26-27

MICROGRADO EN DISEÑO Y PROGRAMACIÓN DE APLICACIONES DISTRIBUIDAS
OPTATIVAS CURSO

GUÍA DE ESTUDIO PÚBLICA

APLICACIONES DISTRIBUIDAS

CÓDIGO 71023022

UNED

26-27

APLICACIONES DISTRIBUIDAS

CÓDIGO 71023022

ÍNDICE

PRESENTACIÓN Y CONTEXTUALIZACIÓN
REQUISITOS Y/O RECOMENDACIONES PARA CURSAR LA ASIGNATURA
EQUIPO DOCENTE
HORARIO DE ATENCIÓN AL ESTUDIANTE
TUTORIZACIÓN EN CENTROS ASOCIADOS
COMPETENCIAS QUE ADQUIERE EL ESTUDIANTE
RESULTADOS DE APRENDIZAJE
CONTENIDOS
METODOLOGÍA
SISTEMA DE EVALUACIÓN
BIBLIOGRAFÍA BÁSICA
BIBLIOGRAFÍA COMPLEMENTARIA
RECURSOS DE APOYO Y WEBGRAFÍA
IGUALDAD DE GÉNERO

NOMBRE DE LA ASIGNATURA	APLICACIONES DISTRIBUIDAS
CÓDIGO	71023022
CURSO ACADÉMICO	2026/2027
DEPARTAMENTO	SISTEMAS DE COMUNICACIÓN Y CONTROL
TÍTULO EN QUE SE IMPARTE	GRADO EN INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN
CURSO - PERIODO - TIPO	ESPECÍFICO PARA INGENIEROS TÉCNICOS EN INFORMÁTICA DE GESTIÓN EN UNED - OPTATIVAS CURSO - SEMESTRE 1 - OBLIGATORIAS
CURSO - PERIODO - TIPO	ESPECÍFICO PARA INGENIEROS TÉCNICOS EN INFORMÁTICA DE GESTIÓN - OPTATIVAS CURSO - SEMESTRE 1 - OBLIGATORIAS
CURSO - PERIODO - TIPO	GRADO EN INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN - TERCER CURSO - SEMESTRE 1 - OBLIGATORIAS
TÍTULO EN QUE SE IMPARTE	MICROGRADO EN DISEÑO Y PROGRAMACIÓN DE APLICACIONES DISTRIBUIDAS
PERIODO - TIPO	- SEMESTRE 1 - OBLIGATORIAS
Nº ETCS	6
HORAS	150.0
IDIOMAS EN QUE SE IMPARTE	CASTELLANO

PRESENTACIÓN Y CONTEXTUALIZACIÓN

En la actualidad la diversificación de los recursos de computación asociados a los diferentes proyectos de desarrollo está íntimamente ligada a la estructura de red disponible y a su evolución. Internet, y el protocolo TCP/IP, marcó un crecimiento exponencial en la disponibilidad de dichos recursos obligando a modificar la visión tradicional del modelo de programación sobre un computador único. El objetivo principal de la asignatura es mostrar las diferentes alternativas de computación distribuida desde el punto de vista práctico, mediante el uso de un lenguaje de programación como Java que dispone de una gran cantidad de ejemplos y desarrollos que el programador y/o analista puede emplear en sus propios proyectos. Para ello se presentarán las diferentes tecnologías existentes, RMI y CORBA, que representan los ejemplos de middlewares aplicados en el paradigma cliente-servidor, finalizando el estudio con las tendencias más actuales en definición e implementación de servicios Web.

En la materia de Tecnologías Web y Aplicaciones distribuidas del Grado en Ingeniería de las Tecnologías de la Información se tres asignaturas en el siguiente orden:

•Aplicaciones Distribuidas: De carácter obligatorio, se imparte en el primer cuatrimestre del tercer curso de dicho grado y supone una carga de 6 créditos ECTS.

•Tecnologías Web: De carácter obligatorio, se imparte en el segundo cuatrimestre del tercer curso de dicho grado y supone una carga de 6 créditos ECTS.
•Tecnologías y Programación Integrativas: De carácter obligatorio, se imparte en el primer cuatrimestre del cuarto curso de dicho grado y supone una carga de 6 créditos ECTS.

