

**Guía docente de la asignatura**

Asignatura	SISTEMAS DISTRIBUIDOS		
Materia	ARQUITECTURA DE COMPUTADORAS, SISTEMAS OPERATIVOS Y SISTEMAS DISTRIBUIDOS		
Módulo	COMUNES A LA INFORMÁTICA		
Titulación	GRADO EN INGENIERÍA INFORMÁTICA (545)		
Plan	545	Código	46916
Periodo de impartición	2º CUATRIMESTRE	Tipo/Carácter	COMPLEMENTOS DE INFORMÁTICA
Nivel/Ciclo	GRADO	Curso	2º
Créditos ECTS	6 ECTS		
Lengua en que se imparte	ESPAÑOL		
Profesor responsable	CÉSAR LLAMAS BELLO		
Datos de contacto	TELÉFONO: 983 423000 ext. 5610 E-MAIL: cllamas@infor.uva.es		
Horario de tutorías	Véase www.uva.es → Centros → Campus de Valladolid → Escuela de Ingeniería Informática → Tutorías		
Departamento	INFORMÁTICA (ATC, CCIA Y LSI)		

1. Situación / Sentido de la Asignatura

Hoy en día es difícil encontrar algún sistema real que no ofrezca la posibilidad de comunicarse y compartir recursos entre usuarios, simultáneamente, y en ubicaciones geográficas diversas. Por ello, los diseñadores de software y los expertos en TI deben conocer y dominar las técnicas que hacen esto posible. Durante las últimas décadas esta disciplina de la informática ha venido en llamarse “Sistemas Distribuidos” (“Distributed Computing”, en inglés), y su cuerpo de doctrina incluye aspectos variados que tienen que ver desde el estudio de las plataformas hardware y redes que lo hacen posible, hasta la investigación sobre los modelos de negocio sobre los que se implantan este tipo de sistemas. Entre medias encontramos teoría y práctica de algoritmos distribuidos, sistemas operativos distribuidos, algoritmos distribuidos, middleware y plataformas para la distribución y arquitecturas software en relación con la distribución.

1.1 Contextualización

Esta asignatura se encuadra en la materia que denominamos en este grado “Entorno Tecnológico”, junto a los Sistemas Operativos, la Arquitectura y Organización de Computadoras, y la Administración y Evaluación de los Sistemas Informáticos.

1.2 Relación con otras materias

En consecuencia con la importancia del estudio de este tipo de sistemas, es lógico que dentro de este grado la noción de sistema distribuido aparezca en las diversas materias de la formación general y específica. En consonancia con ello, esta asignatura aborda un enfoque centrado en el diseño del software que hace posible la construcción de los sistemas distribuidos, a través del estudio de los diversos paradigmas y arquitecturas que los describen, desde los ya clásicos sistemas cliente-servidor hasta llegar a aspectos tan actuales como la construcción de software orientada a servicios.

1.3 Prerrequisitos

Para un óptimo aprovechamiento de la asignatura, se recomienda a los alumnos un nivel suficiente en las competencias alcanzadas en la materia de “Fundamentos Básicos de la Informática”, especialmente en los contemplados específicamente en las asignaturas de “Fundamentos de Programación” y “Fundamentos de Redes de Computadoras”. Asimismo, también se recomienda un especial cuidado en las competencias logradas en las asignaturas de “Fundamentos de Ingeniería de Software” y en general asignaturas de programación dentro de la materia de “Entornos de Software”.

2. Competencias

2.1 Generales

Código	Descripción
CG02	Conocimientos básicos de la profesión
CG16	Capacidad de aplicar los conocimientos en la práctica.
CG18	Capacidad de aprender.
CG21	Habilidad para trabajar de forma autónoma.

2.2 Específicas

Código	Descripción
CI11	Conocimiento y aplicación de las características, funcionalidades y estructura de los Sistemas Distribuidos, las Redes de Computadores e Internet y diseñar e implementar aplicaciones basadas en ellas.
CI14	Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.
IS04	Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.

