



Guía docente de la asignatura

Asignatura	DESARROLLO DE APLICACIONES DISTRIBUIDAS		
Materia	INGENIERÍA DE REDES, SISTEMAS Y SERVICIOS TELEMÁTICOS		
Módulo	MATERIAS ESPECÍFICAS DE TECNOLOGÍAS DE TELECOMUNICACIÓN		
Titulación	GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN		
Plan	460	Código	45020
Periodo de impartición	1er CUATRIMESTRE	Tipo/Carácter	OBLIGATORIA
Nivel/Ciclo	GRADO	Curso	3º
Créditos ECTS	6 ECTS		
Lengua en que se imparte	CASTELLANO		
Profesor/es responsable/s	Jaime Gómez Gil		
Datos de contacto (E-mail, teléfono...)	TELÉFONO: 983 185556 E-MAIL: <a href="mailto:jgomez@tel.uva.es">jgomez@tel.uva.es</a>		
Horario de tutorías	Ver Tutorías del grado de Tecnologías en <a href="http://www.uva.es/export/sites/uva/2.docencia/2.01.grados/2.01.02.ofertaformativagrados/2.01.02.01.alfabetica/Grado-en-Ingenieria-de-Tecnologias-de-Telecomunicacion/">http://www.uva.es/export/sites/uva/2.docencia/2.01.grados/2.01.02.ofertaformativagrados/2.01.02.01.alfabetica/Grado-en-Ingenieria-de-Tecnologias-de-Telecomunicacion/</a> El profesor permite concertar tutorías fuera del horario de tutorías previo correo electrónico a <a href="mailto:jgomez@tel.uva.es">jgomez@tel.uva.es</a> . El profesor pide a los alumnos que envíen un correo electrónico al profesor cuando vayan a asistir a una tutoría, ya sea en el horario oficial, o fuera del horario oficial.		
Departamento	TEORÍA DE LA SEÑAL Y COMUNICACIONES E INGENIERÍA TELEMÁTICA		



## 1. Situación / Sentido de la Asignatura

### 1.1 Contextualización

Los ordenadores son capaces de ejecutar aplicaciones que ejecutan en un único ordenador, y que no hacen uso de redes. A estas aplicaciones se las denomina aplicaciones tradicionales o aplicaciones monolíticas. La aparición y generalización de las redes de ordenadores ha dado lugar a la aparición de un nuevo tipo de aplicaciones, denominadas aplicaciones distribuidas. Estas aplicaciones se caracterizan por estar constituidas por varias partes, que se ejecutan en ordenadores diferentes, y que se comunican entre sí a través de una red, normalmente Internet. Ejemplos de aplicaciones distribuidas son, entre otros, (i) el correo electrónico, (ii) la navegación Web, (iii) las aplicaciones para la comunicación a través de Internet mediante texto, sonido o video, (iv) la compartición de ficheros de datos, sonido o vídeo mediante servidores, y (v) la compartición de ficheros en redes entre iguales P2P (*Peer-to-peer*).

El desarrollo de una aplicación distribuida es normalmente más complejo que el desarrollo de una aplicación tradicional, pues en la aplicación tradicional sólo hay que implementar la lógica de la aplicación, mientras que en la aplicación distribuida hay que implementar la lógica de la aplicación y la comunicación entre las partes que forman la aplicación. La comunicación entre las partes que forman la aplicación se implementa haciendo uso de APIs (*Application Programming Interfaces*), que pueden ser de bajo nivel de abstracción, como el API Sockets de Java, de nivel medio de abstracción, como el API RMI de Java, o de alto nivel de abstracción, como el API JavaBeans de Java.

En esta asignatura (i) se analizarán las principales diferencias entre aplicaciones tradicionales y distribuidas; (ii) se presentarán los diferentes paradigmas o modelos para el desarrollo de aplicaciones distribuidas; (iii) se desarrollarán aplicaciones distribuidas mediante Sockets Java, y (iv) se desarrollarán aplicaciones distribuidas mediante Java RMI.



## 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 primer cuatrimestre del primer curso y en la asignatura de "Ingeniería de Sistemas Software" de la materia "Fundamentos de Sistemas Software" del "Bloque de Materias Básicas de Telecomunicaciones" que se imparte en el segundo cuatrimestre del segundo curso. En dichas asignaturas se proporcionarán los conceptos básicos de programación y desarrollo de sistemas software que facilitarán al alumno la comprensión de otros mostrados en la presente asignatura.

Además, también se utilizarán en esta asignatura conceptos relativos a la capa de transporte y red de la arquitectura de protocolos TCP/IP que han sido introducidos en la asignatura "Arquitectura de Redes, Sistemas y Servicios" de la materia "Fundamentos de Protocolos, Redes y Servicios Telemáticos" del "Bloque de Materias Básicas de Telecomunicaciones", y que se imparte en el primer cuatrimestre del segundo curso.