Esta asignatura es, por tanto, la primera asignatura que el estudiante deberá afrontar dentro de dicha materia. Es un hecho que la diversificación de los recursos de computación en el momento actual traspasa los límites de la computación tradicional, basada en potentes y caros computadores. Estos recursos se encuentran distribuidos a lo largo de redes de comunicación en diferentes localizaciones, listos para ser usados en diferentes ámbitos. Sin embargo, desde el punto de vista de la programación, la accesibilidad a los recursos implica un conjunto de reglas de acceso que permita de manera transparente el uso de dichos recursos. De esta manera surge de manera natural el concepto de middleware (capa de servicios) que proporciona un conjunto de servicios a los programadores que les permite diseñar y construir aplicaciones software distribuidas en diferentes medios de comunicación, principalmente redes. Esta capa de servicios puede ser muy compleja o casi inexistente, dependiendo del tipo de paradigma de computación distribuida empleado en la construcción del software.

Los middlewares de computación distribuida más conocidos son CORBA (Common Object Request Broker Agent, definido por el OMG) y RMI (Remote Method Invocation, de Sun Microsystems). CORBA se caracteriza por su independencia de lenguaje y plataforma, pero a la vez por su complejidad de implementación. RMI, por el contrario, es más simple pero está limitado al mundo Java, lo que hace que sea una solución más limitada. Ambos middlewares disponen de normas de funcionamiento y de protocolos propios que pueden resultar complejos de comprender y “pesados” desde el punto de vista de la transmisión de información. En la actualidad ha surgido otra tecnología de computación distribuida, la arquitectura SOA (Service Oriented Architecture) que pretende estandarizar las normas de funcionamiento para que la transparencia de ubicación sea completa (independiente de lenguaje de programación y/o sistema operativo) y la comunicación se realice mediante el estándar XML bajo distintas normas (SOAP, WSDL, UDDI), Adicionalmente, la arquitectura RESTful se ha convertido en el estándar de desarrollo de sistemas y aplicaciones distribuidas en la red, incluyendo la gestión de APIs e interconexión de los mismos.

Esta asignatura pretende centrarse en los conocimientos y mecanismos necesarios para abordar una aproximación profesional al diseño e implementación de soluciones software en el ámbito de la computación distribuida, presentando las bases y conceptos necesarios para usar la terminología adecuada en los diferentes paradigmas existentes. También se realiza una presentación detallada de los diferentes middlewares, CORBA y RMI, incidiendo en dichas tecnologías y presentando sus antecedentes (sockets y RPC). Son middlewares antiguos, pero sus fundamentos son la base de los conceptos actuales de

desarrollo. Finalmente se introducen las tecnologías SOA/REST y se trabaja en la definición e implementación de servicios Web.
La distribución temporal de la asignatura se ha adecuado de forma que se premia el componente práctico mediante la realización de actividades que complementan de manera concreta diferentes áreas del temario de la asignatura. En concreto para cada módulo de la asignatura se desarrollará un conjunto de preguntas de autoevaluación que se elaborarán en la propia plataforma de formación y que permitirá comprobar la efectividad del aprendizaje. Junto a los cuestionarios se fomentará el uso de los foros de discusión sobre los contenidos teóricos que promueva de forma colaborativa la generación de preguntas frecuentas y debates interesantes sobre diferentes aspectos teóricos. Adicionalmente se ha programado una actividad práctica que condensará lo aprendido respecto de las diferentes arquitecturas, y que consistirá en una implementación/uso de un caso real de entorno de computación distribuida.

REQUISITOS Y/O RECOMENDACIONES PARA CURSAR LA ASIGNATURA

Se considera imprescindible para la realización y seguimiento del curso, que el alumno posea unos sólidos fundamentos en las siguientes áreas:

- Programación con Java. Durante la práctica totalidad del temario de la asignatura, los ejemplos prácticas ya actividades planificadas presuponen un conocimiento medio del lenguaje de programación Java, es decir, la capacidad de desarrollar clases Java que implementen diferentes comportamientos y usen diferentes librerías del propio lenguaje. En particular, la asignatura enseña a emplear ciertas librerías específicas adecuadas para entornos de computación distribuida.
- Redes de computadores. Todo lo relativo a la computación distribuida se centra en el conocimiento del funcionamiento de los diferentes protocolos de comunicación. Es muy importante el conocimiento de la pila de protocolos de TCP/IP, ya que constituyen la base de los protocolos de comunicación de CORBA o RMI, i de los propios sockets.
- Sistemas operativos. Los propios sistemas operativos están pensados como piezas de software que pueden estar distribuidas en diferentes recursos físicos. Por tanto. la propia programación de los sistemas operativos es un ejemplo de desarrollo de sistema distribuido. Los middlewares que se estudian en la asignatura utilizan los servicios de los sistemas operativos en los cuales se ejecutan o proporcionan a su vez servicios distribuidos, por lo que es importante conocer detalles específicos sobre los diferentes ámbitos de funcionamiento de un SO (sistemas de ficheros, memoria compartida, etc.).

