Guía docente de la asignatura

Asignatura	Sistemas Distribuidos		
Materia	Entornos Tecnológicos		
Módulo			
Titulación	Grado en Ingeniería Informática		
Plan	463	**Código**	45199
Periodo de impartición	1º Cuatrimestre	**Tipo/Carácter**	Obligatoria
Nivel/Ciclo	Curso de Adaptación	**Curso**	Único
Créditos ECTS	6		
Lengua en que se imparte	Español		
Profesor/es responsable/s	César Llamas Bello		
Datos de contacto (E-mail, teléfono…)	cllamas@infor.uva.es, 5610		
Horario de tutorías	http://www.uva.es/consultas/guia.php?menu=presentacion&ano_academico=1011&codigo_plan=463&codigo_asignatura=45199&grupo=1		
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

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

G2. Conocimientos básicos de la profesión

G3. Capacidad de análisis y síntesis

G4. 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

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.

IS4. 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

R1. Comprender la estructura y funcionamiento de las diversas variantes de sistemas distribuidos y saberlas aplicar en la caracterización de los mismos.

R2. Comprender y saber aplicar modelos de programación para sistemas distribuidos.

R3. Entender la naturaleza, organización y función del middleware de distribución y usarlo para el desarrollo de aplicaciones.

R4. Conocer los diversos entornos de desarrollo disponibles y saber emplear alguno de ellos en el desarrollo de aplicaciones sencillas.

R5. 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	HORAS
Clases teórico-prácticas (T/M)	30,0 h	Estudio y trabajo autónomo individual	60,0 h
Clases prácticas de aula (A)		Estudio y trabajo autónomo grupal	30,0 h
Laboratorios (L)	17,0 h		
Prácticas externas, clínicas o de campo			
Seminarios (S)	4,0 h		
Tutorías grupales (TG)	7,5 h		
Evaluación	1,5 h		
Total presencial	**60,0 h**	**Total no presencial**	**90,0 h**

5. Bloques temáticos

Esta asignatura se describe someramente en la memoria de verificación del grado en el siguiente conjunto de unidades:

I. Arquitectura y caracterización de los sistemas distribuidos.

II. Modelos de programación para sistemas distribuidos. Middleware.

III. Plataformas de desarrollo para sistemas distribuidos.

IV. Arquitecturas orientadas a servicios.

Si bien esta descripción de materias es útil, y manejable, resulta poco adecuada para su impartición, y resulta más interesante una metodología de lo general a lo particular, donde se desarrollará la asignatura sin perder en ningún momento la perspectiva de los sistemas distribuidos en su conjunto. Con este fin se diseñan dos bloques principales: (I) *Arquitectura y caracterización de los sistemas distribuidos*, y (II) *Modelos fundamentales de diseño de sistemas distribuidos*. Que constan de sub-bloques en los que se detallan convenientemente los objetivos, resultados de aprendizaje, recursos didácticos y carga. Estos subbloques se programan en el tiempo para dar lugar a un desarrollo armónico y constructivo de las habilidades y conocimientos. A continuación se describen los bloques temáticos generales de que consta el desarrollo de la asignatura, y posteriormente los subbloques temáticos detallados.

Bloque 1:Arquitectura y caracterización de los sistemas distribuidos

Carga de trabajo en créditos ECTS: 2,7

a.1. 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.1 Objetivos de aprendizaje

R1. Comprender la estructura y funcionamiento de las diversas variantes de sistemas distribuidos y saberlas aplicar en la caracterización de los mismos.
R2. Comprender y saber aplicar modelos de programación para sistemas distribuidos.
R5. 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.1 Contenidos

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-14).

1.2 Arquitectura cliente-servidor.

1.2.1 Arquitectura Cliente-Servidor (A-7).
1.2.2 Diseño y programación de un protocolo sencillo para la comunicación C-S (L-3).

1.3 Arquitectura P2P.

1.3.1 Arquitectura P2P (A-8).
1.3.2 Programación de una aplicación P2P con JavaRMI (L-5).

1.4 Arquitecturas orientadas a servicios.

1.4.1 Arquitecturas orientadas a servicios, y servicios Web (A-12).

d.1 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.

