

## FICHA TÉCNICA DE LA ASIGNATURA

Datos de la asignatura	
Nombre completo	Ingeniería del Software
Código	DTC-GITT-315
Título	<a href="#">Grado en Ingeniería en Tecnologías de Telecomunicación por la Universidad Pontificia Comillas</a>
Impartido en	Grado en Ingeniería en Tecnologías de Telecomunicación [Tercer Curso]
Nivel	Reglada Grado Europeo
Cuatrimestre	Semestral
Créditos	6,0 ECTS
Carácter	Optativa (Grado)
Departamento / Área	Departamento de Telemática y Computación
Responsable	Atilano Fernández-Pacheco Sánchez-Migallón
Horario	Tarde
Horario de tutorías	A petición del alumno

Datos del profesorado	
Profesor	
Nombre	Atilano Ramiro Fernández-Pacheco Sánchez-Migallón
Departamento / Área	Departamento de Telemática y Computación
Despacho	401
Correo electrónico	afernandezpacheco@icai.comillas.edu

## DATOS ESPECÍFICOS DE LA ASIGNATURA

Contextualización de la asignatura
<p><b>Aportación al perfil profesional de la titulación</b></p> <p>En el perfil profesional del graduado en Ingeniería en Tecnologías de Telecomunicación, la asignatura de Ingeniería del Software se encarga de mostrar los conceptos básicos de esta disciplina, que propone unos principios de ingeniería robustos, orientados a obtener software económico que sea fiable y funcione de manera eficiente sobre máquinas reales, y conseguir que los alumnos los pongan en práctica.</p> <p>Por lo tanto, se trata de que el alumno sea consciente de la necesidad de utilizar un proceso de desarrollo preestablecido y con una serie de fases determinadas a la hora de enfrentarse al desarrollo de un proyecto de telecomunicación de cierta envergadura. Para ello, se desarrolla a lo largo de todo el curso un proyecto en equipo en el que necesariamente han de aplicar las fases, técnicas y herramientas de desarrollo que se irán presentando de manera teórica durante el curso.</p>
<p><b>Prerequisitos</b></p> <p>Conocimientos de Programación Estructurada y Programación Orientada a Objetos</p>



## Competencias - Objetivos

### Competencias

#### GENERALES

<b>CG01</b>	Capacidad para redactar, desarrollar y firmar proyectos en el ámbito de la ingeniería de telecomunicación que tengan por objeto, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 de esta Orden (CIN/352/2009), la concepción y el desarrollo o la explotación de redes, servicios y aplicaciones de telecomunicación y electrónica.
<b>CG02</b>	Conocimiento, comprensión y capacidad para aplicar la legislación necesaria durante el desarrollo de la profesión de Ingeniero Técnico de Telecomunicación.
<b>CG05</b>	Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos en su ámbito específico de la telecomunicación.

#### ESPECÍFICAS

<b>CETM01</b>	Capacidad de construir, explotar y gestionar las redes, servicios, procesos y aplicaciones de telecomunicaciones, entendidas éstas como sistemas de captación, transporte, representación, procesado, almacenamiento, gestión y presentación de información multimedia, desde el punto de vista de los servicios telemáticos.
<b>CFBT02</b>	Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
<b>CRT01</b>	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.
<b>CRT02</b>	Capacidad de utilizar aplicaciones de comunicación e 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.
<b>CRT07</b>	Conocimiento y utilización de los fundamentos de la programación en redes, sistemas y servicios de telecomunicación.