EQUIPO DOCENTE

Nombre y Apellidos	RAFAEL PASTOR VARGAS (Coordinador/a de asignatura)
Correo Electrónico	rpastor@dia.uned.es
Teléfono	91398-8383
Facultad	ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA
Departamento	SISTEMAS DE COMUNICACIÓN Y CONTROL

Nombre y Apellidos	RAFAEL PASTOR VARGAS (Coordinador/a de asignatura)
Correo Electrónico	rpastor@scc.uned.es
Teléfono	91398-8383
Facultad	ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA
Departamento	SISTEMAS DE COMUNICACIÓN Y CONTROL

Nombre y Apellidos	MIGUEL ROMERO HORTELANO
Correo Electrónico	mromero@scc.uned.es
Teléfono	91398-7943
Facultad	ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA
Departamento	SISTEMAS DE COMUNICACIÓN Y CONTROL

HORARIO DE ATENCIÓN AL ESTUDIANTE

La tutorización de los estudiantes tendrá lugar esencialmente a través de los foros de la plataforma, aunque también podrán utilizarse ocasionalmente otros medios, tales como chats interactivos, servicios de mensajería instantánea y el correo electrónico. Adicionalmente, está también previsto, para temas personales que no afecten al resto de los estudiantes, atender consultas en persona o por teléfono.
El seguimiento del aprendizaje se realizará revisando la participación de los alumnos en los distintos foros de debate y las aportaciones de material nuevo además de la entrega en fecha de los diferentes trabajos prácticos que se han planificado durante la evolución del curso.
En caso de necesitar contactar con el Equipo Docente por medios distintos al curso virtual, se utilizará preferentemente el correo electrónico, pudiéndose también realizar consultas telefónicas y entrevista personal en los horarios establecidos:

Rafael Pastor Vargas
Lunes de 16 a 20 horas
rpastor@scc.uned.es
Tfno: 91 398 8383

Dirección postal:
Rafael Pastor Vargas, Despacho 5.07
Dpto. de Sistemas de Comunicación y Control
E.T.S.I. Informática (UNED)
C/ Juan del Rosal, 16
28040 Madrid

TUTORIZACIÓN EN CENTROS ASOCIADOS

Las sesiones de tutoría se imparten presencialmente y en línea desde el Centro Asociado de referencia o, en su defecto, desde otro Centro Asociado. A continuación se muestran los centros asociados y extensiones en las que se imparten tutorías de la asignatura.

COMPETENCIAS QUE ADQUIERE EL ESTUDIANTE

En relación con las competencias de la materia, la asignatura Aplicaciones Distribuidas contribuye al desarrollo de las siguientes competencias, generales y específicas, del grado en que se imparte. Se muestran dichas competencias respetando la etiquetación de las mismas en la memoria de verificación de la ANECA:
- Competencias generales:
(CG.1) Competencias de gestión y planificación: Iniciativa y motivación. Planificación y organización (establecimiento de objetivos y prioridades, secuenciación y organización del tiempo de realización, etc.). Manejo adecuado del tiempo
(CG.2) Competencias cognitivas superiores: selección y manejo adecuado de conocimientos, recursos y estrategias cognitivas de nivel superior apropiados para el afrontamiento y resolución de diversos tipos de tareas/problemas con distinto nivel de complejidad y novedad: Análisis y Síntesis. Aplicación de los conocimientos a la práctica Resolución de problemas en entornos nuevos o poco conocidos. Pensamiento creativo. Razonamiento crítico. Toma de decisiones
(CG.5) Competencias en el uso de las herramientas y recursos de la Sociedad del Conocimiento: Manejo de las TIC. Competencia en la búsqueda de información relevante. Competencia en la gestión y organización de la información. Competencia en la recolección de datos, el manejo de bases de datos y su presentación

- Competencias específicas:
(BC.13) Conocimiento y aplicación de las herramientas necesarias para el almacenamiento, procesamiento y acceso a los Sistemas de Información, incluidos los basados en web
(BC.14) Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real
(BTEti.3) Capacidad para emplear metodologías centradas en el usuario y la organización para el desarrollo, evaluación y gestión de aplicaciones y sistemas basados en tecnologías de la información que aseguren la accesibilidad, ergonomía y usabilidad de los sistemas
(BTEti.5) Capacidad para seleccionar, desplegar, integrar y gestionar sistemas de información que satisfagan las necesidades de la organización, con los criterios de coste y calidad identificados
(BTEti.6) Capacidad de concebir aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil

RESULTADOS DE APRENDIZAJE

Los resultados de aprendizaje asociados al estudio de la asignatura, que se corresponden con la memoria de verificación de ANECA para el grado, son los siguientes (se mantiene la notación usada en dicha memoria):

R1. Diseñar, desarrollar y mantener servicios y aplicaciones en tecnologías web e integralas en los sistemas de información corporativos.

