

**Guía docente de la asignatura**

<b>Asignatura</b>	SISTEMAS DISTRIBUIDOS		
<b>Materia</b>	ENTORNO TECNOLÓGICO		
<b>Módulo</b>			
<b>Titulación</b>	GRADO EN INGENIERÍA INFORMÁTICA (463) GRADO EN INGENIERÍA INFORMÁTICA DE SISTEMAS (464)		
<b>Plan</b>	463   464	<b>Código</b>	45199   45256
<b>Periodo de impartición</b>	1 <sup>er</sup> . CUATRIMESTRE	<b>Tipo/Carácter</b>	OBLIGATORIA
<b>Nivel/Ciclo</b>	GRADO	<b>Curso</b>	2º
<b>Créditos ECTS</b>	6 ECTS		
<b>Lengua en que se imparte</b>	CASTELLANO		
<b>Profesor/es responsable/s</b>	CÉSAR LLAMAS BELLO (Coordinador) JAVIER DAVID FERNÁNDEZ GARCÍA NOEMI MOYA ALONSO		
<b>Datos de contacto (E-mail, teléfono...)</b>	TELÉFONO: 983 423000 ext. 5610 E-MAIL: cllamas@infor.uva.es		
<b>Horario de tutorías</b>	Véase <a href="http://www.uva.es">www.uva.es</a> → Centros → Campus de Valladolid → Escuela Técnica Superior de Ingeniería Informática → Tutorías		
<b>Departamento</b>	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

Las competencias que abarca esta asignatura pertenecen al bloque de conceptos básicos de la ingeniería, y también incluye competencias de índole general que son parte importante de la formación del ingeniero en informática.

### 2.1 Generales

Código	Descripción
G02	Conocimientos básicos de la profesión
G03	Capacidad de análisis y síntesis
G04	Capacidad de organizar y planificar
G11	Capacidad crítica y autocrítica
G16	Capacidad de aplicar los conocimientos en la práctica
G18	Capacidad de aprender
G21	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
CI11.1	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

Los contenidos y profundidad de la materia se diseñan para que el alumno invierta un total de 6 ECTS, que en total resultan unas 150 horas con la intención de que el esfuerzo se realice de la forma más uniforme posible a lo largo del cuatrimestre. En las directrices de la Universidad de Valladolid y la Memoria de Verificación del Título de Ingeniería en Informática, se establece una composición del 40 por ciento de presencialidad de dichas horas lo que redunda en 60 horas de actividad presencial que se desglosan de la siguiente forma:

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORA S
Clases teórico-prácticas (T/M)	30	Estudio y trabajo autónomo individual	60
Clases prácticas de aula (A)		Estudio y trabajo autónomo grupal	30
Laboratorios (L)	16		
Prácticas externas, clínicas o de campo			
Seminarios (S)	4		
Tutorías grupales (TG)	8		
Evaluación (fuera del periodo oficial de exámenes)	2		
<b>Total presencial</b>	<b>60</b>	<b>Total no presencial</b>	<b>90</b>

## 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: Arquitectura y caracterización de los sistemas distribuidos.

Carga de trabajo en créditos ECTS:

#### 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

Código	Descripción
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 Coordinación y acuerdo distribuido (A.11).
  - 1.1.3 Replicación y tolerancia a fallos (A.12).
  - 1.1.4 Seguridad en comunicaciones. Seguridad en sockets e invocación remota (L.7).
- 1.2 **Arquitectura cliente-servidor.**
  - 1.2.1 Arquitectura Cliente-Servidor. Arquitecturas orientadas a servicios y servicios Web. (A.7).



- 1.2.2 Diseño y programación de un protocolo sencillo para la comunicación C-S (L.3).
- 1.2.1 Diseño e implementación de un protocolo sencillo para C-S basado en Java RMI (L.6).
- 1.3 **Arquitectura P2P.**
- 1.3.1 Arquitectura P2P (A.8).
- 1.3.2 Técnicas de comunicación en grupo con JavaRMI (L.5).