### Resultados de Aprendizaje

<b>RA1</b>	Conocer y aplicar los conceptos básicos de Ingeniería del Software.
<b>RA2</b>	Conocer los modelos de Proceso Software más utilizados, describir y planificar un proceso software en particular adecuado a las características del sistema a desarrollar.
<b>RA3</b>	Conocer y aplicar las tareas propias de gestión del software, como gestión de proyectos, gestión de calidad y gestión de configuración. Utilizar las técnicas propias de cada una de las tareas de gestión.
<b>RA4</b>	Analizar las necesidades de un sistema y poder realizar la Especificación del Software, realizando una correcta especificación y validación de requisitos.
<b>RA5</b>	Comprender en qué consisten las técnicas de modelado de un sistema y conocer y aplicar las distintas técnicas de modelado que se utilizan en el Desarrollo de Aplicaciones Telemáticas



RA6	Conocer los criterios de diseño en el desarrollo de un sistema software y realizar el diseño de la arquitectura, de los interfaces, de los componentes y de las bases de datos de un proyecto en equipo utilizando las técnicas de modelado adecuadas.
RA7	Comprender en qué consiste la Validación & Verificación y Test de un sistema y aplicarlo a un proyecto en particular, así como conocer cómo se lleva a cabo la implantación de un sistema software y se gestiona la evolución del mismo.
RA8	Buscar información y bibliografía relacionada con el proyecto de desarrollo elegido por el equipo..
RA9	Presentar y defender, tanto oralmente como por escrito, los resultados obtenidos en el proyecto de desarrollo de una aplicación en grupo realizado a lo largo de todo el curso.

## BLOQUES TEMÁTICOS Y CONTENIDOS

### Contenidos – Bloques Temáticos

#### CAPITULO I - VISIÓN GENERAL DE LOS CONCEPTOS BÁSICOS DE INGENIERÍA DEL SOFTWARE.

1. Introducción.
2. Qué es la Ingeniería del Software y por qué es importante.
3. Procesos de desarrollo, técnicas y herramientas.
4. Principios éticos de Ingeniería del Software (ACM/IEEE Código de ética y práctica profesional de la Ingeniería del Software)

#### CAPÍTULO II – MODELOS DE PROCESOS SOFTWARE PARA EL DESARROLLO DE APLICACIONES. PROCESOS CLÁSICOS Y MÉTODOS ÁGILES. DESCRIPCIÓN DE LAS ACTIVIDADES PROPIAS DE CUALQUIER PROCESO DE DESARROLLO.

1. Introducción. Procesos Software: Dirigidos por un plan y Ágiles.
2. Actividades del Proceso. Objetivo.
  - Especificación del Software.
  - Desarrollo del Software.
  - Validación del Software.
  - Evolución del Software.
3. Procesos software dirigidos por un plan: cascada, incremental, reutilización, espiral, RUP.
4. Procesos software ágiles: XP, Scrum, ...
  1. Historia y manifiesto Agile.
  2. Roles.
  3. Product Backlog.
  4. Producto Mínimo Viable.
  5. Release.
  6. Sprint.
  7. Planning Poker.
  8. Método Kanban.

**CAPÍTULO III – GESTIÓN DEL SOFTWARE: GESTIÓN DE PROYECTO, GESTIÓN DE CALIDAD Y GESTIÓN DE CONFIGURACIÓN. CONFIABILIDAD Y SEGURIDAD.**

1. Gestión de proyectos. Funciones de Gestión:

- Estimación y Planificación.
- Seguimiento.
- Control.
- Gestión de Riesgos.

2. Gestión de Calidad. Garantía y control, Plan de Aseguramiento de Calidad (PAC).

ISO 9001, ISO 90003.

3. Gestión de configuración: control de versiones y de cambios.

4. Confiabilidad y Seguridad.

**CAPÍTULO IV – INGENIERÍA DE REQUISITOS: ESTUDIO DE VIABILIDAD, DESCUBRIMIENTO DE REQUISITOS Y TÉCNICAS PARA ESPECIFICAR LOS REQUISITOS. GESTIÓN DE REQUISITOS.**

1. Estudio de Viabilidad:

- a. Identificación de necesidades.
- b. Análisis económico de Coste – Beneficio.
- c. Conclusión.

2. Técnicas de Recopilación de Información.

- a. La técnica de la ENTREVISTA.