R3. Diseñar, desarrollar y mantener aplicaciones distribuidas basadas en servicios web

R4. Comprender, desarrollar y manejar las técnicas y protocolos de interacción entre aplicaciones utilizando diferentes tecnologías de comunicación

R7. Comprender, analizar y dimensionar especificaciones técnicas de la ingeniería informática y llevarlas a la práctica satisfaciendo los requisitos, optimizando los recursos y asegurando la calidad del resultado.

Adicionalmente, como resultados de aprendizaje particulares englobados en los generales se pretende que el alumno consiga los siguientes objetivos:

O1. Familiarizarse con los conceptos y funcionalidad básicos de los sistemas distribuidos y plantearse su uso en proyectos reales.
O2. Identificar los diferentes paradigmas de la computación distribuida, recopilando información relativa problemas asociados a la programación distribuida: escalabilidad, pérdidas de rendimiento, soporte multiplataforma.
O3. Recordar la definición de conectores (socket) y diferenciar las diferentes alternativas de uso en los escenarios de transmisión confiable (TCP) y no confiable (UDP).
O4. Programar código que implemente una solución basada en conectores (sockets)
O5. Conocer en qué consiste el modelo cliente/servidor y diferenciar las variantes existentes y sus diferentes clasificaciones (con o sin estado, orientado o no a conexión, secuencial o concurrente).
O6. Comprender los mecanismos de implementación de un servicio basado en el modelo cliente/servidor, de forma que sea sencillo gestionar los cambios en el modelo en base a modificaciones de dicho software distribuido.
O7. Relacionar la programación orientada a objetos a la programación distribuida y el paso de mensajes, gestionando la posibilidad de manipular e invocar servicios almacenados en objetos remotos.
O8. Implementar diferentes estrategias de programación con RMI, escribiendo el código correspondiente.
O9. Conocer la normativa estándar del OMG relativas al middleware CORBA y sus fundamentos.

O10. Probar la capacidad de desarrollo distribuido con CORBA y compararla con RMI, mediante la revisión de las herramientas específicas de desarrollo y los factores de complejidad asociados.
O11. Pensar y juzgar de manera conjunta, a modo de equipo de trabajo en un entorno corporativo, las prácticas adecuadas para la implantación de un modelo de programación distribuida con RMI o CORBA.
O12. Conocer la arquitectura basada en servicios (SOA) y las normativas estándar asociadas (SOAP, WDSL, UDDI), y demostrar los beneficios intrínsecos de usar XML como lenguaje de representación común de dichas normas.
O13. Planificar un proyecto de aplicación distribuida en un entorno real y probar que la solución adoptada cumple los requerimientos del proyecto software especificado.
O14. Revisar, juzgar y criticar los conocimientos adquiridos.
O15. Reconocer el espacio de trabajo virtual personalizado del curso y diferenciar las herramientas disponibles por parte del equipo docente
O16. Conocer el funcionamiento básico de la entrega de actividades y/o ejercicios prácticos relativos al seguimiento y evaluación de los progresos del curso

CONTENIDOS

Módulo 1. Introducción a la computación distribuida.

Índice

1.1. Definiciones.
1.2. Historia de la computación distribuida.
1.3. Tipos de computación distribuida.
1.4. Fortalezas y debilidades.
1.5. Fundamentos de sistemas operativos
1.6. Fundamentos de red.
1.7. Fundamentos de ingeniería del software.

La computación distribuida es en la actualidad la tecnología estándar de desarrollo de aplicaciones basadas en la Web. Para ver las tendencias actuales es preciso hacer un análisis de las historia y fundamentos asociados a este paradigma global de programación, mostrando en detalle los factores que inciden en la propia computación distribuida: la red de comunicación, los sistemas operativos subyacentes y el proceso de desarrollo software.

Los contenidos del módulo son introductorios y no tienen una dificultad elevada, siendo sencillos su estudio.

Módulo 2. Paradigmas de la computación distribuida.

Índice

2.1. Comunicación entre procesos

2.2 Paradigmas y abstracción.

2.3. Aplicación de ejemplo.

2.4. Paradigmas para las aplicaciones distribuidas.

2.5. Factores a tener en cuenta en la decisión del paradigma.

Existen diferentes aproximaciones a la computación distribuida, denominadas paradigmas. Entre ellas se pueden destacar los modelos de desarrollo basados en modelo cliente-servidor, basados en pares de iguales, basados en mensajes o los modelos más “antiguos” que usaban RPC (aunque en la actualidad hay otro modelo RPC, denominado XML-RPC que se emplea de manera extensa). El módulo pretende mostrar las características más relevantes de cada una de estas aproximaciones.