Se detallarán en concreto los utilizados en cada subbloque temático.

e.1 Plan de trabajo

Tal y como se refleja en la sección C1, se desarrollará el bloque temático en subbloques donde los recursos docentes serán clases magistrales participativas y tutorías activas (A-1, A2, …), partiendo de los conceptos básicos, para abordar arquitecturas concretas mediante ejemplos basados en resolución de problemas (L-1, L-2, …) que tendrán lugar, en su parte presencial, en el laboratorio asignado a la asignatura. Dichos laboratorios dispondrán de herramientas informáticas y se basan en el desarrollo de guiones de prácticas interactivos donde lo importante es que el alumno comprenda y maneje los conceptos y herramientas con el fin de hacerse con las competencias descritas. Las tutorías activas (T-1, T-2,...) son el complemento ideal donde mediante testy preguntas cortas el alumno realiza una evaluación de sus conocimientos e, in situ, se plantean los problemas que surgen de dichas pruebas para los alumnos. En estas tutorías activas, los alumnos tienen ocasión de medir y contrastar sus conocimientos. Finalmente, este bloque temático incluye un seminario (S) donde los alumnos tendrán la oportunidad de mostrar públicamente un trabajo elaborado en grupo sobre un tema relativo a las competencias contenidas en este bloque.

f.1 Evaluación

La evaluación se realizará considerando cada una de las actividades indicadas en el plan de trabajo anterior. Afortunadamente, la plataforma Moodle permite seguir la pista del trabajo y el esfuerzo realizado por el alumno, y se convierte en herramienta de **autoevaluación** y herramienta de evaluación para el profesor. Como puede comprenderse, con un desglose de competencias y objetivos de aprendizaje tan exhaustivos como los que se indican, es preciso un soporte informático adecuado, y Moodle es el vehículo elegido en este caso.

Bloque 2:Modelos fundamentales de diseño de sistemas distribuidos.

Carga de trabajo en créditos ECTS: 2,7

a.2 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.2 Objetivos de aprendizaje

R3. Entender la naturaleza, organización y función del middleware de distribución y usarlo para el desarrollo de aplicaciones.
R4. Conocer los diversos entornos de desarrollo disponibles y saber emplear alguno de ellos en el desarrollo de aplicaciones sencillas.
R5. 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.2 Contenidos

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-13).
2.1.3 Control transaccional de recursos (A-9).
2.1.4 Características básicas del modelo de seguridad (A-13).

2.2 Comunicación entre paso de mensajes.

2.2.1 Comunicación por paso de mensajes mediante sockets. Servidores multienhebrados (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).
2.3.3 Seguridad en sockets e invocación remota (L-8).

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-6).

d.2 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.

Se detallarán en concreto los utilizados en cada subbloque temático.

e.2 Plan de trabajo

Tal y como se refleja en la sección C1, se desarrollará el bloque temático en subbloques donde los recursos docentes serán clases magistrales participativas y tutorías activas (A-1, A2, …), partiendo de los conceptos básicos, para abordar arquitecturas concretas mediante ejemplos basados en resolución de problemas (L-1, L-2, …) que tendrán lugar, en su parte presencial, en el laboratorio asignado a la asignatura. Dichos laboratorios dispondrán de herramientas informáticas y se basan en el desarrollo de guiones de prácticas interactivos donde lo importante es que el alumno comprenda y maneje los conceptos y herramientas con el fin de hacerse con las competencias descritas. Las tutorías activas (T-1, T-2,...) son el complemento ideal donde mediante testy preguntas cortas el alumno realiza una evaluación de sus conocimientos e, in situ, se plantean los problemas que surgen de dichas pruebas para los alumnos. En estas tutorías activas, los alumnos tienen ocasión de medir y contrastar sus conocimientos. Finalmente, este bloque temático incluye un seminario (S) donde los alumnos tendrán la oportunidad de mostrar públicamente un trabajo elaborado en grupo sobre un tema relativo a las competencias contenidas en este bloque.

Como puede deducirse de lo anterior, la presencialidad es un aspecto muy importante, si no esencial, en este plan de trabajo, pues supone la forma correcta y más adecuada de certificar en la práctica a lo largo del tiempo las competencias exigidas, más allá de un simple examen final.

f.2 Evaluación

La evaluación se realizará considerando cada una de las actividades indicadas en el plan de trabajo anterior. Afortunadamente, la plataforma Moodle permite seguir la pista del trabajo y el esfuerzo realizado por el alumno, y se convierte en herramienta de **autoevaluación** y herramienta de evaluación para el profesor. Como puede comprenderse, con un desglose de competencias y objetivos de aprendizaje tan exhaustivos como los que se indican, es preciso un soporte informático adecuado, y Moodle es el vehículo elegido en este caso.

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 que precisará el alumno se componen de la bibliografía básica, que deberá ser consultada frecuentemente por éste, y que consta de los así llamados “libros de cabecera”. Como puede verse en los apartados anteriores, son libros en español, aunque se recomienda encarecidamente que se consulte en su versión de habla inglesa. Esto permitirá aprender los términos originales en las ediciones más modernas. A esta bibliografía básica hay que añadir una bibliografía de carácter auxiliar, que contiene libros especializados más apropiados para ciertos puntos críticos, y de apoyo a la realización de las prácticas. No se ha creído conveniente añadir a esta lista la innumerable cantidad de recursos y páginas web disponibles en Internet. De ello se dará cuenta convenientemente en el desarrollo de las unidades de Aula y Laboratorio.