3. Introducción a los requisitos.

4. Requisitos Usuarios – Requisitos de Sistema.

5. Enfoque tradicional de requisitos – enfoque ágil.

6. Requisitos Funcionales y No funcionales. Requisitos de dominio.

7. Documento de Requisitos.

8. Obtención, Análisis y Especificación de Requisitos. Técnicas de Descubrimiento de Requisitos.

9. Validación de Requisitos (Actividad V&V&T)

10. Gestión de Requisitos (Actividad V&V&T).

11. Trazabilidad de los requisitos.

**CAPÍTULO V – VALIDACIÓN, VERIFICACIÓN Y TEST. TEORÍA Y EJEMPLOS DE TÉCNICAS A UTILIZAR. DISEÑO DE PRUEBAS.**

1. Tarea continuada a lo largo del Proceso de Desarrollo. Métodos de Prueba.

2. Tipos de prueba.

3. JUnit: Pruebas Unitarias en Java

**CAPÍTULO VI – MODELADO DEL SISTEMA. CONCEPTOS BÁSICOS, MODELOS ESTRUCTURALES Y DE COMPORTAMIENTO. UML Y OTRAS TÉCNICAS DE MODELADO.**

1. Introducción al modelado.

2. Diagramas de UML.

3. Modelado del sistema con UML:

4. Modelos de Contexto.
5. Modelos de Interacción.
6. Modelos Estructurales.
7. Modelos de Comportamiento

**CAPÍTULO VII – DISEÑO E IMPLEMENTACIÓN. DISEÑO ARQUITECTURAL, PATRONES DE DISEÑO, DISEÑOS DE INTERFACES, DISEÑO DE COMPONENTES. NORMAS DE LENGUAJES Y PROCEDIMIENTOS PARA LA IMPLEMENTACIÓN DEL SOFTWARE.**

1. Diseño Arquitectónico.
  - a. ¿En qué consiste el diseño arquitectónico?
  - b. Decisiones en el diseño arquitectónico.
  - c. Patrones arquitectónicos.
  - d. Elaboración del Diseño Arquitectónico.
2. Diseño de Interfaz. Navegación. Prototipado de interfaces.
3. Diseño de Componentes: Diseño Orientado a Objetos con UML. Patrones de Diseño.
4. Diseño de Bases de Datos.

**CAPÍTULO VIII – IMPLANTACIÓN Y EVOLUCIÓN DEL SOFTWARE. GESTIÓN DE CAMBIOS.**

1. Evolución y Mantenimiento.
  - a. Adaptativo.
  - b. Correctivo.
2. Eliminación de la Aplicación.
3. DevOps

**CAPÍTULO IX – TENDENCIAS ACTUALES EN INGENIERÍA DEL SOFTWARE. MODELOS DE REFERENCIA DE CALIDAD DEL SOFTWARE.**

1. Normas y modelos de referencia de gestión y de calidad del software. Mejoras en el Proceso.
  - a. ISO 9001, ISO 90003.
  - b. CMM, CMMI
  - c. ISO/IEC 15504: SPICE
  - d. ITIL
  - e. ISO/IEC 20000
2. Procesos Agiles vs Cascada
3. MDE – MDA.

**METODOLOGÍA DOCENTE**

**Aspectos metodológicos generales de la asignatura**

El método de trabajo combina una concepción flexible de la Lección Magistral con Prácticas dedicadas a la formulación y resolución de

problemas. Con el objeto de promover el papel activo del alumno se proponen problemas para que sean resueltos de forma individual en algunos casos y en grupos reducidos en otros. La evaluación de estos problemas se aprovecha para realizar un trabajo de orientación académica y seguimiento del aprendizaje de los alumnos.

Las sesiones en el laboratorio y sobre todo, el Proyecto Semestral del desarrollo de una aplicación completan la formación del alumno, ya que permite afianzar los conocimientos generales previamente transmitidos en las sesiones teóricas y extenderlos a aplicaciones diversas.