Los contenidos del módulo son introductorios y no tienen una dificultad elevada, siendo sencillos su estudio.

Módulo 3. Los Sockets Java.

Índice

3.1. Introducción.

3.2. El API de sockets de datagramas (UDP).

3.3. El API de sockets de flujo (TCP).

3.4. Sockets seguros.

Uno de los primeros modelos de programación distribuida que se emplearon (y que aún sigue en vigor) es el basado en sockets (conectores). Estos conectores son flujos de información entre dos servicios, usualmente en computadores diferentes, que definen su propio protocolo de intercambio de información. El módulo presenta los dos protocolos basados en conectores: UDP y TCP, finalizando con una presentación al uso de conectores seguros en entornos donde se necesita proteger la información que se transmite entre los servicios.

Los contenidos del tema tienen una dificultad media ya que necesitan de conocimientos de programación para su perfecta comprensión, estos conocimientos se imparten en cursos anteriores del grado.

Módulo 4. El paradigma cliente-servidor.

Índice

4.1. Definición y características.

4.2. Implementación de un servicio de red.
4.3. Servidores orientados y no orientados a conexión.
4.4. Servidores iterativos y concurrentes.
4.5. Servidores con estado.

El paradigma más empleado en la computación distribuida es, sin lugar a duda, el modelo cliente-servidor. En dicho modelo un cliente accede a un servicio proporcionado por un servidor y este le proporciona la información que necesita a través de dicho servicio. Ejemplos de este modelo puede ser la propia Web, a través de su protocolo HTTP donde el servicio ofrecido en el acceso a la información almacenada y la información transmitida es la propia página HTML o equivalente. El módulo muestra los diferentes componentes involucrados la comunicación y presenta varias clasificaciones de los servidores relativa a diferentes aspectos del paradigma cliente-servidor.
Los contenidos del tema tienen una dificultad media ya que necesitan de conocimientos de programación y sistemas operativos para su perfecta comprensión, estos conocimientos se imparten en cursos anteriores del grado.

Módulo 5. Objetos distribuidos en Java: RMI.

Índice
5.1. Paso de mensajes versus Objetos distribuidos.
5.2. Arquitectura de un sistema de objetos distribuidos: RPC y RMI.
5.3. Invocaciones remotas mediante RMI: Arquitectura y API.
5.4. Ejemplo de aplicación RMI.
5.5. RMI avanzado.

RMI (Remote Method Invocation) es una capa de servicios distribuidos (búsqueda de objetos remotos, invocación de interfaces y métodos, etc.) que permite desarrollar aplicaciones distribuidas basadas en el modelo cliente-servidor y centradas en la metodología de desarrollo orientada a objetos. El módulo presenta los fundamentos de RMI, además de características avanzadas que pueden ser muy útiles en entornos profesionales.
Los contenidos del tema tienen una dificultad media ya que necesitan de conocimientos de programación para su perfecta comprensión, estos conocimientos se imparten en cursos anteriores del grado.

Módulo 6. CORBA en Java.

Índice
6.1. Introducción a CORBA.
6.2. Java IDL
6.3. Factores de diseño con CORBA.

CORBA, al igual que RMI, proporciona la capa de servicios distribuidos que permite implementar soluciones software distribuidas, La diferencia fundamental radica en no emplear objetos para el desarrollo de la solución (lo que permite usar cualquier tipo de lenguaje de programación) ni un solo lenguaje de programación (RMI usa solo Java), lo que obliga a definir un lenguaje intermedio (IDL, Interface Definition Language) que debe ser traducido al lenguaje que solicitará/implementará el servicio distribuido correspondiente. El módulo muestra los fundamentos de CORBA, su implementación Java mediante Java IDL y los factores de diseño del software distribuido que se deben tener en cuenta a la hora de emplear esta tecnología.
Los contenidos del tema tienen una dificultad media ya que necesitan de conocimientos de programación para su perfecta comprensión, estos conocimientos se imparten en cursos anteriores del grado.

Módulo 7. Servicios Web.

Índice
7.1. ¿Qué es un servicio Web?
7.2. Funcionalidad de transporte de WSA: SOAP
7.3. Funcionalidad de descripción de WSA: WSDL
7.4. Funcionalidad de descubrimiento de WSA: UDDI
7.5. Implementación de la arquitectura WSA

Los servicios Web son parte de la definición de la arquitectura basada en servicios (SOA). Dicha arquitectura define unas reglas de comportamiento estándar que permiten la definición de servicios mediante XML (WDSL), la búsqueda y localización de servicios (UDDI) y las invocaciones a dichos servicios (SOAP). Todas estas reglas de funcionamiento se implementan mediante el uso de estándares basados en XML, lo que permiten que cualquier tipo marco de desarrollo pueda exportar sus servicios con independencia de la plataforma en la que se desarrollen. El módulo muestra las características de SOA en los ámbitos de comunicación, descripción y descubrimiento.
Los contenidos del tema tienen una dificultad media/alta ya que necesitan de conocimientos de programación, sistemas operativos y lenguajes de etiquetas para su perfecta comprensión, estos conocimientos se imparten en cursos anteriores del grado.