---

#### **d. Métodos docentes**

---

Para el desarrollo de este bloque temático se utilizarán los siguientes recursos didácticos:

- Clase magistral participativa.
- Tutoría activa.
- Resolución de problemas.
- Aprendizaje colaborativo.

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 – Tabla resumen".

---

#### **g. Bibliografía básica**

---

Si se conoce el ISBN o los códigos de referencia de la biblioteca, inclúylos. Ejemplo:

- 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** para desarrollar las lecciones y cuestionarios, pues estos son evaluables.
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 distribuidos.**Carga 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, conociéndolos en la práctica.

**b. Objetivos de aprendizaje**

Código	Descripción
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.

**c. Contenidos****2.1 Modelos fundamentales de diseño de sistemas distribuidos.**

- 2.1.1 Protocolos para la comunicación entre aplicaciones (A.4).
- 2.1.2 Modelos fundamentales del diseño de los sistemas distribuidos (A.5).
- 2.1.3 Control transaccional de recursos (A.9).

**2.2 Comunicación entre paso de mensajes.**

- 2.2.1 Comunicación por paso de mensajes mediante sockets. Servidores multitenhebrados (A.2).
- 2.2.2 Comunicación entre procesos (A.3).
- 2.2.3 Programación con sockets de una aplicación C-S (L.2).
- 2.2.4 Diseño y programación de un protocolo sencillo para la comunicación C-S (L.3).

**2.3 Comunicación entre objetos remotos.**

- 2.3.1 Middleware para la invocación remota entre aplicaciones (A.6).
- 2.3.2 Programación de una aplicación C-S mediante JavaRMI (L.4).

**2.4 Comunicación mediante colas de mensajes.**

- 2.4.1 Middleware para la comunicación mediante colas de mensajes (A.10).
- 2.4.2 Programación de una aplicación mediante colas de mensajes (L.8).

**d. Métodos docentes**

Para el desarrollo de este bloque temático se utilizarán los siguientes recursos didácticos:

- Clase magistral participativa.
- Tutoría activa.
- Resolución de problemas.
- Aprendizaje colaborativo.

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 – 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 1 del bloque temático 1 (bloque anterior).

## 6. Temporalización (por bloques temáticos)

El número de semanas de un cuatrimestre son 15. Ejemplo:

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1: Arquitectura y caracterización de los sistemas distribuidos.	2,7 ECTS	
A.1 Conceptos básicos. Desafíos de diseño de los sistemas distribuidos.	(4h)	Semana 1 y 2
A.7 Arquitectura Cliente-Servidor y arquitecturas orientadas a servicios.	(2h)	Semana 9
A.8 Arquitectura P2P.	(4h)	Semana 10 y 11
A.11 Coordinación y acuerdo distribuido.	(2h)	Semana 14
A.12 Replicación y tolerancia a fallos.	(2h)	Semana 15
L.0 Introducción a las herramientas del laboratorio.	(2h)	Semana 1
L.3 Diseño de un protocolo sencillo para la programación C-S.	(2h)	Semana 5
L.5 Técnicas de comunicación en grupo con Java RMI	(2h)	Semana 9
L.6 Diseño e implementación de un protocolo sencillo para C-S basado en Java RMI	(2h)	Semana 10
L.7 Seguridad en comunicaciones. Seguridad en sockets e invocación remota	(2h)	Semana 12
S.2 Seminario: Sistemas distribuidos reales.	(2h)	Semana 14
Bloque 2: Modelos fundamentales de diseño de sistemas distribuidos	3,3 ECTS	
A.2 Comunicación por paso de mensajes mediante sockets. Servidores multitenhebrados.	(2h)	Semana 3
A.3 Comunicación entre procesos.	(4h)	Semana 4 y 5
A.4 Protocolos para la comunicación de aplicaciones mediante sockets.	(2h)	Semana 6
A.5 Modelos fundamentales de diseño de los SS.DD.	(4h)	Semana 7
A.6 Middleware para la invocación remota de aplicaciones.	(2h)	Semana 8
A.9 Control transaccional de recursos.	(2h)	Semana 12
A.10 Comunicación mediante colas de mensajes.	(2h)	Semana 13
L.1 Programación de una aplicación con sockets	(2h)	Semana 2
L.2 Programación con sockets para una aplicación C-S.	(2h)	Semana 4
L.4 Programación de una aplicación C-S mediante Java RMI.	(2h)	Semana 6
L.8 Implementación de un S.D. Mediante colas de mensajes.	(2h)	Semana 13
S.1 Seminario: Modelos de distribución avanzados.	(2h)	Semana 8