### **Metodología Presencial: Actividades**

#### **Lección magistral:**

La dinámica de este método de trabajo es siempre la misma. Los profesores utilizan los primeros 5 minutos de la clase para situar la clase que se va a impartir en el marco general de la asignatura, y relacionarlo con sesiones anteriores de la asignatura. A continuación plantea cuál es el objetivo de la lección que se va a impartir (¿para qué sirve lo que se va a ver?) y se comienza con la explicación de la lección.

Siempre intentando que los alumnos contextualicen los conceptos vistos en clase con el mundo profesional, se utilizan ejemplos prácticos y experiencias profesionales de los profesores para ligarlos con el temario que se está explicando a través de las transparencias, con el apoyo de un libro de consulta.

#### **Proyecto de Ingeniería de Software en equipo:**

Esta actividad es la central del curso, ya que con ella se quiere conseguir que:

- El alumno sienta la necesidad de utilizar de manera sistemática algún proceso de desarrollo predefinido al enfrentarse a un problema de tal envergadura que no es posible abordarlo de otra manera.
- Al tener que aplicar todo lo que se les va explicando en teoría, vayan estudiando la teoría e interiorizándola.
- Adquieran experiencia en un trabajo en equipo (que requiere una gran colaboración grupal), analizando explícitamente a final de curso las ventajas que este método de trabajo tiene y sus desventajas, que serán los problemas que necesariamente haya ido surgiendo en el desarrollo de la actividad.
- Sean capaces de abordar su proyecto de fin de carrera con mayor confianza dado que ya se han enfrentado a un proyecto de cierta envergadura.
- Experimenten directamente su futura realidad profesional.

A comienzo de curso, los alumnos forman grupos de 2 personas.

La aplicación que van a desarrollar en su Proyecto Semestral de telecomunicación se definirá al comienzo del curso.

Ciertas pautas de este proyecto se irá realizando en clase, bajo la guía y supervisión del profesor, y parte, deberán realizarlo los alumnos fuera del aula.

#### **Presentaciones orales:**

A lo largo del curso se podrían ir realizando presentaciones del proyecto de cada equipo, en el que participan obligatoriamente todos los integrantes del grupo con un reparto de tiempo similar entre ellos.

El profesor además de hacer preguntas a cada miembro del grupo de distintos aspectos de la presentación realizada, invita al resto de alumnos a participar y preguntar (indicando que se va a evaluar la participación activa en las presentaciones del resto de grupos).

#### **Evaluación por parte del alumno:**

Varias veces a lo largo del curso se solicita a los alumnos que se evalúen entre ellos:

- Valoración de las presentaciones: cada alumno rellena por cada presentación de grupo una encuesta nominal sobre aspectos del proyecto presentado, y su valoración de la presentación y del proyecto.

- Evaluación de su proyecto: dos o tres veces a lo largo del curso, se pide a los alumnos que evalúen el trabajo realizado, indicando el porcentaje de trabajo y la nota de proyecto que considera adecuada para todos los integrantes del equipo (incluido él mismo).

### **Metodología No presencial: Actividades**

#### **Estudio individualizado de parte del temario fuera del aula:**

Se le proporciona al alumno diverso material sobre una parte del temario y se le indica que lo prepare para una fecha en particular o que realice la lectura de algún artículo técnico, o prepare material de estudio de cierta parte de la asignatura, o realice el visionado de vídeos, etc. Posteriormente se comprobará la realización de estas tareas.

#### **Ejercicios individuales:**

A lo largo de todo el curso, los profesores proporcionan ejercicios, de los que algunos se resuelven en clase y otros, los resuelven los alumnos fuera del aula.

#### **Proyecto de desarrollo de aplicación de telecomunicación en equipo:**

Parte de las actividades de desarrollo del proyecto se han de realizar fuera del aula, reuniéndose el equipo.