A estos recursos hay que añadir los que provienen de la utilización de las Tecnologías de la Información, principalmente la plataforma de enseñanza Moodle, que se utilizará de modo necesario para todo aquello que implique alguna actividad de comunicación, sincronización y depósito de materiales del profesor hacia los alumnos, y viceversa.
En el plano material, los alumnos deberán utilizarán continuamente medios informáticos a su disposición a través 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.

- 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.

- 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.
- Software de desarrollo para la realización de prácticas de laboratorio: Java SDK, Netbeans con Tomcat y Glassfish.

DESCRIPCIÓN PORMENORIZADA DE LOS SUBBLOQUES TEMÁTICOS

A-1.	**Conceptos básicos. Desafíos de diseño de sistemas distribuidos.**	
Contenidos		
Se presentan los conceptos básicos con los que centrar el estudio de los sistemas distribuidos, junto a los desafíos que plantean este tipo de problemas en relación con los sistemas centralizados.		
Resultados de aprendizaje		
o Conocer las nociones básicas y la organización de la disciplina que rodea el diseño de aplicaciones distribuidas. o Distinguir qué caracteriza los S.D. del resto, sus puntos fuertes y débiles más importantes. o Apreciar las implicaciones de la distribución en el diseño de las aplicaciones. o Comprender requisitos básicos de diseño de aplicaciones distribuidas.		
Recursos didácticos		
o Clase magistral participativa. o Tutoría activa.		
Carga		
ECTS	**Horas presenciales**	**Horas no presenciales**
0,2	2,5	2,5

A-2.	**Comunicación por paso de mensajes mediante sockets. Servidores multienhebrados.**	
Contenidos		
Se expone el mecanismo básico mediante el cual se soporta la noción de distribuido. Los sockets son el mecanismo más sencillo de nivel de transporte que permite implementar esta idea, junto con la noción de servidor multienhebrados. Esta unidad permite preparar el desarrollo de la unidad L-2 que se realiza en el laboratorio.		
Resultados de aprendizaje		
o Entender las dinámicas de trabajo en sistemas con varios procesos. o Conocer cómo comunicar procesos uno a uno mediante sockets. o Entender la forma en que un servidor puede atender diversos clientes. o Conocer las ventajas y el alcance de la comunicación con sockets.		
Recursos didácticos		
o Clase magistral participativa. o Tutoría activa.		
Carga		
ECTS	**Horas presenciales**	**Horas no presenciales**
0,2	2,5	2,5

A-3.	**Comunicación entre procesos.**	
Contenidos		
La comunicación entre procesos se explica mediante un modelo simplificado que permite plantear las distintas opciones posibles que, además, nos permiten clasificar en varios tipos los sistemas distribuidos. Dicha comunicación entre procesos, como problema tecnológico, plantea diversas dificultades que nos permiten exponer temas tan diversos como la necesidad del middleware y el intercambio de datos en la comunicación entre procesos.		
Resultados de aprendizaje		
o Conocer el modelo de comunicación mediante paso de mensajes. o Entender las diversas posibilidades de sincronización en la comunicación por paso de mensajes. o Asimilar la necesidad de middleware para la comunicación, y las alternativas más populares. o Conocer las alternativas para la representación de datos en la comunicación.		
Recursos didácticos		
o Clase magistral participativa. o Tutoría activa.		

Carga		
ECTS	**Horas presenciales**	**Horas no presenciales**
0,2	2,5	2,5

A-4.	**Protocolos para la comunicación de aplicaciones.**

Contenidos

Los protocolos, entre los cuales los protocolos basados en texto son muy habituales en los sistemas distribuidos, son la piedra angular en el mecanismo de sincronización e intercambio de datos entre los procesos de un sistema distribuido. Aunque, en la práctica, no es común diseñar protocolos completos, sí es muy frecuente tener que entenderlos, para lo cual es preciso tener nociones básicas de los principios de dichos protocolos de diseño. Esta unidad permite preparar el desarrollo de la unidad L-3 que se realiza en el laboratorio.

Resultados de aprendizaje

- Reconocer los elementos de una notación de protocolos.
- Conocer el mecanismo usual para el diseño de un protocolo.
- Apreciar las partes principales de la descripción de un rfc de un protocolo sencillo.

Recursos didácticos

- Clase magistral participativa.
- Tutoría activa.