3. Objetivos

Código	Descripción
CI 11.1	Comprender la estructura y funcionamiento de las diversas variantes de sistemas distribuidos y saberlas aplicar en la caracterización de los mismos.
C11.2	Entender la naturaleza, organización y función del middleware de distribución y usarlo para el desarrollo de aplicaciones.
C11.3	Conocer los diversos entornos de desarrollo disponibles y saber emplear alguno de ellos en el desarrollo de aplicaciones sencillas.
CI14.1	Comprender y saber aplicar modelos de programación para sistemas distribuidos.
IS04.1	Comprender los fundamentos de las arquitecturas orientadas a servicios y el papel que juegan en el desarrollo de aplicaciones distribuidas, sobre la base de ejemplos concretos.

4. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	30h-3h 1 = 27h	Estudio y trabajo autónomo individual	60h
Laboratorios (L)	22h-2h·2 = 26h	Estudio y trabajo autónomo grupal	30h
Evaluación (fuera del periodo oficial de exámenes)	3h·1 + 2h·2 = 7h		
Total presencial	60h	Total no presencial	90h

5. Bloques temáticos

Conforme a la memoria de verificación del grado la asignatura comprende los siguientes temas: (a) Arquitectura y caracterización de los sistemas distribuidos; (b) Modelos de programación para sistemas distribuidos. Middleware; (c) Plataformas de desarrollo para sistemas distribuidos; (d) Arquitecturas orientadas a servicios. Esto se consigue en la presente guía mediante dos bloques de conocimientos:

I. *Arquitectura y caracterización de los sistemas distribuidos.*

II. *Modelos fundamentales de diseño de sistemas distribuidos.*

En ellos se refleja de la mejor manera posible la unidad conceptual de la asignatura y se tratan de conseguir los diversos objetivos mediante unidades de carácter teórico-práctico. En el primer bloque se estudian conceptos básicos de sistemas distribuidos, las principales metas de diseño y las aproximaciones arquitectónicas más importantes. En el segundo, se detallan algunos de los modelos más importantes que se emplean en el diseño de los sistemas distribuidos, y que dan lugar a las distintas técnicas de software sobre los que se apoyan. El

método de desarrollo elegido para esta asignatura sigue una aproximación por condensación con una mezcla entre la aproximación “abstracciones-primero”, y “de abajo-arriba”, revisando continuamente los conceptos ya impartidos para converger a las competencias indicadas en estos bloques temáticos, por ello no son bloques temáticos consecutivos, sino que contienen actividades que se entremezclan temporalmente.

Bloque 1: Plataformas Distribuidas: Herramientas y Técnicas.

Carga de trabajo en créditos ECTS:

2,5

a. Contextualización y justificación

Los sistemas distribuidos se construyen para dar solución a problemas de dispersión geográfica, comunicación, coordinación, escalado, seguridad y tolerancia frente a fallos. Por ello existen soluciones muy diversas que dan lugar a sistemas muy distintos. En este bloque se describen los conceptos teóricos básicos de los sistemas distribuidos, y las arquitecturas más utilizadas.

b. Objetivos de aprendizaje

- CI11.1 Comprender la estructura y funcionamiento de las diversas variantes de sistemas distribuidos y saberlas aplicar en la caracterización de los mismos.
- CI14.1 Comprender y saber aplicar modelos de programación para sistemas distribuidos.
- IS04.1 Comprender los fundamentos de las arquitecturas orientadas a servicios y el papel que juegan en el desarrollo de aplicaciones distribuidas, sobre la base de ejemplos concretos.

c. Contenidos

- 1.1 Arquitectura y caracterización de los sistemas distribuidos.
 - 1.1.1 Conceptos básicos. Desafíos de diseño de los sistemas distribuidos (A.1).
 - 1.1.2 Protocolos para la comunicación de aplicaciones (A.4).
 - 1.1.4 Seguridad en comunicaciones. Seguridad en sockets e invocación remota (Laboratorio 5).
- 1.2 Arquitectura cliente-servidor.
 - 1.2.1 Arquitectura Cliente-Servidor. Arquitecturas orientadas a servicios y servicios Web. (A.7).
 - 1.2.2 Programación con sockets de una aplicación C-S (Laboratorio 2)
 - 1.2.3 Programación de una aplicación C-S mediante JavaRMI (Laboratorio 4).
 - 1.2.4 Técnicas de servidores web en diversos lenguajes (Seminario-laboratorio 2).
- 1.3 Arquitectura P2P.
 - 1.3.1 Arquitectura P2P (A.8).
 - 1.3.2 Técnicas de comunicación en grupo con JavaRMI (Laboratorio 4).