### **Metodología Presencial: Actividades**

#### **Lección magistral:**

La dinámica de este método de trabajo es siempre la misma. Los profesores utilizan los primeros 5 minutos de la clase para situar la clase que se va a impartir en el marco general de la asignatura, y relacionarlo con sesiones anteriores de la asignatura. A continuación plantea cuál es el objetivo de la lección que se va a impartir (¿para qué sirve lo que se va a ver?) y se comienza con la explicación de la lección.

Siempre intentando que los alumnos contextualicen los conceptos vistos en clase con el mundo profesional, se utilizan ejemplos prácticos y experiencias profesionales de los profesores para ligarlos con el temario que se está explicando a través de las transparencias, con el apoyo de un libro de consulta.

#### **Resolución de problemas**

A lo largo de todo el curso, los profesores proporcionan ejercicios, de los que algunos se resuelven en clase y otros, los resuelven los alumnos fuera del aula.

#### **Proyecto de Ingeniería de Software en equipo:**

Esta actividad es la central del curso, ya que con ella se quiere conseguir que:

- El alumno sienta la necesidad de utilizar de manera sistemática algún proceso de desarrollo predefinido al enfrentarse a un problema de tal envergadura que no es posible abordarlo de otra manera.
- Al tener que aplicar todo lo que se les va explicando en teoría, vayan estudiando la teoría e interiorizándola.
- Adquieran experiencia en un trabajo en equipo (que requiere una gran colaboración grupal), analizando explícitamente a final de curso las ventajas que este método de trabajo tiene y sus desventajas, que serán los problemas que necesariamente haya ido surgiendo en el desarrollo de la actividad.
- Sean capaces de abordar su proyecto de fin de carrera con mayor confianza dado que ya se han enfrentado a un proyecto de cierta envergadura.
- Experimenten directamente su futura realidad profesional.

A comienzo de curso, los alumnos forman grupos de 3 o 4 personas.

La aplicación que van a desarrollar en su Proyecto Semestral de telecomunicación se definirá al comienzo del curso.

Ciertas pautas de este proyecto se irá realizando en clase, bajo la guía y supervisión del profesor, y parte, deberán realizarlo los alumnos fuera del aula.

### Metodología No presencial: Actividades

#### Resolución de problemas

A lo largo de todo el curso, los profesores proporcionan ejercicios, de los que algunos se resuelven en clase y otros, los resuelven los alumnos fuera del aula.

#### Trabajo autónomo sobre contenidos teóricos

Se le proporciona al alumno diverso material sobre distintas partes del temario y se le indica que lo prepare para una fecha en particular o que realice la lectura de algún artículo técnico, o prepare material de estudio de cierta parte de la asignatura, o realice el visionado de videos, etc. Posteriormente se comprobará la realización de las tareas que van a ser evaluadas.

#### Proyecto de Ingeniería de Software en equipo:

Esta actividad es la central del curso, ya que con ella se quiere conseguir que:

- El alumno sienta la necesidad de utilizar de manera sistemática algún proceso de desarrollo predefinido al enfrentarse a un problema de tal envergadura que no es posible abordarlo de otra manera.
- Al tener que aplicar todo lo que se les va explicando en teoría, vayan estudiando la teoría e interiorizándola.
- Adquieran experiencia en un trabajo en equipo (que requiere una gran colaboración grupal), analizando explícitamente a final de curso las ventajas que este método de trabajo tiene y sus desventajas, que serán los problemas que necesariamente haya ido surgiendo en el desarrollo de la actividad.
- Sean capaces de abordar su proyecto de fin de carrera con mayor confianza dado que ya se han enfrentado a un proyecto de cierta envergadura.
- Experimenten directamente su futura realidad profesional.

A comienzo de curso, los alumnos forman grupos de 3 o 4 personas.

La aplicación que van a desarrollar en su Proyecto Semestral de telecomunicación se definirá al comienzo del curso.

