

**Guía docente de la asignatura**

<b>Asignatura</b>	PARADIGMAS Y TECNOLOGÍAS DE PROCESAMIENTO DISTRIBUIDO		
<b>Materia</b>	INGENIERÍA DE REDES, SISTEMAS Y SERVICIOS TELEMÁTICOS		
<b>Módulo</b>	(Dejar en blanco)		
<b>Titulación</b>	MÁSTER EN INGENIERÍA DE TELECOMUNICACIÓN		
<b>Plan</b>	544	<b>Código</b>	(Dejar en blanco)
<b>Periodo de impartición</b>	2º CUATRIMESTRE	<b>Tipo/Carácter</b>	OBLIGATORIA
<b>Nivel/Ciclo</b>	MÁSTER	<b>Curso</b>	1º
<b>Créditos ECTS</b>	6 ECTS		
<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: manuel.rodriguez@tel.uva.es		
<b>Horario de tutorías</b>	Véase <a href="http://www6.uva.es">www6.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

Los sistemas telemáticos distribuidos forman parte de la economía actual, y son utilizados en dominios tan diversos como la gestión de información de cualquier tipo, el comercio electrónico, el aprendizaje, la medicina, la investigación científica o la gestión de sistemas de producción. El modelo clásico de cliente-servidor es el más utilizado, siendo por ello también el más estudiado en titulaciones de nivel inferior. Algunos tipos de *middleware* de sistemas distribuidos son hoy poco utilizados (API de sockets, RPCs, CORBA...) mientras que otros tienen una gran popularidad, o hay un cierto debate acerca de ellos (Java RMI, Web Services, REST). En cualquier caso, en su forma básica el modelo cliente-servidor presenta limitaciones de escalabilidad, robustez, disponibilidad, coste o anonimato, entre otros, que pueden ser mejorados con sistemas más distribuidos.

Entre los paradigmas de computación distribuida más recientes se encuentran las redes P2P, famosas por su uso para el intercambio de ficheros, pero con gran potencialidad gracias a la proliferación de dispositivos móviles; la computación orientada a servicios, que promueve la interoperabilidad de software de múltiples partes en sistemas heterogéneos mediante el uso de unos protocolos y estándares comunes; la computación en malla (*grid*), que se caracteriza por la federación de recursos de múltiples instituciones para alcanzar una alta escalabilidad, y que ha convergido con la computación orientada a servicios; y la computación en nube (*cloud*) que promueve la contratación bajo demanda de recursos o servicios en la red para ofrecer servicios utilizando a su vez servicios de terceros.

En este contexto, esta asignatura ofrece al alumno la oportunidad de conocer y experimentar de manera práctica con paradigmas de distribución y tipos de *middleware* más complejos que los que ha estudiado en las titulaciones de grado.

### 1.2 Relación con otras materias

Esta asignatura 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 sistemas distribuidos mediante *sockets* o algún tipo de *middleware* (Java RMI...). 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* antecesor del 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 tiene una fuerte carga de contenidos de middleware, algo menor que en los casos anteriores.

Finalmente, dentro de éste 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.

### 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.

## **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.
- G9. Capacidad para comprender la responsabilidad ética y la deontología profesional de la actividad de la profesión de Ingeniero de Telecomunicación.
- G10. Capacidad para aplicar los principios de la economía y de la gestión de recursos humanos y proyectos, así como la legislación, regulación y normalización de las telecomunicaciones.
- 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 los diversos paradigmas, así como las arquitecturas de los sistemas telemáticos distribuidos, pudiendo emplear los más adecuados en cada caso
- Conocer diversas técnicas de procesamiento distribuido.
- Comprender los distintos niveles de abstracción para ofrecer un servicio telemático.
- Aplicar de manera práctica middleware para el desarrollo y despliegue de sistemas de procesamiento distribuido.
- Tener una postura crítica hacia las tecnologías de sistemas telemáticos, así como de conceptos emergentes.
- Aprender y trabajar en grupo.