METODOLOGÍA

De forma resumida la metodología docente se concreta en:
•Adaptada a las directrices del EEES.

•La asignatura no tiene clases presenciales. Los contenidos teóricos se impartirán a distancia, de acuerdo con las normas y estructuras de soporte telemático de la enseñanza en la UNED. Esta asignatura se impartirá a distancia, utilizando una plataforma de educación a través de Internet. Se organizarán foros de discusión para dudas y debates.
•El material docente incluye cuestionarios de autoevaluación sobre los contenidos de cada tema y distintos tipos de actividades relacionadas con la asignatura: consulta bibliográfica, consulta de información en Internet, trabajos de análisis y resumen, y uso avanzado de herramientas software.
•Las actividades de aprendizaje se estructuran en torno al estado del arte en cada una de las materias del curso y a los problemas en los que se va a focalizar en el desarrollo de trabajo final. Cada actividad (prueba de evaluación a distancia) tendrá asociado un pero específico en la evaluación final de la asignatura.

La metodología docente se desarrolla de acuerdo con los siguientes principios:
•Además de adoptar la metodología docente general del grado, y en coherencia con el propósito de utilizar los sistemas interactivos de educación con fines pedagógicos y/o formativos, la asignatura diseñada se apoya en gran medida en los recursos educativos de este medio.
•La metodología del trabajo de la asignatura se basa en una planificación temporal de las actividades. Existirán diferentes módulos o unidades didácticas. Cada uno de éstos tendrá asociado unas unidades de aprendizaje y un material asignado (capítulos del libro base, artículos relacionados, direcciones adicionales de Internet, o cualquier otro material que se proporcione). Se asignará un período para cada módulo, en el que deberán realizar las actividades relacionadas con el mismo, aunque dado el perfil del estudiante de la UNED se manejarán de manera flexible dichas fechas.

SISTEMA DE EVALUACIÓN

TIPO DE PRUEBA PRESENCIAL

Tipo de examen	Examen tipo test
Preguntas test	20
Duración del examen	60 (minutos)

Material permitido en el examen

Calculadora no programable

Criterios de evaluación

La prueba consta de un test de 20 preguntas. Para superar la prueba se deberá obtener una puntuación mínima de 5 puntos. En cada pregunta del test se proponen cuatro respuestas de las cuales sólo una es correcta. Únicamente puntuarán las preguntas contestadas. Si la respuesta es correcta la puntuación será de 0.5 puntos y si es incorrecta restará 0.2 puntos.

% del examen sobre la nota final	50
Nota del examen para aprobar sin PEC	0
Nota máxima que aporta el examen a la calificación final sin PEC	0
Nota mínima en el examen para sumar la PEC	0

Comentarios y observaciones

La asignatura se evalúa mediante el examen principal y la PEC (Práctica de laboratorio), y se deben aprobar de manera individual **cada parte** (examen y y PEC). Es decir, no se puede aprobar una de las partes por separado (examen o PEC) y aprobar la asignatura. Se evalúan como un conjunto completo.

PRUEBAS DE EVALUACIÓN CONTINUA (PEC)

¿Hay PEC? Si

Descripción

Se deberá realizar una práctica de laboratorio asociada al módulo de contenidos correspondiente a Servicios Web.

Dicha práctica consistirá en desarrollar un servicio REST e integrarlo en un entorno de computación distribuida, relacionado con las tecnologías de computación en la nube o *cloud computing*. El peso relativo de este trabajo en la evaluación final es del 50%.

Criterios de evaluación

Como condición necesaria para aprobar la parte de evaluación continua, se debe superar la práctica de laboratorio (PEC), y debe ser al menos del 25% de la nota final para superar la asignatura.

Para la realización de las prácticas, no es necesaria la presencialidad del alumno en el Centro asociado, ya que se proporcionará el material necesario para realizar dichas prácticas en el curso virtual (incluyendo los entornos virtuales de prácticas necesarios), facilitando que el estudiante pueda desarrollar estas prácticas de manera autónoma.