Ciertas pautas de este proyecto se irá realizando en clase, bajo la guía y supervisión del profesor, y parte, deberán realizarlo los alumnos fuera del aula.

#### Estudio y preparación teórica

Los alumnos deben ir estudiando cada una de las partes de la asignatura, para poder ponerlo en práctica en el proyecto de desarrollo.

### RESUMEN HORAS DE TRABAJO DEL ALUMNO

HORAS PRESENCIALES		
Clase magistral y presentaciones generales	Resolución de problemas de carácter práctico o aplicado	Prácticas de laboratorio
15.00	19.00	26.00
HORAS NO PRESENCIALES		
Trabajo autónomo sobre contenidos teóricos por parte del alumno	Trabajos de carácter práctico individual o de grupo	Prácticas de diseño y desarrollo de un proyecto





50.00	30.00	40.00
CRÉDITOS ECTS: 6,0 (180,00 horas)		

## EVALUACIÓN Y CRITERIOS DE CALIFICACIÓN

Actividades de evaluación	Criterios de evaluación	Peso
Exámenes: <ul style="list-style-type: none"><li>Prueba Intersemestral (15%)</li><li>Examen final (45%)</li></ul>	<ul style="list-style-type: none"><li>Comprensión de conceptos.</li><li>Aplicación de conceptos a la resolución de problemas prácticos.</li><li>Presentación y estructura.</li></ul>	60
Evaluación continua del rendimiento: <ul style="list-style-type: none"><li>Proyecto de desarrollo de una Aplicación de Telecomunicación aplicando las técnicas aprendidas durante el curso</li></ul>	<ul style="list-style-type: none"><li>Adecuación de la metodología.</li><li>Documentación entregada.</li><li>Capacidad de trabajo en equipo.</li><li>Calidad del desarrollo realizado.</li><li>Capacidad de comunicación oral.</li></ul>	35
Evaluación del trabajo experimental: <ul style="list-style-type: none"><li>Entrega de trabajos diversos.</li></ul>	<ul style="list-style-type: none"><li>Ejercicios resueltos por el alumno.</li><li>Cuestionarios sobre lecturas realizadas y visionado de videos.</li><li>Otros.</li></ul>	5

## Calificaciones

### Convocatoria ordinaria

La calificación en la convocatoria ordinaria de la asignatura se obtiene de:

#### 1. Exámenes (60% del total):

15% Examen intercuatrimestral.

45% Examen final.

#### 2. Proyecto de Ingeniería del Software (35%)

La nota de equipo del proyecto se obtiene a lo largo del curso con las diferentes entregas ponderadas de la siguiente manera:

- 10%: Estudio de Viabilidad y Backlog inicial.
- 35%: Funcionamiento de Scrum: Refinamientos del backlog, retrospectivas, daily, etc.
- 40%: Documento de Diseño: diseño de la arquitectura técnica, modelo de contexto, diagrama de navegabilidad, diagrama de componentes, diseño de UML (diagrama de clases, de secuencia, etc) y matriz de trazabilidad.
- 10%: Presentación oral del proyecto donde deben participar todos los integrantes del grupo, evaluando el diseño y los contenidos de la misma.
- 5%: Calidad de los informes entregados: redacción, maquetación, etc.

La **Nota final de Proyecto** de cada alumno se obtiene ponderando su **nota de equipo** con su **nota personal**. La nota personal de cada alumno evalúa el grado de participación del alumno en las tareas llevadas a cabo en el equipo y también se evalúa cómo ha realizado la presentación. Se puede dar el caso de un alumno que suspenda el proyecto por no haber participado activamente en el desarrollo del mismo a lo largo del curso, aunque la nota de equipo del proyecto está aprobada.

El Proyecto de Desarrollo de Aplicación de telecomunicación sólo puede aprobarse en la convocatoria ordinaria, ya que al tratarse de un trabajo en equipo, no puede realizarse de manera individual para la convocatoria extraordinaria.