Carga

ECTS	Horas presenciales	Horas no presenciales
0,2	2,5	2,5

A-5.	**Modelos fundamentales del diseño de los sistemas distribuidos.**

Contenidos

Cualquier diseño de sistema distribuido requiere la especificación, o al menos la adhesión, a ciertos modelos fundamentales cuales son: el modelo de interacción, el modelo de fallos, y el modelo de seguridad. De su comprensión depende la caracterización de alto nivel del sistema distribuido que se trata de comprender o diseñar.

Resultados de aprendizaje

- Comprender la necesidad de establecer ciertos modelos al diseñar un SD.
- Especificar modelos sencillos de interacción.
- Especificar modelos de fallos para componentes del sistema.
- Entender los componentes principales del modelo de seguridad.

Recursos didácticos

- Clase magistral participativa.
- Tutoría activa.

Carga

ECTS	Horas presenciales	Horas no presenciales
0,2	2,5	2,5

A-6.	**Middleware para invocación remota entre aplicaciones.**

Contenidos

La invocación remota entre objetos remotos es uno de los mecanismos de diseño de aplicaciones más utilizado. Este tipo de herramienta conceptual requiere la disponibilidad de cierto middleware que permita la realización de este tipo de construcciones de programación. En esta unidad se plantean las bases de un middleware de este tipo como es JavaRMI, que con tanto éxito se usa, junto a CORBA. Esta unidad permite preparar el desarrollo de la unidad L-4 que se realiza en el laboratorio.

Resultados de aprendizaje

- Conocer los elementos software del middleware RMI en Java.
- Conocer el procedimiento estándar para la obtención de una aplicación en JavaRMI.
- Conocer el modelo de despliegue y de los componentes de una aplicación en JavaRMI.
- Conocer las bases del diseño de interfaces y servicios en JavaRMI.

Recursos didácticos

- Clase magistral participativa.
- Tutoría activa.

Carga		
ECTS	**Horas presenciales**	**Horas no presenciales**
(Cálculo automático)	2,5	2,5

A-7.	**Arquitectura Cliente-Servidor.**	
Contenidos		
La arquitectura de diseño de aplicaciones más empleada es la basada en el modelo Cliente-Servidor. Es preciso conocerla cabalmente, pues además se utiliza como punto de referencia para la descripción de otros modelos de interacción. Las unidades A-4 y L-3 nos permiten profundizar en esta unidad con la soltura necesaria y madurez de conocimientos.		
Resultados de aprendizaje		
o Distinguir qué elementos definen una arquitectura C-S y su función. o Conocer las alternativas para la semántica de peticiones en C-S. o Distinguir las alternativas de diseño más habituales basadas en dos y tres capas. o Comprender la noción de estado de un servicio y sus implicaciones en el diseño. o Identificar las características que distinguen a los sistemas Rest.		
Recursos didácticos		
o Clase magistral participativa. o Tutoría activa.		
Carga		
ECTS	**Horas presenciales**	**Horas no presenciales**
0,2	2,5	2,5

A-8.	**Arquitectura P2P.**	
Contenidos		
La arquitectura P2P (*peer-to-peer* o “entre pares”) y sus variantes híbridas con C-S son un mecanismo cada vez más utilizado en el diseño de aplicaciones distribuidas actuales. Esta unidad permite preparar el desarrollo de la unidad L-5 que se realiza en el laboratorio.		
Resultados de aprendizaje		
o Conocer el modelo P2P y en qué condiciones resulta útil. o Conocer los elementos principales de una infraestructura para P2P. o Entender diversos ejemplos de aplicación P2P.		
Recursos didácticos		
o Clase magistral participativa. o Tutoría activa.		
Carga		
ECTS	**Horas presenciales**	**Horas no presenciales**
0,2	2,5	2,5

A-9.	**Control transaccional de recursos.**	
Contenidos		
Gran parte de los recursos de un sistema distribuido se manejan en secuencias de operaciones que es preciso coordinar entre sí para garantizar la consistencia del estado del sistema. Los mecanismos transaccionales son muy habituales en los modelos C-S sobre sistemas de gestión de bases de datos, pero además también se pueden aplicar mediante el middleware conveniente sobre cualquier porción del sistema que requiera la coordinación de accesos a recursos		
Resultados de aprendizaje		
o Comprender las ventajas de la sincronización mediante transacciones en servidores. o Conocer la interfaz transaccional aplicada a un servicio sencillo. o Conocer superficialmente los mecanismos que dan soporte a las transacciones.		
Recursos didácticos		
o Clase magistral participativa. o Tutoría activa.		
Carga		
ECTS	**Horas presenciales**	**Horas no presenciales**

0,2	2,5	2,5

A-10.	Middleware para la comunicación mediante colas de mensajes.

