es</a>		
<b>Horario de tutorías</b>	Véase <a href="http://www.uva.es">www.uva.es</a> → Centros → Campus de Valladolid → Escuela Técnica Superior de Ingenieros de Telecomunicación → Tutorías		
<b>Departamento</b>	TEORÍA DE LA SEÑAL Y COMUNICACIONES E INGENIERÍA TELEMÁTICA		

## 1. Situación / Sentido de la Asignatura

### 1.1 Contextualización

Al enfrentarse a la presente asignatura, puede que el alumno tenga aún la idea preconcebida, y errónea, de que hacer software es, fundamentalmente, elaborar código. Un planteamiento inicial de ese tipo, le llevará en cualquier proyecto de cierta envergadura, sin lugar a dudas, al fracaso, puesto que el escribir el código suele estimarse en, aproximadamente, un 10 % del proceso completo de construcción de un producto software.

Tras una profunda crisis en la industria del software, la necesidad de desarrollar software más complejo hizo que surgiera pareja la necesidad de contar con nuevas técnicas para su desarrollo, se hizo necesario el paso de la artesanía a la ingeniería. Así, en 1993, la IEEE definió la Ingeniería de Software como la aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software, es decir, la aplicación de la ingeniería al 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.

Hacer software es, por tanto, mucho más que escribir código, es necesario pensar en hacer software sin sentarse a escribir código. Por este motivo, en la presente asignatura el énfasis se pondrá, 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. Pero la Ingeniería de Software necesita también de técnicas, incluyendo la notación que estas emplean, y de las herramientas vinculadas al proceso. Por tanto, los tres pilares básicos de la Ingeniería de Software y, por ello, del trabajo en esta asignatura, serán un proceso, una serie de técnicas con su notación y un conjunto de herramientas. El proceso a emplear es el denominado Proceso Unificado (UP, *Unified Process*) de Desarrollo de Software. La notación que le acompaña será el Lenguaje de Modelado Unificado (UML, *Unified Modelling Language*). Y la herramienta CASE (*Computer Aided Software Engineering*) a utilizar será alguna de las muchas disponibles que resulte adecuada para el trabajo a desarrollar en la asignatura.

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 de Desarrollo de Software o algún artefacto del Lenguaje de Modelado Unificado siendo difícil la separación de ambos componentes: Proceso Unificado y Lenguaje de Modelado Unificado, puesto que la Ingeniería de Software hace uso de ambos que interactúan y se realimentan a lo largo de todo el proceso de construcción del software. 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.

Además, y puesto que la creación de un producto software de cierta relevancia es siempre una tarea en equipo, se trabajará bajo dicha perspectiva. 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 quizás incluso en el desarrollo de su proyecto fin de carrera o en su vida profesional posterior de orientarse está al desarrollo

de sistemas software, ya que 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, y dado que la presente asignatura sienta las bases de la Ingeniería de Software, es decir, de la disciplina que se ocupa del software, entendiendo el mismo como un producto de ingeniería que requiere planificación, análisis, diseño, implementación, pruebas y mantenimiento, será recomendable haberla cursado antes de abordar las asignaturas posteriores relacionadas de alguna forma con la construcción de software, a saber:

- “Desarrollo de Aplicaciones Distribuidas” y “Laboratorio de Desarrollo de Sistemas Telemáticos” de la materia “Ingeniería de Redes, Sistemas y Servicios Telemáticos” dentro del “Bloque de Materias Específicas de Tecnologías de Telecomunicación” que se imparten en el 1er cuatrimestre del 3er curso y en el 1er cuatrimestre del 4º curso respectivamente.
- “Desarrollo de Aplicaciones Telemáticas” de la materia “Desarrollo de Aplicaciones Telemáticas” dentro del “Bloque de Materias Específicas de Sistemas de Telecomunicación” que se imparte en el 2º cuatrimestre del 3er curso.
- “Arquitecturas y Tecnologías para el Desarrollo de Aplicaciones Distribuidas” de la materia “Ingeniería de Redes, Sistemas y Servicios Telemáticos” dentro del “Bloque de Materias Específicas de Sistemas Electrónicos” que se imparte en el 2º cuatrimestre del 4º curso.
- “Arquitecturas de Aplicaciones Distribuidas” y “Tecnologías para Aplicaciones Web” de la materia “Desarrollo de Aplicaciones Telemáticas” dentro del “Bloque de Materias Específicas de Telemática” que se imparten en el 2º cuatrimestre del 3er curso y en el 1er cuatrimestre del 4º curso respectivamente.