### 3. Entrega de trabajos diversos (5%)

- Lectura de artículos y resumen o comentarios.
- Ejercicios resueltos por el alumno.
- Cuestionarios sobre lecturas realizadas.
- Visionado de videos
- Otros

La nota mínima de los del Examen Final y la nota final del Proyecto debe ser 5 para ponderar la nota de la convocatoria ordinaria como se indica.

### Convocatoria extraordinaria

La calificación en la convocatoria extraordinaria de la asignatura se obtiene de:

#### 1. Exámenes (60% del total):

60% Examen final extraordinario.

#### 2. Proyecto de Ingeniería del Software (35%)

La nota de equipo del proyecto se obtiene a lo largo del curso con las diferentes entregas ponderadas de la siguiente manera:

- 10%: Estudio de Viabilidad y Backlog inicial.
- 35%: Funcionamiento de Scrum: Refinamientos del backlog, retrospectivas, daily, etc.
- 40%: Documento de Diseño: diseño de la arquitectura técnica, modelo de contexto, diagrama de navegabilidad, diagrama de componentes, diseño de UML (diagrama de clases, de secuencia, etc) y matriz de trazabilidad.
- 10%: Presentación oral del proyecto donde deben participar todos los integrantes del grupo, evaluando el diseño y los contenidos de la misma.
- 5%: Calidad de los informes entregados: redacción, maquetación, etc.

La **Nota final de Proyecto** de cada alumno se obtiene ponderando su **nota de equipo** con su **nota personal**. La nota personal de cada alumno evalúa el grado de participación del alumno en las tareas llevadas a cabo en el equipo y también se evalúa cómo ha realizado la presentación. Se puede dar el caso de un alumno que suspenda el proyecto por no haber participado activamente en el desarrollo del mismo a lo largo del curso, aunque la nota de equipo del proyecto está aprobada.

El Proyecto de Desarrollo de Aplicación de telecomunicación sólo puede aprobarse en la convocatoria ordinaria, ya que al tratarse de un trabajo en equipo, no puede realizarse de manera individual para la convocatoria extraordinaria.

### 3. Entrega de trabajos diversos (5%)

- Lectura de artículos y resumen o comentarios.
- Ejercicios resueltos por el alumno.
- Cuestionarios sobre lecturas realizadas.
- Visionado de videos
- Otros

La nota mínima de los del Examen Final y la nota final del Proyecto debe ser 5 para ponderar la nota de la convocatoria extraordinaria como se indica.

## BIBLIOGRAFÍA Y RECURSOS

### Bibliografía Básica

"Análisis y Diseño orientado a Objetos con UML y el Proceso unificado", Stephen R. Schach, McGrawHill, 2005.

"Software Engineering 9", Ian Sommerville, Pearson, 2010

"Software Engineering Tenth Edition", Ian Sommerville, Finantial Times/Prentice Hall, 2015

### Bibliografía Complementaria

"Software Engineering: A Practitioner's Approach", Roger Pressman, 6ª Edición (2004), 8ª Edición (2014)

"UML distilled: a brief guide to the standard object modeling language". Martin Fowler. Addison Wesley. 2003, Martin Fowler, 3ª Edición (2003)

Metodología del análisis estructurado de sistemas. Jesús Barranco de Areba. Universidad Pontificia Comillas de Madrid. Segunda edición 2001

En cumplimiento de la normativa vigente en materia de **protección de datos de carácter personal**, le informamos y recordamos que puede consultar los aspectos relativos a privacidad y protección de datos que ha aceptado en su matrícula entrando en esta web y pulsando "descargar"

[https://servicios.upcomillas.es/sedelectronica/inicio.aspx?csv=02E4557CAA66F4A81663AD10CED66792](https://servicios.upcomillas.es/sedeelectronica/inicio.aspx?csv=02E4557CAA66F4A81663AD10CED66792)