## 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 las asignaturas de "Programación", "Ingeniería de Sistemas Software", y "Arquitectura de Redes, Sistemas y Servicios".



## 2. Competencias

---

### 2.1 Generales

---

- 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. 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

---

- T7 Conocimiento y utilización de los fundamentos de la programación en redes, sistemas y servicios de telecomunicación
- T13. Capacidad de diferenciar los conceptos de redes de acceso y transporte, redes de conmutación de circuitos y de paquetes, redes fijas y móviles, así como los sistemas y aplicaciones de red distribuidos, servicios de voz, datos, audio, vídeo y servicios interactivos y multimedia.
- TEL6. Capacidad de diseñar arquitecturas de redes y servicios telemáticos.
- TEL7. Capacidad de programación de servicios y aplicaciones telemáticas, en red y distribuidas.



### 3. Objetivos

Al finalizar la asignatura el alumno deberá ser capaz de:

- Describir los objetivos que se persiguen en el desarrollo de aplicaciones distribuidas.
- Comprender la problemática específica asociada al desarrollo de aplicaciones en distribuidas.
- Comprender los conceptos relacionados con el *middleware* como arquitectura básica para el desarrollo de aplicaciones distribuidas.
- Conocer las técnicas básicas sobre las que se basan las aplicaciones distribuidas.
- Diseñar, desarrollar y desplegar aplicaciones distribuidas utilizando una API de acceso a los servicios de transporte de datos.
- Diseñar, desarrollar y desplegar aplicaciones distribuidas utilizando tecnologías basadas en *middleware*.

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

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	30	Estudio y trabajo autónomo individual	45
Clases prácticas de aula (A)	0	Estudio y trabajo autónomo grupal	45
Laboratorios (L)	30		
Prácticas externas, clínicas o de campo	0		
Seminarios (S)	0		
Tutorías grupales (TG)			
Evaluación (fuera del periodo oficial de exámenes)	0		
<b>Total presencial</b>	<b>60</b>	<b>Total no presencial</b>	<b>90</b>



## 5. Bloques temáticos

### Bloque 1: Desarrollo de aplicaciones distribuidas

Carga de trabajo en créditos ECTS:

#### a. Contextualización y justificación

Véase el apartado 1.1.

#### b. Objetivos de aprendizaje

Véase el apartado 3.

#### c. Contenidos

El temario a impartir en las clases teórico-prácticas es el siguiente:

##### TEMA 1: Introducción a la computación distribuida

- Definiciones
- Historia de la computación distribuida
- Diferentes formas de computación
- Virtudes y limitaciones de la computación distribuida
- Conceptos básicos de sistemas operativos
- Conceptos básicos de redes
- Conceptos básicos de ingeniería del software

##### TEMA 2: IPC-Comunicación entre procesos

- Un arquetipo de interfaz de programación para comunicación entre procesos
- Sincronización de eventos
- Temporizadores e hilos de ejecución
- Interbloqueos y temporizadores
- Representación y codificación de datos
- Protocolos basados en texto y protocolos de solicitud-respuesta
- Diagramas de eventos y de secuencia
- Comunicación entre procesos orientada y no orientada a conexión
- Evolución de los paradigmas de comunicación entre procesos

##### TEMA 3: Paradigmas de computación distribuida

- Paradigmas y abstracción
- Paradigmas para aplicaciones distribuidas
- Comparativa



#### **TEMA 4: El API de *sockets***

Antecedentes

La metáfora de *sockets* en la comunicación entre procesos

El API de *sockets* datagrama

*Sockets* con operaciones de E/S no bloqueantes

El API de *sockets* seguros

#### **TEMA 5: Objetos distribuidos-Java RMI**

Paso de mensajes frente a objetos distribuidos

Una arquitectura típica de objetos distribuidos

Sistemas de objetos distribuidos

Llamadas a procedimientos remotos (RPC)

Invocaciones a Métodos Remotos (RMI)

La arquitectura de Java RMI

El API de Java RMI

Una aplicación RMI de ejemplo

Pasos para construir una aplicación RMI

Comparación entre RMI y el API de *Sockets*

#### **TEMA 6: Nombres en sistemas distribuidos**

Nombres, identificadores y direcciones

Nombres planos

Tablas HASH distribuidas

Nombres estructurados

#### **TEMA 7: Sincronización en sistemas distribuidos**

Relojes físicos

Sincronización de relojes físicos

Relojes lógicos

Relojes Vectoriales

#### **TEMA 8: Tolerancia a fallos en sistemas distribuidos**

Disponibilidad, confiabilidad, seguridad, mantenimiento y otros conceptos básicos

Modelos de fallas

Disfrazado de fallas por redundancia

Enmascaramiento de fallas y replicación

Acuerdos en sistemas defectuosos

El problema del acuerdo bizantino

Estos temas se impartirán en el orden que resulte más conveniente para la realización de forma simultánea de las prácticas de la asignatura.