## 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 de 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.

Los test de preparación de la tutoría activa grupal tienen como objetivo servir de mecanismo de evaluación del progreso del estudio individual y nos permiten enfocar la asignatura dinámicamente sobre aquellos puntos en los que se detecta un resultado más pobre. Dichos test también tienen como meta preparar al alumno para la prueba teórica final de la asignatura con preguntas de elección múltiple 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, promoviéndose el trabajo continuo y progresivo.

Las prácticas se desarrollan en torno a guiones bien diseñados (Lecciones) para conducir al alumno en las competencias necesarias, dentro de este guión, 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 (MiniLecciones) que el alumno podrá cumplimentar en su 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, y como es obvio esta actividad es presencial y se realizará individualmente en el laboratorio dispuesto por el centro en su horario respectivo. Por si esto fuera poco, en el caso de que el alumno no haya superado satisfactoriamente una lección en una sesión, podrá repetirla en la siguiente sesión de laboratorio haciendo uso de parte del tiempo correspondiente al tiempo de la sesión siguiente. En este caso, se consigna la media aritmética con la nota de la sesión anterior. La primera sesión de laboratorio no es evaluable y se encamina a la familiarización con la dinámica del laboratorio, y la última sesión es potestativa y no cuenta en el 75%, aunque sí en la nota.

La evaluación y consiguiente presentación del trabajo en seminarios públicos contribuye con un 15 por ciento al valor final de la nota del alumno, para obtener un resultado positivo en este apartado, el alumno deberá realizar la presentación y defensa de su trabajo. La nota de esta parte es la media aritmética de la calificación de la documentación y la calificación de la presentación. Ambos seminarios cuentan igual, y contribuyen con un 15% al peso de la nota final del alumno. 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.

La prueba final constará aproximadamente de 50 preguntas breves y de tipo test, sobre toda la asignatura. Es preciso superar esta prueba para poder superar satisfactoriamente la asignatura. 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.



INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Cuestionarios de preparación de la tutoría activa grupal (15 minutos).	15,00%	Habrà 4 tutorías activas cada 3 o cuatro semanas.
Entregas y presentación de dos seminarios.	15,00%	Primer seminario sobre la semana 8 y segundo sobre la semana 14.
Laboratorios	30,00%	Cada semana o cada dos semanas dependiendo de seminarios y tutorías.
Examen final escrito	40,00%	Periodo de exámenes

En la siguiente tabla se indican los criterios de calificación para la convocatoria ordinaria y extraordinaria indicando las calificaciones mínimas u otros requisitos necesarios para aprobar la asignatura:

CRITERIOS DE CALIFICACIÓN
<ul style="list-style-type: none"><li>• <b>Convocatoria ordinaria:</b> <b>(Para superar la asignatura deben de cumplirse todas las condiciones siguientes)</b><ul style="list-style-type: none"><li>○ Haber obtenido al menos el 50% de la máxima calificación en el examen final escrito.</li><li>○ Haber superado satisfactoriamente al menos el 75% de los laboratorios evaluables.</li><li>○ Haber presentado y documentado satisfactoriamente al menos 1 seminario.</li><li>○ Obtener al menos una calificación de 5,0.</li></ul></li><li>• <b>Convocatoria extraordinaria:</b><ul style="list-style-type: none"><li>○ Haber obtenido al menos el 50% de la máxima calificación en el examen final escrito.</li><li>○ Haber superado satisfactoriamente al menos el 75% de los laboratorios evaluables.</li><li>○ Haber presentado y documentado satisfactoriamente al menos 1 seminario.</li><li>○ Obtener al menos una calificación de 5,0.</li></ul></li></ul>