Ponderación de la PEC en la nota final	50
Fecha aproximada de entrega	Se indica en el curso virtual, de acuerdo a las fechas al cuatrimestre y la planificacion de actividades del plan de trabajo

Comentarios y observaciones

La PEC de desarrollo (práctica de laboratorio) se podrá entregar en dos plazos:

Plazo 1 (convocatoria ordinaria): Recibidas antes del 15 de Enero.

Plazo 2 (convocatoria extraordinaria): Recibidas con posterioridad al 15 de Enero y antes del 20 de Septiembre.

OTRAS ACTIVIDADES EVALUABLES

¿Hay otra/s actividad/es evaluable/s?	No
Descripción	
Criterios de evaluación	
Ponderación en la nota final	0
Fecha aproximada de entrega	
Comentarios y observaciones	

¿CÓMO SE OBTIENE LA NOTA FINAL?

Como condición necesaria para superar la asignatura, se deben superar los criterios de valoración de cada parte de manera independiente: aprobar el examen y aprobar la práctica. Se deben cumplir estos criterios para aprobar la asignatura. El cálculo de la nota de evaluación final se hace en base a la ponderación asociada a cada parte: 50% examen y 50% PEC, por lo que la nota final se calcula de la siguiente manera:

Nota (final) = [Nota (examen) x 0.5] + [Nota (evaluación continua/PEC) x 0.5]
En el caso de no superar los criterios de valoración de alguna de las partes independientes, no se compensarán las notas entre las diferentes partes.
Para la convocatoria extraordinaria, se aplicarán los mismos criterios que para la ordinaria. Las notas del examen o evaluación continua de la convocatoria ordinaria, en caso de ser aptos (nota mayor que cinco) se mantienen para la convocatoria extraordinaria (septiembre). En el caso de las PECs de septiembre, se indicará en el curso virtual las fechas exactas de entrega, que serán posteriores a la semana de exámenes de Septiembre.

BIBLIOGRAFÍA BÁSICA

ISBN(13):9788478290666
Título:COMPUTACIÓN DISTRIBUIDA: FUNDAMENTOS Y APLICACIONESnull
Autor/es:Liu, Mei-Ling ;
Editorial:PEARSON ADDISON-WESLEY

El material didáctico a emplear es el siguiente:

- Mei-Ling Liu. “Computación distribuida. Fundamentos y Aplicaciones”. Ed. Addison Wesley, 2004.
- Apuntes sobre el módulo 7, servicios Web
- Enlaces y recursos sobre los diferentes módulos del curso proporcionados por el equipo docente.

El libro base de Liu recoge la mayor parte del temario de la asignatura y presenta de una forma clara y concisa todos los conceptos clave. En cada capítulo se introducen los

conceptos de una manera sencilla, progresiva y acompañados de ejemplos aclaratorios finalizando con un resumen que recoge los conceptos fundamentales presentados al alumno. El único tema no detallado de forma completa en el libro de texto base se corresponde con el módulo 7 y, por tanto, se proporcionará a los alumnos un documento de aproximadamente 15 páginas con el contenido de dicho tema que será complementado con parte del capítulo 11 del libro. Adicionalmente se le entregarán al estudiante una colección de enlaces y recursos relevantes que completaran la bibliografía básica con contenidos de actualidad sobre la programación distribuida.

BIBLIOGRAFÍA COMPLEMENTARIA

ISBN(13):9781787122734
Título:DISTRIBUTED COMPUTING IN JAVA 92017
Autor/es:Raja Malleswara Rao Pattamsetti ;
Editorial:Packt Publishing

El alumno puede consultar la siguiente bibliografía con el fin de aclarar o extender los conocimientos que debe adquirir a lo largo del curso, y más en concreto en lo concerniente a conectores, RMI y CORBA para la realización de las actividades prácticas:

•Dick Steflik, Prashant Sridharan, Richard Steflik. "Advanced Java Networking" (2nd Ed.) Prentice Hall PTR (2000).

El libro realizar una introducción detallada a las tecnologías de programación en red con Java, abarcando tecnologías comoIDL/CORBA, RMI, JDBC/SQL, JavaBeans, JMAPI, servlets y JavaOS, entre otras. Sin embargo, no es solo un libro para iniciarse en la computación en red con Java porque proporciona al programador con la información necesaria para realizar implementaciones sólidas. Por ejemplo, en el caso de RMI, se muestran las ventajas e inconvenientes de la especificación mientras se está desarrollando una aplicación de calendario distribuido.

•David Reilly, Michael Reilly. “Java™ Network Programming and Distributed Computing”. Ed. Pearson Ed. 2002.

