

**Proyecto/Guía docente de la asignatura**

<b>Asignatura</b>	DESARROLLO DE APLICACIONES TELEMÁTICAS DISTRIBUIDAS		
<b>Materia</b>	INFRAESTRUCTURAS, REDES Y SERVICIOS		
<b>Módulo</b>	TECNOLOGÍAS DE TELECOMUNICACIÓN		
<b>Titulación</b>	MÁSTER EN INGENIERÍA DE TELECOMUNICACIÓN		
<b>Plan</b>	736	<b>Código</b>	55255
<b>Periodo de impartición</b>	2º CUATRIMESTRE (1º bimestre)	<b>Tipo/Carácter</b>	OBLIGATORIA
<b>Nivel/Ciclo</b>	MÁSTER	<b>Curso</b>	1º
<b>Créditos ECTS</b>	3		
<b>Lengua en que se imparte</b>	CASTELLANO		
<b>Profesor/es responsable/s</b>	Manuel Rodríguez Cayetano		
<b>Datos de contacto (E-mail, teléfono...)</b>	TELÉFONO: 983 423000 ext. 5541 E-MAIL: <a href="mailto:manuel.rodriguez@tel.uva.es">manuel.rodriguez@tel.uva.es</a>		
<b>Departamento</b>	TEORÍA DE LA SEÑAL Y COMUNICACIONES E INGENIERÍA TELEMÁTICA		
<b>Fecha de revisión por el Comité de Título</b>	7 de julio de 2025		



## 1. Situación / Sentido de la Asignatura

---

### 1.1 Contextualización

---

La aparición y generalización de las redes de ordenadores ha dado lugar a la aparición de dos nuevos tipos de aplicaciones: las aplicaciones en red y las aplicaciones distribuidas. Estas aplicaciones se caracterizan por estar constituidas por varias partes que se pueden ejecutar en diferentes ordenadores. La coordinación entre estas partes se lleva a cabo mediante el intercambio de información a través de la red.

El desarrollo de una aplicación en red supone un esfuerzo superior al de una aplicación centralizada, ya que el desarrollador debe considerar todas las tareas de bajo nivel necesarias para el envío y recepción de información a través de la red y la gestión de los errores que puedan surgir durante esta comunicación. El desarrollo de las aplicaciones en red se basa en el uso de una API (interfaz de programación de aplicaciones) de acceso a los servicios de transporte de datos proporcionados por el sistema operativo. Esta API permite la solicitud de operaciones de bajo nivel relacionadas con el uso de la red (establecimiento y liberación de conexiones, envío y recepción de paquetes, etc.)

Para simplificar el desarrollo de las aplicaciones en red, de forma que la complejidad sea similar a la del desarrollo de una aplicación centralizada, surge el concepto de middleware o software intermediario. El middleware es, básicamente, una capa de software situada entre el sistema operativo y la aplicación a desarrollar, y que permite a la aplicación usar funciones de alto nivel (ejecución de una función en un ordenador remoto) sin necesidad de usar explícitamente las funciones de bajo nivel necesarias para la interacción con la red de comunicación. La aparición del middleware ha hecho surgir un nuevo tipo de aplicaciones, las aplicaciones distribuidas, que básicamente se diferencian de las aplicaciones en red en que el desarrollador no necesita incluir código para acceder a las funciones de bajo nivel relacionadas con la red (las operaciones sobre la red necesarias para acceder a un recurso remoto se realizan implícitamente mediante el software intermediario). En este concepto de middleware se basan las tecnologías para el desarrollo de aplicaciones distribuidas utilizadas mayoritariamente en la actualidad, como pueden ser .Net o servicios web.

En las titulaciones de grado previas al máster al que pertenece la presente asignatura se han estudiado las aplicaciones en red y los middlewares más sencillos, que permiten adaptar una aplicación centralizada y convertirla en una distribuida manteniendo la estructura de las funciones o métodos que ofrece. La asignatura que nos ocupa tratará tipos de middleware más avanzados que ofrecen mejores prestaciones en tiempo de ejecución y mejor escalabilidad ante el aumento de clientes, pero que requieren un diseño adaptado a la distribución desde las primeras fases de la construcción de la aplicación.

Los conceptos teóricos impartidos serán complementados con prácticas de diseño y desarrollo de una aplicación distribuida basada en middleware.

### 1.2 Relación con otras materias

---

Esta asignatura se construye sobre conocimientos de sistemas distribuidos más básicos, y en particular de una buena comprensión de los problemas de la arquitectura cliente/servidor, así como de programación de aplicaciones en red mediante *sockets* o aplicaciones distribuidas basadas en algún tipo de *middleware* (Java

RMI, .Net, etc.). Dichos conocimientos son impartidos, según el caso, en asignaturas de titulaciones de grado, o en asignaturas de complementos de formación en este máster.