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: “Desarrollo de Aplicaciones Distribuidas”, “Desarrollo de Aplicaciones Telemáticas”, “Arquitecturas y Tecnologías para el Desarrollo de Aplicaciones Distribuidas” y “Arquitecturas de Aplicaciones Distribuidas”.

## 1.3 Prerrequisitos

---

No existen condiciones previas excluyentes para cursar esta asignatura, aunque sí recomendaciones lógicas que el alumno debería tener en cuenta. En concreto es recomendable haber cursado con anterioridad la asignatura “Programación” de la materia “Informática” del “Bloque de Materias Instrumentales” que se imparte en el 1er cuatrimestre del 1er curso.

## 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

---

Al finalizar la asignatura el alumno deberá ser capaz de:

- Describir el Proceso Unificado de Desarrollo de Software, así como otros modelos de procesos más clásicos.
- Producir la documentación relativa a diversos entregables vinculados a las diferentes Fases y Actividades del Proceso Unificado de Desarrollo de Software, como por ejemplo el documento de Especificación de Requisitos del Software (SRS, *Software Requirements Specification*), relativos a un sistema software

determinado, con el rigor y nivel de calidad adecuados, puesto que el contenido de dichos documentos deberá ser asumido y entendido tanto por el cliente/usuario como por el ingeniero de software. Asimismo, optimizar el esfuerzo de producir dichos entregables y enfatizar su utilidad para las siguientes Fases y Actividades.

- Modelar un sistema software concreto haciendo uso del Lenguaje de Modelado Unificado como notación fundamental vinculada a la metodología utilizada en la Ingeniería del Software, a partir de la comprensión de un dominio/negocio concreto y de la captura de Requisitos de un sistema software real mediante la comunicación con un agente especialista en dicho dominio/negocio en un tiempo limitado.
- Construir diferentes artefactos UML como el Diagrama de Casos de Uso con la ayuda de herramientas informáticas de apoyo a la Ingeniería Software, es decir, con las denominadas herramientas CASE, relativos a un sistema software determinado.
- Aprender la rentabilidad de trabajar haciendo uso de la metodología propuesta por la Ingeniería de Software frente a un enfoque no ingenieril y caótico que se centre en la elaboración de código, prescindiendo de las actividades de Análisis y Diseño.
- Aprender de forma autónoma:
  - Localizar y asimilar una determinada información a partir de su referencia.
  - Buscar información técnica relevante para una tarea especificada.
- Autoevaluar el trabajo propio realizado, identificando sus errores y aspectos a mejorar.
- Trabajar en equipo:
  - Intercambiar información a través del correo electrónico, foros de discusión y otras herramientas de comunicación síncrona y/o asíncrona.
  - Explicar a un equipo el resultado de una tarea realizada y asegurarse de que todos los demás lo han comprendido.
  - Identificar adecuadamente las tareas a realizar por el equipo, repartir equitativamente las tareas, establecer plazos de entrega, e integrar el trabajo realizado por los diferentes miembros del equipo.
  - Identificar los aspectos positivos relativos al funcionamiento del equipo. Identificar y abordar los aspectos a mejorar así como los conflictos en el funcionamiento del equipo.
- Realizar informes de carácter técnico.

#### 4. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teóricas	20	Estudio y trabajo autónomo individual	45
Clases prácticas		Estudio y trabajo autónomo grupal	45
Laboratorios	40		
Prácticas externas, clínicas o de campo			
Seminarios			
Otras actividades			
<b>Total presencial</b>	<b>60</b>	<b>Total no presencial</b>	<b>90</b>

#### 5. Bloques temáticos



## Bloque 1: Ingeniería de Sistemas Software

Carga de trabajo en créditos ECTS:

### 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 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



## 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

## PRÁCTICAS

- El trabajo en el laboratorio se desarrollará en torno a un proyecto, sobre el cual habrá que ir generando diferentes entregables correspondientes a las actividades de Requisitos, Análisis, Diseño e Implementación propuestas por el Proceso Unificado de Desarrollo de Software.
- Las entregas concretas y su planificación se detallará en el Anexo I mencionado en la guía, que se facilitará al comienzo de la asignatura.

### d. Métodos docentes

---

- Clase magistral participativa.
- Estudio de casos en aula y en laboratorio.
- Aprendizaje colaborativo.
- Método de proyectos.