## 8. Anexo: Métodos docentes

### • Clase de teoría

*Clase magistral participativa:* donde el instructor presenta los materiales teóricos y estudios de caso. El profesor podrá llevar cuenta del nivel de asistencia y participación y considerarlo positivamente en la calificación final. Existirá un único grupo de teoría. El alumno dispondrá previo a la impartición de aquellos materiales escritos, en red y bibliográficos que precise el alumno, lo que incluye apuntes y diapositiva. Todo ello soportado por las plataformas Moodle de las que dispone el centro (aulas.inf.uva.es) y el Campus Virtual de la Uva (campusvirtual.uva.es).

### • Clase práctica

*Resolución de problemas:* Cada sesión de práctica de laboratorio dispondrá de un guión elaborado siguiendo el patrón de Lección de Moodle, donde se integran pequeñas preguntas y tareas que deberá responder y resolver el alumno a modo de guión de laboratorio. La plataforma Moodle ofrecerá una realimentación en línea de los resultados para favorecer el aprendizaje. La asistencia presencial del alumno es muy importante, pues el tipo de competencias que se adquieren en este tipo de laboratorios no puede adquirirse únicamente mediante el estudio teórico. El profesor resolverá individualmente aquellas dudas que surjan durante la realización. Es importante resaltar que no existen prácticas para entregar, por lo que el alumno podrá superar la parte de laboratorio sin más que realizar in situ cada una de las prácticas. Las prácticas podrán repetirse en la sesión siguiente de laboratorio en el caso de no conseguir resultados positivos.

◦ Previo a la realización de cada laboratorio, se publicará un guión de apoyo (MiniLección) que permitirá que los alumnos realicen pequeñas prácticas en casa para hacer más fácil y corta la sesión de laboratorio, ilustrando los puntos difíciles de la sesión siguiente.

### • Seminarios

*Aprendizaje colaborativo:* En las sesiones de seminario, los alumnos, en grupos de tres o más personas (dependiendo de los tamaños de grupo que fije la Uva), presentarán los resultados del trabajo de síntesis realizado en torno a temas propuestos por el profesor o los propios alumnos. En esta actividad los alumnos deberán desplegar iniciativa y capacidad de organización, búsqueda y síntesis, para conseguir un resultado satisfactorio. Cada grupo podrá aportar un portafolio electrónico que será tutorizado en tiempo de tutoría personalizada a petición de los alumnos y que se concretará en un documento que se entregará mediante la plataforma Moodle en el plazo que se indique. El trabajo será presentado en grupo en sesiones de 15 minutos incluyendo un turno de preguntas, durante las sesiones de seminario de dos horas. Tras ello cada grupo recibirá la realimentación de la opinión de los asistentes. Las rúbricas de evaluación se entregarán como parte del enunciado de los requisitos de los seminarios.

### • Tutoría en grupo

*Tutoría Activa:* Cada sesión de tutoría activa se inicia con una sesión de aproximadamente 15 minutos donde cada alumno recibe un cuestionario de tipo test donde se evalúan los conocimientos de las unidades vistas en las clases magistrales participativas. A continuación se muestra el solucionarlo, que nos permitirá abordar el estudio de los problemas encontrados por los alumnos, y dado el caso, iniciar actividades en grupo para tratar de resolver de modo colaborativo cualquier otro tipo de problemas encontrado en el estudio de la asignatura. Este cuestionario contendrá preguntas del mismo tipo y calado que las que se utilizarán en el examen final. Una vez completados los turnos de tutoría de todos los grupos, se liberarán cuestionarios mediante la plataforma Moodle para que los alumnos, individualmente o en grupo, puedan utilizarlo como mecanismo de autoevaluación.

