



Proyecto/Guía docente de la asignatura

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		
Fecha de revisión por el Comité de Título	27 de junio 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*.

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:

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 RPC y gRPC
- 1.7 Resumen

TEMA 2: Middleware orientado a objetos

- 2.1 Características comunes
- 2.2 Sun: 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: 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 tradicionales (*big web services*)
- 4.3 Servicios web restful
- 4.4 Ejemplo de desarrollo de una aplicación telemática distribuida orientada a servicios
- 4.5 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

Ver página oficial de la Uva para la bibliografía recomendada para asignaturas (Leganto):

https://buc-uva.alma.exlibrisgroup.com/leganto/public/34BUC_UVA/lists/5041841150005774?auth=SAML

h. Recursos necesarios

- Documentación de apoyo
- 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. Cada ordenador contará con el entorno de desarrollo para java SE y EE.

i. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
6 ECTS	Semanas 1 a 15

5. Métodos docentes y principios metodológicos

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

6. Tabla de dedicación del estudiantado a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	25	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)	5		
Tutorías grupales (TG)			
Evaluación (fuera del periodo oficial de exámenes)	0		
Total presencial	60	Total no presencial	90

7. Sistema de calificaciones – Tabla resumen

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Informes de prácticas de laboratorio y revisión del código y funcionamiento de los programas desarrollados	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, código y funcionamiento de los programas). Es obligatorio presentar todas las prácticas para poder superar esta parte. En caso de no hacerlo, la nota obtenida será un No presentado.
Examen final escrito	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:**
 - 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. 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 anterior) de las dos notas.
- **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 en él en la convocatoria ordinaria es inferior a 5 sobre 10 puntos.
- 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 en la convocatoria ordinaria es inferior a 5 sobre 10 puntos.

(*) Se entiende por convocatoria extraordinaria la segunda convocatoria.

Art 35.4 del ROA 35.4. La participación en la convocatoria extraordinaria no quedará sujeta a la asistencia a clase ni a la presencia en pruebas anteriores, salvo en los casos de prácticas externas, laboratorios u otras actividades cuya evaluación no fuera posible sin la previa realización de las mencionadas pruebas.

<https://secretariageneral.uva.es/wp-content/uploads/VII.2.-Reglamento-de-Ordenacion-Academica.pdf>

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 o código 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 y similares.

8. Consideraciones finales

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