



Asignatura: Programación Paralela y Distribuida

Curso: 2012/2013

TITULACION: Ingeniería Informática

CURSO: 5º

TEMPORALIDAD: 1er. Cuatrimestre

CARÁCTER: Optativa

CRÉDITOS: 6 (3 teóricos y 3 prácticos en laboratorio)

HORARIO CURSO ACTUAL:

Teoría: Lunes (8³⁰-9³⁰) y Martes (8³⁰-9³⁰). Aula I.5

Prácticas laboratorio: Sala NORBA

Grupo 1 (Martes, 15³⁰-17³⁰);

PROFESOR: Juan Hernández Núñez (juanher@unex.es)

WEB: <http://campusvirtual.unex.es>

TUTORÍAS: Lunes, martes y viernes, de 10-11 y 11³⁰ - 12³⁰

OBJETIVOS GENERALES:

La asignatura de Programación Paralela y Distribuida (PPD) abarca una de las áreas de conocimiento en mayor auge de la informática actual. El volumen de información con el que trabajan los sistemas de información actuales, crece día a día, lo que obliga a desarrollar programas y aplicaciones en sistemas distribuidos.

Esta asignatura optativa se complementa con la asignatura Algoritmos Paralelos (también optativa) de cuarto curso. Mientras que en la anterior se estudian diversos tipos de algoritmos en arquitecturas fuertemente acopladas de memoria compartida y distribuida, la asignatura PPD abarca fundamentalmente el estudio de los problemas existentes en la programación distribuida, centrándose en arquitecturas débilmente acopladas. Además, otro de los objetivos fundamentales es el estudio y conocimiento de los modelos y plataformas de componentes.

COMPETENCIAS ESPECÍFICAS DEL TÍTULO RELACIONADAS CON LA ASIGNATURA:

Las competencias del libro blanco del Ingeniero en Informática relacionadas con la asignatura son:

- 42. Estudio de la evolución de las nuevas tecnologías, sobre todo de aquellas que pueden aportar mejoras importantes en los sistemas utilizados en la empresa.
- 45. Analizar y decidir la alternativa óptima de software de mercado a adquirir.
- 108. Garantizar una calidad permanente a través de los procedimientos y de las herramientas. Apoyar las demandas cotidianas de los usuarios.
- 117. Proponer las soluciones de mejora y controlar la puesta en marcha.

OBJETIVOS ESPECÍFICOS:

Esta materia contribuye a la formación del alumno dentro del perfil profesional *Desarrollo de software y aplicaciones* definido por el consorcio Career Space (www.carrer-space.com). Además, la asignatura se encuadra en el bloque temático 1, Tecnologías avanzadas de la programación, cubriendo los contenidos de Tecnologías orientadas a componentes y plataformas middleware, por lo que los objetivos específicos de la asignatura son:

Relacionados con competencias académicas y disciplinares:

- Dotar al alumno de un conocimiento general sobre la Programación distribuida.
- Conocer los distintos problemas que se plantean en la programación de aplicaciones para sistemas distribuidos, aplicando soluciones efectivas a los mismos.
- Conocer distintos modelos de lenguajes para la programación distribuida, entre los que destacan dos grandes familias: modelos basados en compartición de memoria y modelos basados en paso de mensajes.
- Conocer diferentes entornos de programación reales para la programación de aplicaciones de alto rendimiento: PVM y MPI.
- Conocer los principales modelos y plataformas de componentes, con especial énfasis en CORBA y SCA así como el sistema de objetos distribuidos de Java/RMI
- Conocer, evaluar, seleccionar y desarrollar aplicaciones empresariales utilizando arquitecturas software actuales: arquitecturas multicapas, arquitecturas orientadas a servicios, plataformas middleware y marcos de trabajo (frameworks) basados en tecnologías de componentes.
- Diseñar y desarrollar aplicaciones distribuidas conectando diferentes plataformas de presentación (dispositivos móviles, aplicaciones cliente, etc), con distintas tecnologías que sustenten la lógica de negocio y con servidores de bases de datos.

Relacionados con otras competencias personales y profesionales:

- Planificar y organizar el trabajo personal.
- Comunicar de forma efectiva (en expresión y comprensión) oral y escrita, conocimientos, procedimientos, resultados e ideas relacionadas con las TIC, con especial énfasis, en la redacción de documentación técnica.
- Tener iniciativa y ser resolutivo, aportando soluciones efectivas a los problemas planteados incluso en situaciones de falta de información y/o con restricciones temporales y/o de recursos.
- Mostrar una adecuada capacidad de relación interpersonal
- Encontrar, analizar, criticar, relacionar, estructurar y sintetizar información proveniente de diversas fuentes, así como integrar ideas y conocimientos.
- Ser capaz de argumentar y justificar lógicamente las decisiones, sabiendo aceptar otros puntos de vista.