El tamaño de los grupos de laboratorio adecuados es 20, y de realizarse en el laboratorio puede utilizarse la plataforma Moodle como soporte de los cuestionarios. En caso contrario podría aumentarse el tamaño y realizarse en papel, aunque se desaconseja. Cada tutoría en grupo tendrá una duración aproximada de hora y media.



<b>Actividad</b>	<b>Metodología</b>
<b>Clase de teoría</b>	<ul style="list-style-type: none"><li>• Clase magistral participativa</li><li>• Estudio de casos en aula</li><li>• Resolución de problemas</li></ul>
<b>Clase práctica</b>	<ul style="list-style-type: none"><li>• Guiones de laboratorio preparatorio.</li><li>• Guiones de laboratorio individuales con preguntas de evaluación con realimentación que requieren búsqueda, iniciativa y pequeños desafíos sobre proyectos ya realizados.</li></ul>
<b>Seminarios</b>	<ul style="list-style-type: none"><li>• Talleres de aprendizaje,</li><li>• Baremación entre pares.</li><li>• Trabajo en grupo.</li></ul>
<b>Tutoría</b>	<ul style="list-style-type: none"><li>• Evaluación de los contenidos teóricos y puesta en común de los puntos conflictivos en el seguimiento de la asignatura.</li></ul>

**9. Anexo: Cronograma de actividades previstas**

Véase el punto 6, "Temporización por bloques temáticos".

BLOQUE TEMÁTICO	TIPO	PERIODO PREVISTO DE DESARROLLO
A.1 Conceptos básicos. Desafíos de diseño de los sistemas distribuidos.	A	Semana 1 y 2
L.0 Introducción a las herramientas del laboratorio.	L	Semana 1
L.1 Programación de una aplicación con sockets	L	Semana 2
A.2 Comunicación por paso de mensajes mediante sockets. Servidores multienhebrados.	A	Semana 3
T.1 Tutoría Activa grupal	T	Semana 3
A.3 Comunicación entre procesos.	A	Semana 4 y 5
L.2 Programación con sockets para una aplicación C-S.	L	Semana 4
L.3 Diseño de un protocolo sencillo para la programación C-S.	L	Semana 5
A.4 Protocolos para la comunicación de aplicaciones mediante sockets.	A	Semana 6
L.4 Programación de una aplicación C-S mediante Java RMI.	L	Semana 6
A.5 Modelos fundamentales de diseño de los SS.DD.	A	Semana 7
T.2 Tutoría Activa grupal	T	Semana 7
A.6 Middleware para la invocación remota de aplicaciones.	A	Semana 8
S.1 Seminario: Modelos de distribución avanzados.	S	Semana 8
A.7 Arquitectura Cliente-Servidor y arquitecturas orientadas a servicios.	A	Semana 9
L.5 Técnicas de comunicación en grupo con Java RMI	L	Semana 9
A.8 Arquitectura P2P.	A	Semana 10 y 11
L.6 Diseño e implementación de un protocolo sencillo para C-S basado en Java RMI	L	Semana 10
T.3 Tutoría Activa grupal	T	Semana 11
A.9 Control transaccional de recursos.	A	Semana 12
L.7 Seguridad en comunicaciones. Seguridad en sockets e invocación remota	L	Semana 12
A.10 Comunicación mediante colas de mensajes.	A	Semana 13
L.8 Implementación de un S.D. Mediante colas de mensajes.	L	Semana 13
A.11 Coordinación y acuerdo distribuido.	A	Semana 14
S.2 Seminario: Sistemas distribuidos reales.	S	Semana 14



A.12 Replicación y tolerancia a fallos.	A	Semana 15
T.4 Tutoría Activa grupal	T	Semana 15

A: Lecciones teóricas. T: Tutorías activas grupales. L: Laboratorios. S: Seminarios.