Contenidos

Los sistemas de comunicación mediante colas de mensajes, son muy utilizados, especialmente en la integración de aplicaciones de origen diverso, además de sus cualidades especiales para simular otros mecanismos de comunicación como la comunicación en grupo. Existen diversas plataformas middleware para la comunicación mediante colas de mensajes, de las cuales nos centraremos en JMS. Esta unidad permite preparar el desarrollo de la unidad L-6 que se realiza en el laboratorio.

Resultados de aprendizaje

- Conocer los elementos básicos de comunicación mediante colas de mensajes.
- Conocer el ámbito de aplicación de la comunicación mediante colas de mensajes.
- Conocer diversos middleware existentes y la tecnología subyacente.

Recursos didácticos

- Clase magistral participativa.
- Tutoría activa.

Carga

ECTS	Horas presenciales	Horas no presenciales
0,2	2,5	2,5

A-11.	Coordinación y acuerdo distribuido.

Contenidos

La coordinación y el acuerdo distribuido entre procesos son una necesidad en los sistemas distribuidos reales, para cuestiones tan importantes como el mantenimiento de la consistencia de los sistemas, gestión de la redundancia, y gestión de grupos, por poner algún ejemplo. En esta unidad se tratan los aspectos teóricos de la coordinación y el acuerdo con la suficiente generalidad como para permitir abordar un estudio más pormenorizado en otro momento.

Resultados de aprendizaje

- Comprender las complicaciones de la exclusión mutua distribuida.
- Conocer la existencia de diversos algoritmos de exclusión mutua y su rango de aplicación.
- Comprender el problema de la elección y distinguir las diversas soluciones existentes.
- Conocer el problema del consenso y su repercusión en algunos sistemas.

Recursos didácticos

- Clase magistral participativa.
- Tutoría activa.

Carga

ECTS	Horas presenciales	Horas no presenciales
0,2	2,5	2,5

A-12.	Arquitecturas orientadas a servicios, y servicios Web.

Contenidos

La integración de sistemas distribuidos diversos requiere de herramientas conceptuales que favorezcan la flexibilidad y la extensibilidad del sistema (*openness*), sobre todo en el caso de sistemas grandes y corporativos. Las arquitecturas orientadas a servicios se ofrecen como una aportación para facilitar este tipo de desafíos. Los servicios Web permite construir estos sistemas basados en la infraestructura de la web yh el modelo de POO.

Resultados de aprendizaje

- Conocer los principios de la arquitectura orientada a servicios.
- Poder discutir las ventajas y desventajas de la aplicación de SOA.
- Conocer los elementos de middleware que requiere una aplicación orientada a servicios.
- Comprender los fundamentos de un servicio Web.
- Conocer los elementos de un servicio web bajo SOAP y el ámbito de aplicación.

Recursos didácticos

- Clase magistral participativa.
- Tutoría activa.

Carga

ECTS	Horas presenciales	Horas no presenciales
0,1	1,25	2

A-13.	Características básicas del modelo de seguridad.

Contenidos

El modelo de seguridad de un sistema distribuido describe la forma en que se consiguen determinadas garantías de confidencialidad, integridad y no repudio. En esta unidad, sin entrar en la naturaleza de los algoritmos se describen los modelos más importantes para conseguir seguridad en la práctica, con el fin de abordar un punto de vista práctico, más tarde en la lección de laboratorio (L-8).

Resultados de aprendizaje

- Conocer el concepto de seguridad y las amenazas a la seguridad de un sistema distribuido.
- Conocer los modelos de seguridad más comunes de un sistema distribuido.
- Ser capaz de entender un modelo de seguridad y las especificaciones comunes de protocolo y suite criptográfica.

Recursos didácticos

- Clase magistral participativa.
- Tutoría activa.

Carga

ECTS	Horas presenciales	Horas no presenciales
0,1	1,25	2

A-14.	Replicación y tolerancia a fallos.

Contenidos

Una de las ventajas más aireadas de los sistemas distribuidos sobre los sistemas convencionales, es su robustez y tolerancia a fallos. En esta unidad se presentan los modelos y técnicas básicas que se incluyen en los sistemas modernos para tratar de realizar soluciones a este desafío de diseño.

Resultados de aprendizaje

- Comprender el modelo que soporta la replicación en un sistema distribuido.
- Comprender las características los servicios tolerantes a fallos.
- Conocer ejemplos reales de implementación de réplicas y tolerancia a fallos.

Recursos didácticos

- Clase magistral participativa.
- Tutoría activa.

Carga

ECTS	Horas presenciales	Horas no presenciales
0,2	2,5	2,5

A-15.	Referencia y recapitulación de sistemas distribuidos.

Contenidos

La recapitulación de ciertos conceptos, una vez discurrida la totalidad de las unidades, permite revisar y corregir pequeños defectos de perspectiva que inevitablemente nos vemos obligados a cometer como consecuencia de la secuencialidad del aprendizaje. Además se hace preciso revisitar la utilidad práctica de lo visto en los proyectos que contemplen la distribución, y a los que invariablemente se ha de enfrentar el futuro graduado.