En el primer caso se encuentra la asignatura “Desarrollo de Aplicaciones Distribuidas”, del Grado en Tecnologías de Telecomunicación (tercer curso), en el que se imparten conocimientos de middleware diferente al presentado aquí.

En el Grado en Tecnologías Específicas de Telecomunicación, mención en Telemática, la asignatura “Arquitecturas de Aplicaciones Distribuidas” (tercer curso) también presenta conocimientos de middleware más básicos que el presentado en esta asignatura.

La asignatura “Desarrollo de Aplicaciones Telemáticas”, del Grado en Tecnologías Específicas de Telecomunicación, mención en Sistemas de Telecomunicación (tercer curso) también incluye contenidos relacionados con el middleware, aunque con menor extensión que en las anteriores debido a que en su temario también se incluyen contenidos relacionados con el desarrollo de aplicaciones web.

Finalmente, dentro de este Máster la asignatura “Arquitecturas y tecnologías para el desarrollo de aplicaciones distribuidas”, está presente en los complementos de formación para graduados en Tecnologías Específicas de Telecomunicación con Mención en Sistemas Electrónicos, ofreciendo una formación semejante a las tres asignaturas de grado mencionadas anteriormente para aquellos alumnos que en su grado original no tenían dicha formación.

También está muy relacionada con el “Taller de Proyectos II”.

---

### 1.3 Prerrequisitos

No existen requisitos previos.

Las prácticas de laboratorio conllevarán una carga importante de actividades de programación en Java. Es muy recomendable que los alumnos manejen dicho lenguaje con soltura, así como los conceptos de la orientación a objetos. Por este motivo, es aconsejable que los alumnos hayan superado las asignaturas de Programación e Ingeniería de Sistemas Software, que son comunes a todas las titulaciones de grado, en sus distintas menciones, que son requisito para el acceso al Máster en Ingeniero de Telecomunicación.

---

## 2. Competencias

---

### 2.1 Generales

G8. Capacidad para la aplicación de los conocimientos adquiridos y resolver problemas en entornos nuevos o poco conocidos dentro de contextos más amplios y multidisciplinares, siendo capaces de integrar conocimientos.

G11. Capacidad para saber comunicar (de forma oral y escrita) las conclusiones -y los conocimientos y razones últimas que las sustentan- a públicos especializados y no especializados de un modo claro y sin ambigüedades.

G12. Poseer habilidades para el aprendizaje continuado, autodirigido y autónomo.

## 2.2 Específicas

---

TEL1. Capacidad para modelar, diseñar, implantar, gestionar, operar, administrar y mantener redes, servicios y contenidos.

TEL3. Capacidad de comprender y saber aplicar el funcionamiento y organización de Internet, las tecnologías y protocolos de Internet de nueva generación, los modelos de componentes, software intermediario y servicios.

## 3. Objetivos

---

Al finalizar la asignatura el alumno deberá ser capaz de:

- Conocer distintas alternativas de middleware para la programación de una aplicación telemática distribuida.
- Conocer las características de un diseño orientado a recursos y del estilo arquitectural REST.
- Diseñar aplicaciones telemáticas distribuidas siguiendo el estilo arquitectural REST.
- Conocer someramente marcos REST para la implementación de aplicaciones.
- Desarrollar aplicaciones telemáticas distribuidas apoyándose en un marco de desarrollo REST concreto.
- Trabajar en grupo.

## 4. Contenidos y/o bloques temáticos

---

### Bloque 1: Middleware para el desarrollo de aplicaciones telemáticas distribuidas

---

Carga de trabajo en créditos ECTS: 3

#### a. Contextualización y justificación

---

Véase el apartado 1.1.

#### b. Objetivos de aprendizaje

---

Véase el apartado 3.

#### c. Contenidos

---

##### Tema 1: Conceptos básicos

- 1.1 Tipos de sistemas informáticos: sistemas centralizados, redes de ordenadores y sistemas distribuidos
- 1.2 Tipos de software geográficamente distribuido: aplicaciones en red, sistemas operativos distribuidos y aplicaciones distribuidas basadas en middleware
- 1.3 Concepto de middleware y tipos de middleware

##### Tema 2: Sistemas distribuidos avanzados

- 2.1 Middleware orientado a servicios
- 2.2 El estilo arquitectural REST
- 2.3 Arquitectura orientada a recursos

##### Tema 3: Marcos de desarrollo REST

- 3.1 Introducción
- 3.2 El marco de desarrollo Jersey
- 3.3 Prácticas de diseño y desarrollo según el estilo arquitectural REST y el marco Jersey



#### d. Métodos docentes

Se emplearán:

- Clase magistral participativa
- Taller de prácticas guiadas en el laboratorio

#### e. Plan de trabajo

La planificación detallada (plan de trabajo o Anexo I) se entregará al comienzo de la asignatura.