### e. Plan de trabajo

---

Véase el Anexo I.

### f. Evaluación

---

La evaluación de la adquisición de competencias se basará en:

- Los informes realizados por los alumnos sobre el proyecto en el que se trabaje en la asignatura.
- La prueba escrita de evaluación al finalizar la asignatura.

### g. 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., algunos de ellos extraídos de la bibliografía propuesta como complementaria.
- C. Larman, *Applying UML and Patterns*, Prentice Hall, 3ª edición, Upper Saddle River, NJ, USA, 2005.

### h. Bibliografía complementaria

---

- Ariadne Training, *UML Applied. Object Oriented Analysis and Design*, 2nd ed. Ariadne Training, 2005.
- R.S. Pressman, *Ingeniería de Software. Un Enfoque Práctico*, 5th ed. Mc Graw Hill, 2002.

- J. Rumbaugh, I. Jacobson, G. Booch, *El Proceso Unificado de Desarrollo de Software*, Addison Wesley, 1999.

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.

#### i. Recursos necesarios

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

- Aula con proyector multimedia y pizarra para las clases magistrales participativas.
- Entorno de trabajo en la plataforma Moodle ubicada en el Campus Virtual de la Universidad de Valladolid u otra plataforma virtual alternativa.
- Laboratorio de prácticas, con al menos un ordenador para cada dos alumnos, para las sesiones de laboratorio. Cada ordenador contará con una herramienta CASE que permita construir artefactos UML.
- Documentación de apoyo.
- Acceso al material bibliográfico complementario recomendado en la biblioteca del Campus o mediante otras vías como Internet.

#### 6. Temporalización (por bloques temáticos)

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1: Ingeniería de Sistemas Software	6 ECTS	Semanas 1 a 15

#### 7. Sistema de calificaciones – Tabla resumen

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Valoración por el profesor de los entregables de equipo relativos al proyecto a desarrollar.	70%	Es condición necesaria (pero no suficiente) para superar la asignatura, el alcanzar una calificación igual o superior al 50% en esta parte.  <a href="#">Los entregables se agruparán en cuatro grupos de actividades (Análisis, Diseño, Implementación y Otros), siendo necesario obtener una calificación igual o superior al 20% de la nota máxima agregada en cada grupo de actividades. Si no se realiza alguno de los entregables establecidos, la calificación final en la asignatura será de No Presentado (N.P.).</a>
Prueba escrita final.	30%	



		<p>Para superar la asignatura será necesario demostrar que se ha adquirido un grado de conocimiento suficiente en los contenidos tratados en la asignatura mediante una prueba escrita que se realizará al final del cuatrimestre.</p> <p>Es condición necesaria (pero no suficiente) para superar la asignatura alcanzar una calificación igual o superior al 50% de esta parte para que se haga la media ponderada con las otras calificaciones obtenidas en la asignatura.</p> <p>En caso de no realizarse esta prueba escrita la calificación final en la asignatura será de No Presentado (N.P.).</p>
--	--	--

Además deberá tenerse en cuenta que:

- La nota final para superar la asignatura en la convocatoria ordinaria deberá ser al menos de 5.0 sobre 10.0.
- Si un alumno no alcanza los requisitos mínimos descritos en la tabla anterior, su calificación final en la asignatura será el mínimo entre el valor calculado según la ponderación descrita en la tabla y 4.5.

En el caso de la convocatoria extraordinaria se contemplan los dos siguientes escenarios:

1. Por defecto se mantiene la calificación del primer instrumento de la tabla y el alumno debe alcanzar una calificación igual o superior al 50% en la prueba escrita final. Además, la nota total resultante deberá ser igual o superior a 5.0 sobre 10.0.
2. A petición del interesado, el alumno puede renunciar a la calificación del proyecto (primer instrumento de la tabla) y realizar una serie de entregables correspondientes a un nuevo proyecto. Adicionalmente, el alumno debe alcanzar una calificación igual o superior al 50% en la prueba escrita final y obtener una nota final igual o superior a 5.0 sobre 10.0.

Además deberá tenerse en cuenta que, al igual que en la convocatoria ordinaria, si un alumno no alcanza los requisitos mínimos establecidos, su calificación final en la asignatura será el mínimo entre el valor calculado según la ponderación descrita y 4.5.

## 8. Consideraciones finales

- El Anexo I mencionado en la guía, donde se describe la planificación detallada, se entregará al comienzo de la asignatura.