Resultados de aprendizaje

- Distinguir las principales áreas de trabajo desarrolladas en la asignatura.
- Disponer de una visión unificada la repercusión de la distribución en el diseño de un sistema.
- Conocer como documentar los aspectos más importantes de la distribución en un proyecto informático.
- Reconocer las fuentes documentales más relevantes en la materia.

Recursos didácticos

- Clase magistral participativa.
- Tutoría activa.

Carga

ECTS	Horas presenciales	Horas no presenciales
0,2	2,5	2,5

L-1.	Primera sesión, toma de contacto con el laboratorio y utilización del compilador.

Contenidos

Esta primera unidad práctica se encarga de la presentación del entorno tecnológico en que se van a desarrollar las prácticas de la asignatura, y las herramientas didácticas planteadas para esta asignatura.

Resultados de aprendizaje

- Conocer el mecanismo de realización de las prácticas en la plataforma Moodle.
- Manejar el entorno de trabajo de desarrollo de las prácticas.
- Apreciar el mecanismo de corrección y evaluación de las prácticas.

Recursos didácticos

- Clase magistral participativa.

Carga

ECTS	Horas presenciales	Horas no presenciales
0,25	2	4

L-2.	Programación con sockets de una aplicación cliente-servidor.

Contenidos

La programación de sockets es la forma más sencilla de tomar contacto con la programación de sistemas distribuidos sencillos. Es muy fácil de comprender y no requiere de conocimientos especiales. Por ello sirve de excusa a la presentación de algunos de los desafíos básicos que plantean los sistemas distribuidos, como son el problema de la comunicación y compartición de datos, y la concurrencia.

Resultados de aprendizaje

- Entender las dinámicas de trabajo en sistemas con varios procesos.
- Comunicar procesos uno a uno mediante sockets.
- Entender la forma en que un servidor multienhebrados atiende diversos clientes.
- Programar mecanismos de sincronización entre procesos en Java.

Recursos didácticos

- Resolución de problemas.

Carga

ECTS	Horas presenciales	Horas no presenciales
0,25	2	4

L-3.	Diseño y programación de un protocolo sencillo para la comunicación C-S.

Contenidos

En esta unidad, el alumno se enfrenta a la necesidad de implementar un protocolo para la comunicación cliente-servidor, lo que permitirá conocer de primera mano los problemas de coordinación entre procesos distribuidos.

Resultados de aprendizaje

- Comprender la forma de implementar un protocolo cliente-servidor.
- Entender cómo influye la noción de estado del protocolo en el servicio.
- Diseñar un protocolo sencillo para interacción interna en una aplicación.

Recursos didácticos

- Resolución de problemas.

Carga

ECTS	Horas presenciales	Horas no presenciales
0,25	2,	4,

L-4.	Programación de una aplicación C-S mediante JavaRMI.

Contenidos

JavaRMI es uno de los mecanismos middleware más sencillos para iniciarse en la programación orientada al objeto de sistemas distribuidos, sin perder la generalidad que nos permite entender otros sistemas similares como CORBA. Al “hacer manos” con un sistema real, se nos plantean los habituales problemas tecnológicos que, de su solución, nos permiten conocer cómo funciona en la realidad este tipo de tecnologías.

Resultados de aprendizaje

o Aplicar la POO a la construcción de una aplicación distribuida sencilla. o Compilar y desplegar aplicaciones basadas en objetos remotos. o Aplicar las variantes de comunicación mediante referencias remotas y locales. o Diseñar y programar un servicio sencillo para JavaRMI.
Recursos didácticos
o Resolución de problemas.
Carga

ECTS	Horas presenciales	Horas no presenciales
0,25	2	4

L-5.	**Programación del mecanismo de CS mediante Callbacks.**
Contenidos	
Los sistemas distribuidos requieren de mecanismos de invocación bidireccional (push/pull). La implementación del patrón observador permite la construcción de este tipo de aplicaciones, mediante lo que habitualmente se conoce como mecanismo de callbacks.	
Resultados de aprendizaje	
o Aplicar el conocido mecanismo de callbacks para implementar aplicaciones “full responsiveness”. o Profundizar en la comprensión de las implicaciones de la concurrencia en el diseño de una aplicación distribuida de cierta complejidad..	
Recursos didácticos	
o Resolución de problemas.	
Carga	