d. Métodos docentes

Para el desarrollo de este bloque temático se utilizarán los recursos didácticos que ofrecen las (a) Clases magistrales participativas, (b) tutorías activas, (c) laboratorios de resolución de problemas.

Véase el Punto 8 de la presente guía “Anexo: Métodos docentes”.

e. Plan de trabajo

Véase el Punto 8 de la presente guía “Anexo: Métodos docentes”.

f. Evaluación

Véase el punto 7 de la presente guía “Sistema de calificaciones y Tabla resumen”.

g. Bibliografía básica

- Liu, M. L. “Computación Distribuida. Fundamentos y Aplicaciones”. Addison-Wesley. 2004.
- Coulouris, G., Dollimore, J., Kindberg, T. “Sistemas Distribuidos. Conceptos y Diseño, 3ª Ed.”. Addison Wesley, 2001.

h. Bibliografía complementaria

- Tanenbaum, A.S. “Sistemas Distribuidos: Principios y Paradigmas”, Prentice-Hall México, 2008.
- Sinha, P.K. “Distributed Operating Systems. Concepts and Design”. IEEE Press, 1997.
- Arnold, Gosling y Holmes. “El Lenguaje de Programación JAVA. 3ª Ed.”. Addison-Wesley, 2001.
- Farley, J. “Java Distributed Computing”. O’Reilly, 1997.

i. Recursos necesarios

Los recursos contienen una bibliografía básica que deberá ser la base de estudio primaria, comprende libros básicos en español aunque se recomienda encarecidamente que se consulte en su versión en inglés con el fin de aprender los términos originales usados en el resto del mundo. Se indica también una bibliografía de carácter auxiliar con libros más especializados. No parece conveniente añadir a esta lista la innumerable cantidad de recursos y páginas web disponibles en Internet. En la página de la asignatura se irá detallando en cada actividad la bibliografía secundaria más pertinente.

A estos recursos se suman las plataformas de gestión de contenidos como Moodle, en su versión del Campus Virtual de la Uva como en la versión de la ETSIInf, que se utilizará para todo aquello que implique alguna actividad de comunicación, sincronización y depósito de materiales del profesor hacia los alumnos, y viceversa. Como medio material de laboratorio se utilizarán los medios informáticos de la ETSI Informática, donde podrán realizar los encargos y tareas de cada unidad. Estos recursos incluyen internet, servidores de aplicaciones y plataformas de desarrollo de software.

1. Plataforma de apoyo a la docencia “Moodle”. En ella se depositarán las diapositivas de apoyo, cuestionarios, depósitos de tareas, foros, wikis, agendas y blogs, así como enlaces a material de interés para cada unidad.
2. Aulas de ordenadores conectadas a Internet. Es completamente necesario que *cada alumno disponga de un ordenador individual integrado administrado por la Escuela de Informática para desarrollar las lecciones y cuestionarios*, pues estos son evaluables y sólo en esas condiciones es posible comprobar y hacer cumplir las normas de evaluación y realización de pruebas individuales que dan validez a las calificaciones de la Universidad de Valladolid. En caso de no poder cumplirse este requisito, los objetivos prácticos de laboratorio carecen de validez alguna, y sería preciso cambiar la metodología de la presente guía docente.
3. Software de desarrollo para la realización de prácticas de laboratorio: Java SDK, Netbeans con Tomcat y Glassfish.

