



Asignatura	ARQUITECTURAS DE APLICACIONES DISTRIBUIDAS		
Materia	ARQUITECTURAS Y TECNOLOGÍAS PARA APLICACIONES DISTRIBUIDAS		
Módulo	MATERIAS ESPECÍFICAS DE LA MENCIÓN EN TELEMÁTICA		
Titulación	GRADO EN INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS DE TELECOMUNICACIÓN – MENCIÓN EN TELEMÁTICA		
Plan	512	Código	46661
Periodo de impartición	2º CUATRIMESTRE	Tipo/Carácter	OPTATIVA (OBLIGATORIA DE LA MENCIÓN)
Nivel/Ciclo	GRADO	Curso	3º
Créditos ECTS	6 ECTS		
Lengua en que se imparte	CASTELLANO		
Profesor/es responsable/s	MANUEL RODRÍGUEZ CAYETANO		
Datos de contacto (E-mail, teléfono...)	TELÉFONO: 983 423000 ext. 5541 E-MAIL: manuel.rodriguez@tel.uva.es		
Departamento	TEORÍA DE LA SEÑAL Y COMUNICACIONES E INGENIERÍA TELEMÁTICA		



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 *enterprise java beans*, *.Net* o *servicios web*.

La situación actual con redes ubicuas y de relativamente altas prestaciones ha dado lugar a un gran número de aplicaciones que necesitan usar una red de comunicaciones para proporcionar las funciones para las que han sido diseñadas. Por este motivo, resulta necesario en la actualidad formar profesionales que comprendan los principales problemas inherentes al desarrollo de aplicaciones en red y distribuidas y conozcan las principales tecnologías disponibles para desarrollar este tipo de aplicaciones de forma eficiente.

En esta asignatura se analizarán las principales diferencias entre aplicaciones centralizadas, en red y distribuidas, se describirán los principales tipos de *middleware* existentes (orientado a objetos, orientado a componentes y orientado a servicios) y se explicará e ilustrará con ejemplos el proceso de desarrollo de aplicaciones en red y distribuidas siguiendo distintas tecnologías ampliamente usadas en la actualidad y que están basadas en los distintos tipos de *middleware* previamente presentados.



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", 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

- 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.
- GB5. Conocimiento de materias básicas, científicas y tecnologías, que le capacite para el aprendizaje de nuevos métodos y tecnologías.
- 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

- T2. 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.
- T3. Capacidad para utilizar herramientas informáticas de búsqueda de recursos bibliográficos o de información relacionada con las telecomunicaciones y la electrónica.
- 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:
- Comprender la problemática específica del desarrollo de aplicaciones en red.
- Comprender los conceptos relacionados con el *middleware* como arquitectura básica para el desarrollo de aplicaciones distribuidas.
- Describir y comparar los distintos tipos de *middleware*.
- Analizar las principales soluciones existentes en el ámbito del *middleware* orientado a objetos, a componentes y a servicios.
- Diseñar, desarrollar y desplegar aplicaciones en red utilizando una API de acceso a los servicios de transporte de datos.
- Diseñar, desarrollar y desplegar aplicaciones distribuidas utilizando distintas tecnologías basadas en *middleware*.



4. Contenidos y/o bloques temáticos

Bloque 1: Arquitectura de aplicaciones distribuidas

Carga de trabajo en créditos ECTS: 6

a. Contextualización y justificación

Véase el apartado 1.1.

b. Objetivos de aprendizaje

Véase el apartado 3.

c. Contenidos

TEMA 1: Introducción a los sistemas distribuidos y el middleware

- 1.1 Conceptos básicos: sistemas centralizados, redes de ordenadores y sistemas distribuidos
- 1.2 Requisitos para los sistemas distribuidos
- 1.3 Tipos de transparencia en los sistemas distribuidos
- 1.4 Tipos de software distribuido:
 - 1.4.1 Aplicaciones en red
 - 1.4.2 Software para sistemas operativos distribuidos
 - 1.4.3 Aplicaciones basadas en *middleware*
- 1.5 Ejemplo de desarrollo de una aplicación telemática en red basada en APIs de acceso a los servicios de transporte de datos
- 1.6 Resumen

TEMA 2: Middleware orientado a objetos

- 2.1 Características comunes
- 2.2 Sun/Oracle: Java RMI
- 2.3 OMG: CORBA
- 2.4 Microsoft: COM/DCOM
- 2.5 Ejemplo de desarrollo de una aplicación telemática distribuida orientada a objetos
- 2.6 Resumen