ECTS	Horas presenciales	Horas no presenciales
0,25	2	4

L-6.	**Programación de una aplicación P2P con JavaRMI.**
Contenidos	
Existen plataformas específicas para la programación peer-to-peer, sin embargo, la dificultad de operar con este tipo de sistemas queda suficientemente patente en un diseño basado en sockets, RPC, o en RMI. En este caso, optamos por esta última opción, lo que permite ver conceptos como *callback* y profundizar en nuestro conocimiento de JavaRMI.	
Resultados de aprendizaje	
o Conocer los requisitos prácticos para aplicar la tecnología JavaRMI al diseño de aplicaciones P2P. o Diseñar interfaces de tipo *callback*. o Apreciar las restricciones prácticas de la invocación remota entre diversos procesos.	
Recursos didácticos	
o Resolución de problemas.	
Carga	

ECTS	Horas presenciales	Horas no presenciales
0,25	2	4

L-7	**Programación de una aplicación mediante colas de mensajes, en Java.**
Contenidos	
La programación mediante sistemas basados en colas de mensajes es un mecanismo de aglutinar muchas aplicaciones, y nos permite construir modelos de comunicación con semánticas muy diversas. En esta práctica se trata de entrar en contacto con un modelo de colas de mensajes muy popular, como el que ofrece Java. Se utilizará algún middleware popular de carácter abierto, como puede ser GlassFish.	
Resultados de aprendizaje	
o Conocer las implicaciones tecnológicas del uso del middleware de manejo de colas de mensajes. o Aplicar la programación de colas de mensajes al diseño de aplicaciones distribuidas. o Compilar y desplegar aplicaciones basadas en colas de mensajes. o Aplicar las variantes de semántica de colas de mensajes para la comunicación C-S y de grupo. o Programar una aplicación sencilla C-S y P2P mediante colas de mensajes.	
Recursos didácticos	
o Resolución de problemas.	

Carga		
ECTS	**Horas presenciales**	**Horas no presenciales**
0,25	2	4

L-8.	**Seguridad en sockets e invocación remota.**

Contenidos

Existen mecanismos para incorporar seguridad a la comunicación basada en paso de mensajes mediante sockets, y por extensión para la invocación remota en JavaRMI. A pesar de ello no es un mecanismo directo, y exige el conocimiento de las técnicas básicas. Los alumnos deberán generar su propio certificado y hacer funcionar sockets seguros mediante SSL y soportar la comunicación segura mediante la factoría de sockets de JavaRMI.

Resultados de aprendizaje

- Comprender la infraestructura básica de sockets seguros mediante SSL.
- Ser capaces de generar certificados para la comunicación con sockets.
- Conseguir mantener una comunicación segura mediante sockets.
- Conseguir invocación segura en JavaRMI.

Recursos didácticos

- Resolución de problemas.

Carga

ECTS	Horas presenciales	Horas no presenciales
0,25	2	4

S-1	**Modelos de distribución avanzados.**

Contenidos

Existen muchos modelos de distribución avanzados, o variantes de los existentes, que no es posible visitar con profundidad suficiente; este es el caso de grid computing, cooperative y cloud computing. Los alumnos recibirán un encargo en grupos de 4 personas para realizar una revisión documental y una síntesis que se concretarán en un portafolio y en una presentación pública.

Resultados de aprendizaje

- Discutir modelos de distribución y plataformas novedosas de actualidad.
- Descubrir documentación y manejar terminología usual en sistemas distribuidos.
- Adquirir vocabulario y discutir sobre la tecnología de los SD.

Recursos didácticos

- Aprendizaje colaborativo.
- Tutoría Activa.

Carga

ECTS	Horas presenciales	Horas no presenciales
0,7	2	15

S-2	**Ejemplos de aplicaciones distribuidas comerciales.**

Contenidos

Las aplicaciones distribuidas comerciales son el punto de referencia que nos permite ver en perspectiva los conocimientos de esta asignatura. En este seminario se encarga en grupos de 4 personas, realizar una revisión documental (en las mismas condiciones que el seminario S-1) de una aplicación real (o un grupo de ellas, según el caso), que nos permita contrastar los conocimientos adquiridos con las aplicaciones reales.

Resultados de aprendizaje

- Discutir modelos de distribución y aplicaciones reales.
- Aplicar los conceptos de los sistemas distribuidos a ejemplos tecnológicos reales.
- Contrastar las nociones de SD vistas con la terminología propia de las empresas de sistemas.
- Conocer nuevas plataformas y aplicaciones reales.

Recursos didácticos

- Aprendizaje colaborativo
- Tutoría Activa.