Bloque 2: Modelos fundamentales de diseño de sistemas distribuidosCarga de trabajo en créditos ECTS: **a. Contextualización y justificación**

El diseño de los sistemas distribuidos requiere una comprensión de los modelos existentes, más allá de su comprensión teórica.

b. Objetivos de aprendizaje

- CI11.2 Entender la naturaleza, organización y función del middleware de distribución y usarlo para el desarrollo de aplicaciones.
- CI11.3 Conocer los diversos entornos de desarrollo disponibles y saber emplear alguno de ellos en el desarrollo de aplicaciones sencillas.
- CI14.1 Comprender y saber aplicar modelos de programación para sistemas distribuidos.
- IS04.1 Comprender los fundamentos de las arquitecturas orientadas a servicios y el papel que juegan en el desarrollo de aplicaciones distribuidas, sobre la base de ejemplos concretos.

c. Contenidos

- 2.1 Modelos fundamentales de diseño de sistemas distribuidos.
 - 2.1.1 Modelos fundamentales del diseño de los sistemas distribuidos (A.5).
 - 2.1.2 Protocolos para la comunicación de aplicaciones. (A.4)
- 2.2 Comunicación entre paso de mensajes.
 - 2.2.1 Comunicación por paso de mensajes mediante sockets (A.2).
 - 2.2.2 Programación de aplicaciones con sockets (Laboratorio 1).
 - 2.2.3 Comunicación entre procesos (A.3).
 - 2.3.4 Comunicación de aplicaciones mediante sockets UDP (Seminario-laboratorio 1).
- 2.3 Comunicación entre objetos remotos.
 - 2.3.1 Middleware para la invocación remota entre aplicaciones (A.6).
 - 2.3.2 Programación mediante Java RMI.
 - 2.3.3 Patrones de programación distribuida en Java RMI.
 - 2.3.2 Programación de una aplicación C-S mediante JavaRMI (L.4).
- 2.4 Comunicación mediante colas de mensajes.
 - 2.4.1 Comunicación mediante colas de mensajes (A.9).
 - 2.4.2 Comunicación mediante colas de mensajes en Java (Laboratorio 6).

d. Métodos docentes

Para el desarrollo de este bloque temático se utilizarán los recursos didácticos que ofrecen las (a) Clases magistrales participativas, (b) tutorías activas, (c) laboratorios de resolución de problemas.

Véase el Punto 8 de la presente guía "Anexo: Métodos docentes".

e. Plan de trabajo

Véase el Punto 8 de la presente guía “Anexo: Métodos docentes”.

f. Evaluación

Véase el punto 7 de la presente guía “Sistema de calificaciones y Tabla resumen”.

g. Bibliografía básica

- Liu, M. L. “Computación Distribuida. Fundamentos y Aplicaciones”. Addison-Wesley. 2004.
- Coulouris, G., Dollimore, J., Kindberg, T. “Sistemas Distribuidos. Conceptos y Diseño, 3ª Ed.”. Addison Wesley, 2001.

h. Bibliografía complementaria

- Tanenbaum, A.S. “Sistemas Distribuidos: Principios y Paradigmas”, Prentice-Hall México, 2008.
- Sinha, P.K. “Distributed Operating Systems. Concepts and Design”. IEEE Press, 1997.
- Arnold, Gosling y Holmes. “El Lenguaje de Programación JAVA. 3ª Ed.”. Addison-Wesley, 2001.
- Farley, J. “Java Distributed Computing”. O’Reilly, 1997.

i. Recursos necesarios

Véase el punto “i” del bloque temático anterior.

6. Temporalización (por bloques temáticos)

La planificación siguiente se estima en términos de 15 semanas netas, que forman el grueso de la docencia del cuatrimestre:

Bloque 1: Arquitectura y caracterización de los sistemas distribuidos. / 2,5 ECTS

- Lección A.1: Conceptos básicos y desafíos de diseño de los sistemas distribuidos.
- Laboratorio 2: Programación con sockets de una aplicación C-S.
- Lección A.4: Protocolos para la comunicación de aplicaciones.
- Lección A.7: Arquitectura Cliente-Servidor.
- Lección A.8: Arquitectura P2P
- Laboratorio 5: Seguridad de la comunicación.
- Seminario-laboratorio 2: Técnicas de servidores web en diversos lenguajes.