TEMA 3: Middleware orientado a componentes

- 3.1 Características comunes
- 3.2 Sun/Oracle: J2EE / EJB
- 3.3 OMG: CORBA/CCM
- 3.4 Microsoft: .Net
- 3.5 Ejemplo de desarrollo de una aplicación telemática distribuida orientada a componentes
- 3.6 Resumen

TEMA 4: Middleware orientado a servicios

- 4.1 Características comunes



- 4.2 Servicios web
- 4.3 Ejemplo de desarrollo de una aplicación telemática distribuida orientada a servicios
- 4.4 Resumen

d. Métodos docentes

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

e. Plan de trabajo

Véase el Anexo I.

f. Evaluación

La evaluación de la adquisición de competencias se basará en:

- Prueba escrita al final del cuatrimestre
- Funcionamiento y estructura de los programas desarrollados en el laboratorio por el alumno, así como la documentación asociada a dichos programas
- Respuestas a las preguntas planteadas en los enunciados de prácticas

g Material docente

g.1 Bibliografía básica

- G. F. Coulouris, J. Dollimore, y T. Kindberg, Distributed systems: concepts and design, 5a ed. Addison Wesley. Madrid, 2012
- R. W. Stevens, Unix Network Programming. Networking APIs: sockets and XTI, 2a ed. Prentice-Hall, 1998.
- W. Emmerich, Engineering distributed objects. John Wiley & Sons, 2001.
- C. Szyperski, Component software: beyond object oriented programming. Addison Wesley, 1999.
- G. Alonso, F. Casati, H. Kuno, y V. Machiraju, Web Services: Concepts, Architectures and Applications. Springer, 2004.
- L. Richardson, S. Ruby, y D. H. Hansson, RESTful web services. O'Reilly, 2007.

g.2 Bibliografía complementaria

- J. Pritchard, COM and CORBA side by side : architectures, strategies, and implementations. Addison-Wesley, 1999.
- Oracle: Java Platform, Standard Edition 8 API Specification. 2019. <http://docs.oracle.com/javase/8/docs/api/>
- Oracle: Java Platform, Enterprise Edition (Java EE). Documentation home page. <http://www.oracle.com/technetwork/java/javaee/documentation/index.html>
- Oracle: The java tutorials: RMI. 2017. <http://docs.oracle.com/javase/tutorial/rmi/index.html>
- Oracle: Dynamic code downloading using java RMI. 2019. <http://docs.oracle.com/javase/8/docs/technotes/guides/rmi/codebase.html>



- Oracle: Java SE documentation: Java Remote Method Invocation (Java RMI). 2019. <http://docs.oracle.com/javase/8/docs/technotes/guides/rmi/index.html>
- Oracle: Java SE documentation: Java RMI over IIOP. 2019. <http://docs.oracle.com/javase/8/docs/technotes/guides/rmi-iiop/index.html>
- The java tutorials: Java Naming and Directory Interface. 2017. <http://docs.oracle.com/javase/tutorial/jndi/index.html>
- Oracle: Java SE Documentation. Java IDL (CORBA). 2019. <http://docs.oracle.com/javase/8/docs/technotes/guides/idl/>
- Microsoft Component Object Model. 2018. <https://docs.microsoft.com/en-us/windows/desktop/com/the-component-object-model>
- R. P. Sriganesh, G. Brose, y M. Silverman, Mastering Enterprise JavaBeans. Wiley, 2006.
- Object Management Group: Tutorial sobre CCM, 2003. http://www.omg.org/news/meetings/workshops/RT_2003_Manual/Tutorials/T3_CCM_Wang-Rodrigues.pdf
- Microsoft: Microsoft .NET, 2019. <http://www.microsoft.com/net>
- Codeguru: The .NET Architecture, 2015. http://www.codeguru.com/csharp/sample_chapter/article.php/c8245
- W3C: Web of services. 2015. <http://www.w3.org/standards/webofservices/>
- W3C: Web Services Activity. 2011. <http://www.w3.org/2002/ws>

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

h. 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 un ordenador por alumno, para las sesiones de laboratorio. Cada ordenador contará con un entorno de desarrollo de aplicaciones para los sistemas operativos para los que se desarrollen las prácticas.
- Documentación de apoyo.
- Acceso al material bibliográfico y los recursos telemáticos complementarios recomendados, al menos a parte del mismo, en la biblioteca del Campus o mediante otras vías como Internet.