#### f. Evaluación

La evaluación de la adquisición de competencias se basará en:

- Programas e informes realizados por grupos de alumnos según lo indicado en los enunciados de prácticas.
- Prueba escrita al final del cuatrimestre.

#### g Material docente

##### g.1 Bibliografía básica

Bibliografía disponible en el sistema Leganto, URL:

[https://buc-uva.alma.exlibrisgroup.com/leganto/public/34BUC\\_UVA/lists/8258669280005774?auth=SAML](https://buc-uva.alma.exlibrisgroup.com/leganto/public/34BUC_UVA/lists/8258669280005774?auth=SAML)

##### g.2 Bibliografía complementaria

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

#### h. Recursos necesarios

Bibliografía y documentación proporcionada por los profesores.

Entorno de desarrollo formado por distintos elementos gratuitos y/o de código abierto (Eclipse, Oracle JDK, marco Restlet, etc.).

#### i. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
3 ECTS	Semanas 1 a 8 del segundo cuatrimestre (primer bimestre)

### 5. Métodos docentes y principios metodológicos

- Clase magistral participativa

- Taller de prácticas guiadas en el laboratorio

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

ACTIVIDADES PRESENCIALES o PRESENCIALES A DISTANCIA <sup>(1)</sup>	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	15	Estudio y trabajo autónomo individual	15
Clases prácticas de aula (A)		Estudio y trabajo autónomo grupal	30
Laboratorios (L)	15		
Total presencial	30	Total no presencial	45
TOTAL presencial + no presencial			75

1. Actividad presencial a distancia es aquella en la que 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
Informes de prácticas de laboratorio y evaluación del funcionamiento de los programas desarrollados (LAB)	50%	Es condición necesaria (pero no suficiente) alcanzar una calificación igual o superior a 5 puntos sobre 10 en esta parte para superar la asignatura. <b>Es obligatorio presentar todas las prácticas planteadas por el profesor. En caso contrario, la nota de laboratorio obtenida será un No presentado.</b>
Examen final escrito (EXA)	50%	Es condición necesaria (pero no suficiente) alcanzar una calificación igual o superior a 5 puntos sobre 10 en esta parte para superar la asignatura.

**CRITERIOS DE CALIFICACIÓN**

- **Convocatoria ordinaria:**
  - La nota final para superar la asignatura en la convocatoria ordinaria deberá ser igual o superior a 5 puntos sobre 10.
  - Si un alumno no alcanza los requisitos descritos en la tabla anterior, su calificación final en la asignatura será el mínimo entre el valor calculado según la ponderación descrita en dicha tabla y 4,5 puntos sobre 10.
- **Convocatoria extraordinaria:**
  - Se mantiene la calificación obtenida en cada instrumento de la tabla siempre que dicha calificación sea igual o superior a 5 sobre 10 puntos.
  - El alumno deberá realizar de nuevo el examen final escrito si la nota obtenida es inferior a 5 sobre 10 puntos en la convocatoria ordinaria.
  - El alumno deberá realizar de nuevo la práctica o prácticas suspensas (puntuación inferior a 5 sobre 10 puntos), siguiendo los enunciados planteados para la convocatoria extraordinaria, si la nota total de prácticas es inferior a 5 sobre 10 puntos en la convocatoria ordinaria.

**USO DE INTELIGENCIA ARTIFICIAL GENERATIVA**

El uso de herramientas de Inteligencia Artificial generativa (como ChatGPT u otras) no está recomendado por el profesor, aunque está permitido, exclusivamente, como apoyo para aclarar dudas, obtener sugerencias, mejorar la comprensión de conceptos o identificar errores en el trabajo del estudiante, de forma que le ayude a depurar y perfeccionar el mismo.

No está permitido, no obstante, utilizar estas herramientas para generar de forma parcial, completa o automática entregables, ya sean informes, código o ficheros de configuración de programas, requeridos en esta asignatura.

El incumplimiento de esta norma podrá considerarse una infracción de la honestidad académica y tendrá una penalización sobre la calificación de hasta el 100% de la nota obtenida.

El profesorado prefiere y sugiere al alumno que use sistemas de búsqueda tradicional (no basados en inteligencia artificial) y especializados en los temas relativos a la asignatura, en el que los resultados estén respaldados por personas claramente identificadas, como la página [www.stackoverflow.com](http://www.stackoverflow.com) y similares.

**8. Consideraciones finales**

- La planificación detallada de la asignatura (plan de trabajo o Anexo I) se entregará al comienzo de la asignatura.