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

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	15	Estudio y trabajo autónomo individual	20
Clases prácticas de aula (A)		Estudio y trabajo autónomo grupal	70
Laboratorios (L)	45		
Prácticas externas, clínicas o de campo			
Seminarios (S)			
Tutorías grupales (TG)			
Evaluación (fuera del periodo oficial de exámenes)			
<b>Total presencial</b>	<b>60</b>	<b>Total no presencial</b>	<b>90</b>

## 5. Bloques temáticos

---

### Bloque 1: Middleware para el desarrollo de sistemas telemáticos

---

Carga de trabajo en créditos ECTS: 

2,8
-----

#### a. Contextualización y justificación

---

La asignatura se organiza en dos grandes bloques: uno de middleware que permite la programación de sistemas telemáticos o distribuidos, y otro acerca de paradigmas de distribución. Este primer bloque permitirá a los alumnos conocer algunas tecnologías que facilitan al programador la comunicación entre las partes de un sistema distribuido y profundizar en una de ellas, realizando distintas prácticas de programación. El siguiente bloque le permitirá conocer arquitecturas de distribución y desplegar las prácticas realizadas en este bloque en una nube computacional.

#### b. Objetivos de aprendizaje

---

Al finalizar este bloque temático, el alumno deberá ser capaz de:

- Conocer distintas alternativas de middleware para la programación de un sistema distribuido.
- Conocer las características de un diseño orientado a recursos y del estilo arquitectural REST.
- Diseñar sistemas distribuidos en casos prácticos siguiendo el estilo arquitectural REST.
- Conocer someramente varios marcos REST para la implementación de aplicaciones.
- Desarrollar sistemas distribuidos en casos prácticos apoyándose en un marco de desarrollo REST concreto.
- Aprender y trabajar en grupo.

#### c. Contenidos

---

##### TEMA 1: Sistemas distribuidos avanzados

- 1.1 Repaso de conceptos de sistemas distribuidos
- 1.2 Middleware y tipos de middleware
- 1.3 El estilo arquitectural REST
- 1.4 Marcos de desarrollo REST
- 1.5 Prácticas de diseño y desarrollo según el estilo arquitectural REST

#### d. Métodos docentes

---

Se emplearán:

- Clase magistral participativa
- Estudio de casos en aula y en laboratorio
- Resolución de problemas
- Método de proyectos
- Trabajo en grupo

#### e. Plan de trabajo

---

Véase el Anexo I.

#### **f. Evaluación**

---

La evaluación de la adquisición de competencias se basará en:

- Informes realizado por grupos de alumnos sobre el caso práctico
- Prueba escrita al final del cuatrimestre.

#### **g. Bibliografía básica**

---

L. Richardson, S. Ruby, *Restful Web Services*, O'Reilly, May 2007.

J. Louvel, T. Templier, T. Boileau, *Restlet in Action: Developing RESTful web APIs in Java*, Manning Publications, 2013.

#### **h. Bibliografía complementaria**

---

#### **i. 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...).

### **Bloque 2:Paradigmas de sistemas distribuidos**

---

Carga de trabajo en créditos ECTS: 

3,2
-----

#### **a. Contextualización y justificación**

---

Este segundo bloque versa sobre las aproximaciones arquitectónicas para la creación de sistemas distribuidos, desde la clásica arquitectura cliente/servidor hasta sistemas fuertemente distribuidos como el P2P, con un énfasis particular en la nube computacional. En este sentido, complementa al bloque anterior logrando el resto de los objetivos de la asignatura, y permitiendo que los alumnos puedan desplegar las prácticas realizadas en él en un entorno de nube computacional, con distintas configuraciones.

#### **b. Objetivos de aprendizaje**

---

Al finalizar este bloque temático, el alumno deberá ser capaz de:

- Conocer los diversos paradigmas de sistemas distribuidos.
- Argumentar sobre la conveniencia de uno u otro según los requisitos de la aplicación.
- Conocer qué se entiende comúnmente por nube computacional.
- Comprender los distintos niveles de abstracción para ofrecer un servicio telemático.
- Configurar, desplegar, monitorizar y terminar máquinas virtuales en una infraestructura de nube computacional.
- Desarrollar aplicaciones que interaccionen con la API de una infraestructura de nube computacional.
- Aprender y trabajar en grupo.