En esta asignatura se utilizan herramientas docentes online para la docencia y la evaluación. En caso de un transcurso normal de la docencia, estarán disponibles las aulas informáticas del centro. En caso de una afección por medidas sanitarias especiales, el alumno debe contar con medios informáticos y telemáticos suficientes para interactuar con el Campus Virtual y con los sistemas de videoconferencia, así como para realizar los entregables propuestos en el laboratorio de la asignatura.

i. Temporalización



CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
6 ECTS	Semanas 1 a 13

5. Métodos docentes y principios metodológicos

- Clase magistral participativa: Se utilizará esta técnica expositiva para presentar los contenidos de la asignatura.
- Taller de prácticas en el laboratorio: Es esencial en el ámbito tecnológico. Cuando se plantea una práctica o proyecto, es necesario pasar por una serie ordenada y lógica de fases necesarias para obtener un producto o servicio tecnológico capaz de satisfacer una necesidad planteada por un cliente o de resolver un problema técnico.



6. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES o PRESENCIALES A DISTANCIA ⁽¹⁾	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	22	Estudio y trabajo autónomo individual (lectura de documentación, preparación de las prácticas)	49
Clases prácticas de aula (A)	0	Estudio y trabajo autónomo grupal (preparación de las prácticas, diseño y desarrollo de programas)	49
Laboratorios (L)	26		
Prácticas externas, clínicas o de campo	0		
Seminarios (S)	4		
Tutorías grupales (TG)			
Evaluación (fuera del periodo oficial de exámenes)	0		
Total presencial	52	Total no presencial	98
TOTAL presencial + no presencial			150

(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

Criterio: cuando al menos el 50% de los días lectivos del cuatrimestre transcurran en normalidad, se asumirán como criterios de evaluación los indicados en la guía docente. Se recomienda la evaluación continua ya que implica minimizar los cambios en la adenda.

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Revisión del funcionamiento de los programas desarrollados y su documentación asociada y evaluación de los Informes de prácticas de laboratorio (LAB)	50%	Es condición necesaria (pero no suficiente) para superar la asignatura alcanzar una calificación igual o superior a 5 sobre 10 puntos en la nota total de laboratorio (informes+funcionamiento de los programas).
Examen final escrito (EXA)	50%	Es condición necesaria (pero no suficiente) para superar la asignatura alcanzar una calificación igual o superior a 5 sobre 10 puntos en este examen.



CRITERIOS DE CALIFICACIÓN

- **Convocatoria ordinaria:**
 - o 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 puntos sobre 10. Por ejemplo, si obtiene un 4 sobre 10 en la nota total de laboratorio (no supera los requisitos mínimos) y un 6 sobre 10 en el examen de teoría (que supondría una nota final de 5 sobre 10 según la ponderación dada en la tabla si no se tuviesen en cuenta los requisitos mínimos), la nota final de la asignatura será de 4,5 sobre 10 puntos.
 - o En caso de no cumplir los requisitos mínimos en ninguna de las dos partes, la nota final de la asignatura será la suma ponderada (según los pesos indicados en la tabla) de las dos notas.
- **Convocatoria extraordinaria:**
 - o Se mantiene la calificación obtenida en cada instrumento de la tabla siempre que dicha calificación sea igual o superior a 5 sobre 10 puntos.
 - o El alumno deberá realizar de nuevo el examen final escrito si la nota obtenida es inferior a 5 sobre 10 puntos.
 - o 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.

8. Consideraciones finales

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



La adenda debe reflejar las adaptaciones sobre cómo se desarrollaría la formación si tuviese que ser desarrollada en modalidad online por mandato de autoridades competentes. Se deben conservar los horarios de asignaturas y tutorías publicados en la web de la UVa, indicar el método de contacto y suministrar un tiempo razonable de respuesta a las peticiones de tutoría (2-4 días lectivos). Describir el modo en que se desarrollarán las actividades prácticas. En el caso de TFG/TFM, desarrollar detalladamente los sistemas de tutorías y tutela de los trabajos.