Bloque 2: Modelos fundamentales de diseño de sistemas distribuidos / 3,5 ECTS

- Lección A.2: Comunicación por paso de mensajes mediante sockets.
- Laboratorio 1: Programación de aplicaciones con sockets.
- Lección A.3: Comunicación entre procesos.
- Seminario-laboratorio 1: Comunicación de aplicaciones mediante sockets UDP.
- Lección A.5: Modelos fundamentales de diseño de los SS.DD.

- Laboratorio 3: Programación mediante Java RMI.
- Lección A.6: Middleware para la invocación remota.
- Laboratorio 4: Patrones de programación distribuida en Java RMI.
- Lección A.9: Comunicación mediante colas de mensajes.
- Laboratorio 6: Comunicación mediante colas de mensajes.

7. Sistema de calificaciones – Tabla resumen

La evaluación de los resultados del aprendizaje se realizará conforme se realizan las actividades de la asignatura. Las actividades de evaluación son: Test de preparación en la tutoría activa grupal, cuestionario de desarrollo de los guiones de prácticas, informe de trabajo en común y presentación de resultados de los seminarios y una prueba de test final. Además se valorará la participación del alumno en las actividades presenciales.

Estas actividades, y en lo sucesivo el sistema de calificaciones, se sostienen en la medida en que sea posible asegurar el trabajo individual en función de la disponibilidad de laboratorio y puestos de trabajo donde pueda asegurarse el control de la actividad por parte del profesor que imparta las tutorías activas, el laboratorio o el seminario práctico. Si esto no fuera posible, el mecanismo de evaluación pasaría a ser el habitual de examen global, manteniéndose los porcentajes, pero concentrándose el proceso de evaluación en una sola prueba.

Los test de preparación de la tutoría activa grupal son pruebas objetivas de elección múltiple, y tienen como objetivo evaluar el progreso del estudio individual y permiten carencias de aprendizaje. Cada unidad (lección) de bloques teóricos se evalúa en una prueba independiente. También tienen como objetivo preparar la prueba teórica final de la asignatura con preguntas del mismo tipo y nivel. No es preciso superar estas pruebas para superar satisfactoriamente la asignatura, en ningún modo, aun así contribuyen con un 15% al valor final de la nota, promoviendo el trabajo continuo y progresivo. Tal y como se ha indicado previamente, siempre que se permita asegurar la individualidad del trabajo del alumno.

Las prácticas se desarrollan en torno a guiones bien diseñados (Lecciones) para conducir al alumno en las competencias necesarias, dentro de este guion, el alumno encontrará desafíos y preguntas relacionadas con los laboratorios que permitirán que el alumno obtenga una mejor comprensión de la práctica de los sistemas distribuidos. Estos guiones se acompañan de unos guiones previos (Mini-Lecciones preparatorias) que el alumno podrá cumplimentar en horas de laboratorio y en casa, y donde se allana el camino para que los guiones de laboratorio se completen satisfactoriamente. Es necesario superar satisfactoriamente el 75% de las prácticas para superar la asignatura en la convocatoria ordinaria. Tal y como se ha indicado previamente, siempre que la disponibilidad de recursos permitan asegurar la actividad del alumno.

Se programan también dos seminarios/laboratorios calificables en dos sesiones adicionales, que se valoran con un 15% de la calificación final. No hay un umbral de nota mínimo para esta parte de la asignatura, aunque sí es preciso haber realizado al menos un seminario de los dos. Tal y como se ha indicado previamente, siempre que la disponibilidad de recursos permitan asegurar la actividad del alumno.

Finalmente, hay un examen global que se compone de pruebas por cada unidad de teoría, de modo análogo a las tutorías activas. Hay que superar cada una de las pruebas de cada unidad para superar la convocatoria ordinaria. La contribución de esta prueba a la nota final es de un 40 por ciento. Esta prueba se realizará en el lugar y hora fijado para el examen por el centro.

La participación activa del alumno en las clases y demás actividades del curso podrá tenerse en cuenta con hasta un 10 % "suplementario" que se sumará a la nota final, truncándose en caso que supere la nota máxima

posible. Para poder optar a esta puntuación extra el alumno deberá participar en al menos el 75 por ciento de las actividades de la asignatura, y no es una condición necesaria para obtener la calificación máxima.