#### **c. Contenidos**

---

**TEMA 2: Introducción a la computación en nube**



- 2.1 Definición del concepto de nube computacional.
- 2.2 Niveles de abstracción de servicios (XaaS).
- 2.3 Plataformas comerciales de IaaS y PaaS.
- 2.4 Infraestructuras abiertas para el despliegue de nubes IaaS.
- 2.5 Manejo práctico de una infraestructura de nube IaaS.
- 2.5 Despliegue de aplicaciones distribuidas en una nube IaaS.

### **TEMA 3: Manejo de las API de nubes IaaS**

- 3.1 Introducción a las API de las infraestructuras más populares.
- 3.2 Desarrollo de aplicaciones sobre una API de infraestructura de nube.

### **TEMA 4: Otros paradigmas de sistemas distribuidos**

- 4.1 Computación en malla (*grid*).
- 4.2 Computación P2P.

---

#### **d. Métodos docentes**

Se emplearán:

- Clase magistral participativa
- Estudio de casos en aula y en laboratorio
- Resolución de problemas
- Método de proyectos
- Trabajo en grupo

---

#### **e. Plan de trabajo**

Véase el Anexo I.

---

#### **f. Evaluación**

La evaluación de la adquisición de competencias se basará en:

- Informes realizados por grupos de alumnos sobre el caso práctico
- Prueba escrita al final del cuatrimestre.

---

#### **g. Bibliografía básica**

G. Coulouris, J. Dollimore, T. Kindberg, *Distributed Systems, Concepts and Design*, 5th edition, Addison-Wesley, 2012.

#### **h. Bibliografía complementaria**

---

Sun Microsystems, *Introduction to Cloud Computing Architecture*, White Paper, 1st edition, June 2009.  
<https://java.net/jira/secure/attachment/29265/CloudComputing.pdf>

A. T. Velte, T.J. Velte, R. Elsenpeter, *Cloud computing: a practical approach*, McGraw Hill, 2010.

#### **i. 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...).

Infraestructura de nube computacional a partir de una plataforma abierta (ej. OpenStack) y pública (ej. Amazon).

## 6. Temporalización (por bloques temáticos)

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1: Middleware para el desarrollo de sistemas telemáticos	2,8 ECTS	Semanas 1 a 7
Bloque 2: Paradigmas de sistemas distribuidos	3,2 ECTS	Semanas 8 a 15

## 7. Sistema de calificaciones – Tabla resumen

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Informes de prácticas de laboratorio (LAB)	25%	Es condición necesaria (pero no suficiente) alcanzar una calificación igual o superior a 5 puntos sobre 10 en esta parte para superar la asignatura.
Examen final escrito (EXA)	75%	Es condición necesaria (pero no suficiente) alcanzar una calificación igual o superior a 5 puntos sobre 10 en esta parte para superar la asignatura.

Además deberá tenerse en cuenta que:

- 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 la tabla y 4,5 puntos sobre 10.

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

- Si el alumno ha obtenido en el laboratorio (LAB) una puntuación igual o superior a 5 puntos sobre 10, se le mantiene dicha puntuación para la convocatoria extraordinaria. El alumno debe presentarse al examen escrito (EXA) de la convocatoria extraordinaria y obtener en él una nota igual o superior a 5 puntos sobre 10 para superar la asignatura.
- Incluso en el caso anterior, el alumno puede solicitar la renuncia a la nota del laboratorio (LAB). Debe hacerlo de manera expresa, por escrito, y dentro del plazo que se indique cuando se publiquen las notas de la convocatoria ordinaria.
- Si un alumno no ha alcanzado una puntuación igual o superior a 5 puntos sobre 10 en el laboratorio (LAB), o ha renunciado a dicha nota según el punto anterior, deberá realizar unas nuevas prácticas de laboratorio cuyo enunciado se publicará con posterioridad a la convocatoria ordinaria. Dichos informes serán calificados y su nota considerada la nota de laboratorio (LAB), siendo necesario que obtenga al menos 5 puntos sobre 10 para superar la asignatura. Además, el alumno debe presentarse al examen escrito (EXA) de la convocatoria extraordinaria y obtener en él una nota igual o superior a 5 puntos sobre 10 para superar la asignatura.
- En ningún caso se mantiene la nota alcanzada en la convocatoria ordinaria en el examen escrito (EXA) para la convocatoria extraordinaria.

## **8. Consideraciones finales**

---

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