A4. Contenidos y/o bloques temáticos

Bloque 1: Arquitectura de aplicaciones distribuidas

Carga de trabajo en créditos ECTS:

c. Contenidos Adaptados a formación online

Los contenidos se impartirán de forma íntegra.

d. Métodos docentes online

- Presentación de contenidos teóricos y resolución de dudas sobre los contenidos teóricos mediante el uso del sistema de videoconferencia Cisco Webex Meetings o BlackBoard Collaborate. También se proporcionarán ejemplos resueltos.
- Realización de prácticas guiadas a distancia.
- Resolución de dudas de las prácticas o proyecto apoyada por el sistema de videoconferencia Cisco Webex Meeting o BlackBoard Collaborate y por foros específicos en el Campus Virtual. Uso de vídeos para proporcionar aclaraciones, si fuese necesario.

e. Plan de trabajo online

El plan de trabajo no sufrirá alteraciones, respetándose tanto el contenido de las diferentes sesiones propuestas como la temporización.

f. Evaluación online

La evaluación de la adquisición de competencias se basará en:

- Prueba on-line al final del cuatrimestre
- Funcionamiento y estructura de los programas desarrollados en el laboratorio por el alumno, así como la documentación asociada a dichos programas
- Respuestas a las preguntas planteadas en los enunciados de prácticas

i. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
6 ECTS	Semanas 1 a 13

A5. Métodos docentes y principios metodológicos

- **Taller de prácticas en el laboratorio:** Es esencial en el ámbito tecnológico. Cuando se plantea una práctica o proyecto, es necesario pasar por una serie ordenada y lógica de fases necesarias para obtener un producto o servicio tecnológico capaz de satisfacer una necesidad planteada por un cliente o de resolver un problema técnico.
- **Videoconferencia** (sistema de videoconferencia Cisco Webex Meeting o BlackBoard Collaborate): Para la resolución de dudas sobre los contenidos de la asignatura y sobre la realización de la práctica o proyecto. Es un complemento idóneo, puesto que proporciona la cercanía de la que carecen otros recursos y métodos como las audio-guías y los vídeos. No obstante, el alumno no tendrá obligación de encender la cámara si no lo considera oportuno, para que su privacidad se vea respetada en todo momento.

A6. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES A DISTANCIA ⁽²⁾	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	22	Estudio y trabajo autónomo individual (lectura de documentación, preparación de las prácticas)	49
Clases prácticas de aula (A)	0	Estudio y trabajo autónomo grupal (preparación de las prácticas, diseño y desarrollo de programas)	49
Laboratorios (L)	26		
Prácticas externas, clínicas o de campo	0		
Seminarios (S)	4		
Tutorías grupales (TG)			
Evaluación (fuera del periodo oficial de exámenes)	0		
Total presencial	52	Total no presencial	98
Total presencial a distancia + no presencial			150

⁽²⁾ Actividad presencial a distancia en este contexto es aquella en que el grupo sigue por videoconferencia la clase impartida por el profesor en el horario publicado para la asignatura.

A7. Sistema y características de la evaluación

Criterio: cuando más del 50% de los días lectivos del cuatrimestre transcurran en situación de contingencia, se asumirán como criterios de evaluación los indicados en la adenda.

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Revisión del funcionamiento de los programas desarrollados y su documentación asociada y evaluación de los Informes de prácticas de laboratorio (LAB)	50%	Es condición necesaria (pero no suficiente) para superar la asignatura alcanzar una calificación igual o superior a 5 sobre 10 puntos en la nota total de laboratorio (funcionamiento de los programas+documentación+informes).
Examen final en línea (EXA)	50%	Es condición necesaria (pero no suficiente) para



		superar la asignatura alcanzar una calificación igual o superior a 5 sobre 10 puntos en este examen.
--	--	--

CRITERIOS DE CALIFICACIÓN

- **Convocatoria ordinaria:**
 - o 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 puntos sobre 10. Por ejemplo, si obtiene un 4 sobre 10 en la nota total de laboratorio (no supera los requisitos mínimos) y un 6 sobre 10 en el examen de teoría (que supondría una nota final de 5 sobre 10 según la ponderación dada en la tabla si no se tuviesen en cuenta los requisitos mínimos), la nota final de la asignatura será de 4,5 sobre 10 puntos.
 - o En caso de no cumplir los requisitos mínimos en ninguna de las dos partes, la nota final de la asignatura será la suma ponderada (según los pesos indicados en la tabla) de las dos notas.
- **Convocatoria extraordinaria:**
 - o Se mantiene la calificación obtenida en cada instrumento de la tabla siempre que dicha calificación sea igual o superior a 5 sobre 10 puntos.
 - o El alumno deberá realizar de nuevo el examen final (EXA) si la nota obtenida es inferior a 5 sobre 10 puntos.
 - o 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.