La siguiente tabla resume las consideraciones anteriores.

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Cuestionarios de preparación de la tutoría activa grupal. Una prueba por cada lección teórica, agrupando dos o más pruebas en una sesión.	15%	3 o cuatro sesiones de tutorías activas tras las lecciones teóricas correspondientes. Asistencia no obligatoria. El peso en esta nota se reparte proporcionalmente.
2 trabajos de seminario práctico (laboratorio) a realizar en el laboratorio.	15%	Hay que realizar al menos 1 trabajo práctico de entre los dos. El peso en la nota se reparte a partes iguales.
Sesiones de laboratorio.	30%	Las sesiones evaluables de laboratorios (no las sesiones preparatorias de laboratorios) son importantes para superar la asignatura en la convocatoria ordinaria, y es preciso superar al menos un 75% de ellos. La asistencia es obligatoria.
Examen final escrito (tipo prueba objetiva) sobre los conceptos teóricos de la asignatura. Una prueba por cada bloque del mismo modo que en los cuestionarios de tutorías activas.	40%	Es condición indispensable para superar la asignatura, obtener un resultado positivo en esta prueba, por cada unidad teórica (lección).

Puesto que el Reglamento de Ordenación Académica vigente requiere fijar condiciones de calificación distintas para las dos convocatorias de examen, en el siguiente cuadro se detallan los criterios de calificación correspondientes para ambas:

CRITERIOS DE CALIFICACIÓN
<ul style="list-style-type: none"> • Convocatoria ordinaria: <ul style="list-style-type: none"> ○ Superar la prueba final de contenidos teóricos por cada una de las unidades (lecciones) teóricas. ○ Completar los objetivos de al menos un 75% de los laboratorios evaluables, cumpliendo las competencias mínimas de la asignatura. ○ El alumno debe conseguir al menos un 50% de la calificación mediante la acumulación de los dos criterios anteriores y los resultados mostrados en el resto de instrumentos de verificación de capacidades (seminarios y evaluaciones de tutoría activa.). ○ El resto de instrumentos de evaluación, se ponderan correspondientemente para cubrir el 30% restante de la calificación. ○ En el caso de poder garantizarse los recursos suficientes como para realizar el control y seguimiento de la actividad y trabajo de los alumnos mediante laboratorios individuales, tutorías activas y seminarios prácticos, se concentrará todo el proceso evaluador en dos controles de teoría y práctica del mismo modo en el que se detalla a continuación en la convocatoria extraordinaria. • Convocatoria extraordinaria: <ul style="list-style-type: none"> ○ Superar la prueba final de contenidos teóricos por cada una de las unidades teóricas (lecciones) de la asignatura. Este instrumento contribuye con un 55% a la nota final de esta convocatoria. ○ Aquellos alumnos que hayan superado todos los bloques de la evaluación teórica tal y como se especifica en la convocatoria ordinaria, no precisan realizar esta evaluación teórica extraordinaria, conservando la nota de la convocatoria ordinaria. ○ Superar una prueba de laboratorio donde se fijan competencias prácticas centrales a la asignatura. Este instrumento contribuye con un 45% a la nota final de esta convocatoria. ○ Aquellos alumnos que hayan superado las condiciones prácticas de la asignatura fijadas en la convocatoria ordinaria, no precisan realizar esta última prueba de laboratorio conservando la nota que tuvieran en esa parte.

8. Anexo: Métodos docentes

Actividad	Metodología
Clase de teoría	<ul style="list-style-type: none">• Clase magistral participativa, donde el alumno dispone anticipadamente de los temas a discutir.• Estudio de casos en aula• Resolución de problemas
Clase práctica	<ul style="list-style-type: none">• Mini-laboratorios: sesiones previas de laboratorio siguiendo guiones elaborados por el profesor, y que servirán de introducción a las sesiones evaluables.• Laboratorios: sesiones de laboratorio evaluables siguiendo guiones elaborados por el profesor.
Laboratorios Teórico prácticos (seminarios)	<ul style="list-style-type: none">• Talleres de presentación de contenidos teórico de marcado carácter experimental.• Se realiza un trabajo individual guiado por material teórico práctico y cuestionarios.
Tutoría activa.	<ul style="list-style-type: none">• Cuestionarios de evaluación de competencias teóricas.• Se sigue una sesión de revisión de dificultades y enfoque de los problemas presentados en los cuestionarios.