El libro ha sido escrito como una descripción extensa de las características de computación en red de Java, demostrando porqué este lenguaje es único en este ámbito de la programación. Se muestran y explican los APIs correspondientes además de los conceptos relativos a la programación en red y las capacidades necesarias para ser un programador en red efectivo. Para ello se muestran en profundidad varias características: la arquitectura de

Internet y TCP/IP; el sistema de entrada/salida de Java; la tecnología de conectores UDP y TCP para la implementación de clientes y servidores; las ventajas de las aplicaciones multitarea; el protocolo HTTP y las tecnologías Java para el desarrollo de servidores; las tecnologías de computación distribuida CORBA y RMI.

•William Grosso. “Designing and Building Distributed Applications: Java RMI”. Ed. O'Reilly Media, Inc.; 1ª edición, 2001.

El texto contiene ejemplos y experiencias en el diseño e implementación de Java RMI, permitiendo a los desarrolladores que se inician en esta tecnología comprobar que RMI es una tecnología muy poderosa y fácil de usar en la programación distribuida. Además se presentan estrategas de uso para trabajar con la serialización, la multitarea, el registro RMI, conectores y factorías de conectores, activación, descarga dinámica de clases, túneles HTTP, liberación de memoria dinámica distribuida, JNDI y CORBA (todos ellos conceptos avanzados de RMI).

•Darrel Ince. “Developing Distributed and E-Commerce Applications“. Ed. Pearson Ed. 2003.

El creciente interés surgido sobre los sistemas distribuidos, sobre todo en el ámbito de las necesidades de negocio de las compañías, y el uso de tecnologías de red, de forma preferencial sobre Internet, esta siendo una de las áreas de interés del desarrollo de aplicaciones software. El libro describe el desarrollo de las nuevas tecnologías que han emergido o madurado durante años, como los servidores Web, CORBA, HRML, XML y Java. Además el libro discute los paradigmas distribuidos y el modelo cliente/servidor antes de comenzar a detallar tecnologías más específicas como ejemplos de estos conceptos. Al final del libro se detallan conceptos avanzados como la seguridad en Internet, la concurrencia, los agentes y la computación ubicua/móvil.

•Max Goff. “Network Distributed Computing”. Ed. Pearson Ed. 2004.

Los sistemas de computación distribuida en red (Networked distributed computing, NDC) han sido una revolución tecnológica que se ha expandido con el crecimiento de Internet transformando el mundo en un sistema global de información. El libro intenta desmitificar las tecnologías mostrando los errores comunes que se cometen al abordar los desarrollos con los nuevos marcos de desarrollo como J2EE, .NET, CORBA o RMI. Además aborda los protocolos de comunicación y aproximaciones de mensajería para los sistemas distribuidos interoperables, además de realizar una descripción detallada de XML, los servicios Web, JINI y otras tecnologías clave.

•Esmond Pitt, Kathleen Mcniff. “JAVA.RMI The Remote Method Invocation Guide”. Ed. Pearson Ed. 2001.

El libro proporciona una revisión en profundidad de todas las características de RMI, de manera que sea posible aplicar y comprender la tecnología RMI en entornos profesionales. Contiene información detallada de cómo aplicar RMI para conseguir aplicaciones distribuidas reales, mientras aclara y extiende la información oficial encontrada en la especificación RMI.

RECURSOS DE APOYO Y WEBGRAFÍA

Curso virtual

Para alcanzar todos los objetivos propuestos, el curso se va a articular a través de una plataforma especialmente diseñada para facilitar el trabajo colaborativo en Internet (basada en comunidades virtuales), desarrollada por la Sección de Innovación del Centro de Innovación y Desarrollo Tecnológico de la UNED: aLF, ubicada en https://www.innova.uned.es.

Una vez familiarizados con su uso, es importante tener en cuenta que todas las novedades, instrucciones, actividades se van a publicar utilizando este medio, por tanto, el alumno debe entrar en el grupo frecuentemente para ver si hay alguna novedad en el curso. Si, además, tiene activados ciertos avisos, podrá recibir notificaciones en el correo electrónico utilizado para acceder a la plataforma de los mensajes republicados en los foros, los documentos subidos, las citas puestas en el calendario, por lo que tendrá una información instantánea de todo lo que acontece en la plataforma.

Software para prácticas.

Se ubicará en la propia plataforma, en el área correspondiente, o bien se darán los enlaces correspondientes de las ubicaciones originales donde descargar tanto el software como los correspondientes manuales.

IGUALDAD DE GÉNERO

En coherencia con el valor asumido de la igualdad de género, todas las denominaciones que en esta Guía hacen referencia a órganos de gobierno unipersonales, de representación, o miembros de la comunidad universitaria y se efectúan en género masculino, cuando no se hayan sustituido por términos genéricos, se entenderán hechas indistintamente en género femenino o masculino, según el sexo del titular que los desempeñe.