## Guía docente de la asignatura

---

En el laboratorio de la asignatura se llevarán a cabo **casos prácticos guiados** como:

- Desarrollo en Java del juego “hundir la flota”.
- Desarrollo de un “servidor de hora” con sockets Java.
- Desarrollo de un “chat” con sockets Java.
- Desarrollo de una “calculadora remota” con Java RMI.
- Desarrollo de un “chat con interfaz gráfico” y Java RMI.
- Desarrollo del “juego buscaminas” con interfaz gráfico.

### d. Métodos docentes

---

- Clase magistral participativa.
- Taller de casos prácticos guiados en el laboratorio.

### e. Plan de trabajo

---

Véase el Anexo I.

### f. Evaluación

---

La evaluación de la adquisición de competencias se basará en:

- Una prueba escrita al final del cuatrimestre en la que se evaluarán los conocimientos adquiridos tanto en las clases teórico-prácticas como en el laboratorio. Denominaremos **nota E** al resultado de la calificación de esta prueba escrita.
- Evaluaciones del trabajo del alumno en el laboratorio, con la que se obtendrá una **nota L** asociada al trabajo del alumno en el laboratorio. En esta evaluación se tendrán en cuenta:
  - Los informes escritos del trabajo realizado por el alumno, que supondrán un 25% del valor de la nota L.
  - Las respuestas de los alumnos a las preguntas planteadas por el profesor en el laboratorio, que supondrán un 60% del valor de la nota L.
  - La actitud y proactividad del alumno en el laboratorio, que supondrán un 15% del valor de la nota L.

### g. Bibliografía básica

---

Los siguientes son libros básicos que se consideran adecuados para la adquisición de las competencias asociadas a la asignatura por parte del alumno.

- M. L. Liu. Distributed Computing, principles and applications. Prentice-Hall, 2004.
- A. S. Tananbaum & M. V. Steen. Distributed Systems: Principles and Paradigms, 2nd ed. Pearson-Prentice Hall, 2007.
- P. Deitel & H. Deitel. Java how to program, 10th ed. Prentice-Hall, 2015.
- Java API – Oracle Help Center. <https://docs.oracle.com/javase/7/docs/api/>





#### h. Bibliografía complementaria

Los siguientes libros resultan útiles para algunas partes concretas de la asignatura.

- K. Sierra & B. Bates. *Head First Java*, 2<sup>nd</sup> ed. O'Reil, 2005.
- G. F. Coulouris, J. Dollimore & T. Kindberg. *Distributed systems: concepts and design*, 5<sup>th</sup> ed. Addison-Wesley, 2012.
- W. Emmerich. *Engineering distributed objects*. John Wiley & Sons, 2000.
- B. Eckel. *Piensa en Java* 4<sup>a</sup> edición, Pearson 2007.

#### i. Recursos necesarios

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.

### 6. Temporalización (por bloques temáticos)

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1: Desarrollo de aplicaciones distribuidas	6 ECTS	Semanas 1 a 15



## 7. Sistema de calificaciones

La nota final de los alumnos en la asignatura viene dada por la expresión

$$Nota\ final = \begin{cases} E & \text{si } E < 5 \\ \text{máximo} \{E, E \cdot 0.6 + L \cdot 0.4\} & \text{si } E \geq 5 \end{cases} \quad (1)$$

siendo:

**E** la nota que al alumno obtiene en el examen final de la asignatura,

**L** la nota que el alumno obtiene en las evaluaciones del trabajo en el laboratorio.

La nota **E** se caracteriza por ser la nota asociada al examen final de la asignatura, el cual está asociado tanto a las clases teórico-prácticas como al trabajo en el laboratorio. Este examen se realiza en el día y hora fijado por el centro para la realización del mismo, y consta de tres partes. En la primera parte de este examen no se permite a los alumnos emplear documentación, en la segunda parte se permite a los alumnos usar un formulario manuscrito de tamaño A4, y en la tercera parte se permite a los alumnos que empleen toda la documentación que deseen. Las partes se realizan una a continuación de otra, con descansos de 10 minutos entre ellas. La duración de cada una de las partes es típicamente una hora. Cada una de las tres partes contribuye normalmente en un porcentaje similar al valor **E**. Tal y como se deriva de la Expresión (1), los alumnos tienen obtener un valor mínimo de  $E=5$  para aprobar la asignatura.

La nota **L** del laboratorio se compone de (i) la calificación de la actitud del alumno en el laboratorio, ponderada 10%, (ii) la calificación de la evaluación de los informes que el alumno realiza en su trabajo en el laboratorio, ponderada al 30%, y (iii) la calificación de la evaluación de la presentación que los alumnos hacen de su trabajo en el laboratorio al finalizar algunas de las jornadas del mismo, ponderada al 60%. Tal y como se deriva de la Expresión (1), los alumnos pueden no asistir al laboratorio de la asignatura.

No se mantiene ninguna nota de un curso académico para el siguiente.

## 8. Consideraciones finales

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