9. Anexo: Cronograma de actividades previstas

El plan de trabajo propuesto para el desarrollo de asignatura es el siguiente:

(se ha indicado especialmente con [**♦ Presencia**] aquellas actividades donde es necesario que el alumno realice presencialmente las actividades para poder ser calificado positivamente) La planificación siguiente se estima en términos de 15 semanas netas, que forman el grueso de la docencia del cuatrimestre:

Semana	Actividades a realizar
1	<ul style="list-style-type: none">Presentación de la asignatura.Lección A.1 / Conceptos básicos. Desafíos de diseño de los sistemas distribuidos. (1ª parte)Presentación de los objetivos del laboratorio.
2	<ul style="list-style-type: none">Lección A.1 / Conceptos básicos. Desafíos de diseño de los sistemas distribuidos. (2ª parte).Preparación del Laboratorio 1: Programación de una aplicación con sockets.
3	<ul style="list-style-type: none">Lección A.2 / Comunicación por paso de mensajes mediante sockets. Servidores multitenhebrados.[♦ Presencia] Laboratorio 1: Programación de una aplicación con sockets.
4	<ul style="list-style-type: none">Tutoría Activa 1 sobre las lecciones A1 y A2.A.3 / Comunicación entre procesos. (1ª parte)Preparación del Laboratorio 2: Programación con sockets para una aplicación C-S.
5	<ul style="list-style-type: none">Lección A.3 / Comunicación entre procesos. (2ª parte).[♦ Presencia] Laboratorio 2: Programación con sockets para una aplicación C-S..
6	<ul style="list-style-type: none">Lección A.4 / Protocolos para la comunicación de aplicaciones.[♦ Presencia] Seminario 1: Comunicación con sockets UDP..
7	<ul style="list-style-type: none">[♦ Presencia] Tutoría Activa 2 sobre las lecciones A3 y A4.Lección A.5 / Modelos fundamentales de diseño de los SS.DD. (1ª parte)Preparación del Laboratorio 3: Programación mediante Java RMI.
8	<ul style="list-style-type: none">Lección A.5 / Modelos fundamentales de diseño de los SS.DD. (2ª parte)[♦ Presencia] Laboratorio 3: Programación mediante Java RMI.
9	<ul style="list-style-type: none">A.6 / Middleware para la invocación remota.Preparación del Laboratorio 4: Patrones de programación distribuida mediante Java RMI.
10	<ul style="list-style-type: none">[♦ Presencia] Tutoría Activa 3 sobre las lecciones A5 y A6.A.7 / Arquitectura Cliente-Servidor. (1ª parte)[♦ Presencia] Laboratorio 4: Patrones de programación distribuida mediante Java RMI
11	<ul style="list-style-type: none">A.7 / Arquitectura Cliente-Servidor. (2ª parte)Preparación del laboratorio 5: Seguridad de la comunicación.



12	<ul style="list-style-type: none">• Lección A.8 / Arquitectura P2P (1ª parte)• [♦ <i>Presencia</i>] Laboratorio 5: Seguridad de la comunicación.
13	<ul style="list-style-type: none">• A.8 / Arquitectura P2P (2ª parte)• Preparación del laboratorio 6 sobre comunicación mediante colas de mensajes.
14	<ul style="list-style-type: none">• Tema A.9 / Comunicación mediante colas de mensajes.• [♦ <i>Presencia</i>] Laboratorio 6: Comunicación mediante colas de mensajes.
15	<ul style="list-style-type: none">• [♦ <i>Presencia</i>] Tutoría Activa 4 sobre los temas A7, A8 y A9.• Recapitulación de la asignatura.• [♦ <i>Presencia</i>] Seminario 2: Técnicas de servidores web en diversos lenguajes..