Carga

ECTS	Horas presenciales	Horas no presenciales
0,7	2	15

6. Temporalización

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
A-0. Presentación de la asignatura, metodología y evaluación.	0,10	20/09/11
A-1. Conceptos básicos. Desafíos de diseño de los sistemas distribuidos.	0,10	21/09/11
L-1. Toma de contacto coon el laboratorio y utilización del compilador.	0,25	27/09/11
A.2. Comunicación por paso de mensajes mediante sockets.	0,10	28/09/11
L-2. Programación con sockets de una aplicación cliente-servidor.	0,25	04/10/11
A-3. Comunicación entre procesos.	0,10	05/10/11
TA-1. Tutoría Activa - 1 sobre: A-1, A-2 y A-3	0,25	11/10/11
A-4. Protocolos para la comunicación de aplicaciones.	0,10	18/10/11
A-5. Modelos fundamentales del diseño de los sistemas distribuidos.	0,10	19/10/11
L-3. Diseño y programación de un protocolo sencillo para la C-S.	0,25	25/10/11
A-6. Middleware para la invocación remota entre aplicaciones.	0,10	26/10/11
A-7. Arquitectura Cliente-Servidor	0,10	02/11/11
TA-2. Tutoría Activa - 2 sobre: A-4, A-5, A-6 y A-7	0,25	08/11/11
A-8. Arquitectura P2P	0,10	09/11/11
L-4. Programación de una aplicación C-S mediante Java-RMI	0,25	15/11/11
S-1. Modelos de distribución avanzados.	0,70	16/11/11
L-5. Programación del mecanismo de C-S con Callbacks	0,25	22/11/11
A-9. Control transaccional de recursos.	0,10	23/11/11
A-10. Middleware para la comunicación mediante colas de mensajes.	0,10	29/11/11
L-6. Programación de una aplicación P2P con JavaRMI.	0,25	30/11/11
A-11. Coordinación y acuerdo distribuido.	0,10	07/12/11
TA-3. Tutoría Activa - 3 sobre: A-8, A-9 y A-10	0,25	13/12/11
A-12. Arquitecturas orientadas a servicios, y servicios Web.	0,10	14/12/11
L-7. Programación de una aplicación mediante colas de mensajes.	0,25	20/12/11
A-13. Seguridad y distribución.	0,10	21/12/11
L-8. Seguridad - sockets y RMI.	0,25	10/01/12
A-14. Replicación y tolerancia frente a fallos.	0,10	11/01/12
S-2. Ejemplos de aplicaciones distribuidas comerciales.	0,70	17/01/12
TA-4. Tutoría Activa - 4 sobre: A-11, A-12, A-13 y A-14.	0,25	18/01/12
Examen Teoría ordinario (véase la página de la ETSI Inf).	0,25	01/02/12
Examen Teoría extraordinario (véase la página de la ETSI Inf.)	0,25	10/07/12

7. Tabla resumen de los instrumentos, procedimientos y sistemas de evaluación/calificación

La evaluación de los resultados del aprendizaje será continua a lo largo de las actividades de la asignatura, y consta de las siguientes actividades: Test de preparación de la tutoría activa, cuestionario de desarrollo de los guiones de prácticas, presentación de resultados en las sesiones de seminario y una prueba de test final. Además se valorará la participación del alumno en las actividades presenciales y no presenciales.

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Varios **test de elección múltiple**, de una duración de 10 minutos (aproximadamente), que incluye las unidades indicadas.	15,00%	• Asociado a cada tutoría activa. • No es preciso superar estos cuestionarios para poder superar la asignatura.
Cuestionario de desarrollo de guión de prácticas, con Test de elección múltiple y respuestas cortas.	30,00%	• Asociado a cada sesión de laboratorio. • El alumno deberá completar satisfactoriamente al menos el 75 por ciento de las prácticas para poder superar la asignatura, y las prácticas se realizan obligatoriamente en el laboratorio dispuesto por el centro en el horario. En el caso de que el alumno no haya conseguido superar satisfactoriamente una lección de laboratorio en una sesión, podrá hacerlo en la siguiente sesión de laboratorio a costa del tiempo disponible para la lección actual
Presentación pública en las sesiones de Seminario de un encargo de búsqueda y síntesis de información, a través de presentación y evaluación del portafolio asociado. A esta evaluación se unen la regularidad en la supervisión del trabajo mediante al menos dos tutorías personalizadas por grupo, para la revisión del portafolio de trabajo.	15,00%	• Asociado a cada sesión de seminario • El alumno deberá realizar la defensa de los trabajos para obtener una evaluación positiva en este apartado
Se tendrá en cuenta la **regularidad** y participación del alumno en las sesiones presenciales de Aula, Laboratorio y Seminarios.	5,00% adicional	• Se considerará actividad regular cuando el alumno asista al menos al 75 % de las actividades presenciales desarrolladas. • Esta cantidad se añadirá a la obtenida por el alumno antes de truncar al 100% para obtrener la calificación final.
Examen escrito con aproximadamente 50 preguntas de elección múltiple.	40,00%	• Asociado a la convocatoria ordinaria y extraordinaria de la asignatura