PROGRAMA TEÓRICO:

- I. *Introducción.*
Tema 1. Introducción a la Programación Distribuida.
- II. *Problemas de la Programación Distribuida.*
Tema 2. Problemas de la programación distribuida.
 - 2.1. Introducción.
 - 2.2. Técnicas básicas de programación distribuida.
 - 2.3. Problemas básicos en la programación distribuida.
 - 2.3.1 La exclusión mutua distribuida.
 - 2.3.2 La Detección de terminación distribuida.
 - 2.3.3 La detección del deadlock en la programación distribuida
 - 2.3.4 Manejo de datos distribuido.
- III. *Modelos y Lenguajes basados en paso de mensajes.*
Tema 3. Lenguajes basados en objetos.
 - 3.1. Características de los lenguajes basados en objetos.Tema 4. Librerías de procesamiento Paralelo.
 - 4.1 El entorno PVM
 - 4.2 El entorno MPI
- IV. *Modelos y Lenguajes basados en compartición de memoria.*
Tema 5. Espacios de tuplas. El modelo de Linda. JavaSpaces
- V. *Plataformas de componentes distribuidos.*
Tema 6. Plataformas de componentes. Principios y arquitecturas.
Tema 7. Java RMI
Tema 8. El estandar CORBA.
- VI. *Arquitecturas orientadas a servicios.*
Tema 9. El estandar SCA.

PROGRAMA DE PRÁCTICAS:

- ❖ Programación en PVM
 - Desarrollo de aplicaciones en PVM.
 - Configuración de máquina virtual
 - Diseño de aplicaciones SPMD y MPMD en PVM
- ❖ Programación en Java RMI.
 - Estructura de las aplicaciones en Java RMI.
 - Diseño de servicios remotos.
 - Acceso a servicios remotos.
- ❖ Programación en CORBA (ICE)
 - Estructura de las aplicaciones en CORBA.
 - Diseño de servicios remotos.
 - Acceso a servicios remotos.
 - Servicio de Nombres.
- ❖ Desarrollo en Service Component Architecture (SCA)
 - Estructura de las aplicaciones en SCA.
 - Diseño de servicios con SCA

METODOLOGÍAS/ACTIVIDADES:

La asignatura “Programación paralela y distribuida” busca la participación activa y continuada de los estudiantes, quienes deberán hacer frente a nuevos retos que se irán proponiendo a lo largo de la asignatura.

La asignatura se divide en 3 bloques:

- Bloque de teoría
- Bloque de laboratorio
- Bloque de proyecto

En el bloque de teoría se combinan las clases expositivas con la resolución de problemas individualmente y en grupo con metodologías activas de aprendizaje.

Para el desarrollo correcto del bloque de laboratorio, y al inicio del curso, cada alumno dispondrá de un usuario-password de acceso al servidor de prácticas. Las sesiones de laboratorio estarán a disposición de los alumnos antes del inicio de cada sesión. Cada sesión dispone de un guión que contiene los trabajos que se deben desarrollar. El alumno puede realizar parte de la sesión de forma remota, de manera que en el momento de la sesión de laboratorio, sea capaz de implantar la solución software real.

En el bloque de proyecto los estudiantes desarrollarán un estudio, evaluación e implantación de una tecnología de programación distribuida concreta propuesta por el profesor. Siguiendo un método de aprendizaje basado en problemas, el alumno deberá dominar la tecnología en cuestión para la realización de un trabajo y exposición pública a final de curso.

Todas las tareas y actividades realizadas en cada uno de los bloques se apoyará en los materiales desarrollados por el profesor y que están disponibles en el aula virtual de la asignatura.

Medios materiales utilizados

- Pizarra
- Cañón de video
- Ordenador
- Internet

Materiales y recursos utilizados

Los materiales y recursos utilizados están en versión electrónica, a la que se puede acceder bien a través del aula virtual o bien por medio del servidor de la asignatura.

- Trasparencias para cada tema del programa y recursos complementarios (aula virtual)
- Guiones de las sesiones de laboratorio (servidor dedicado a la asignatura)

RECOMENDACIONES

Conocimientos previos

Para aquellos alumnos que deseen cursar esta asignatura, es **recomendable** haber cursado antes las siguientes asignaturas optativas:

- *Administración de sistemas operativos*
- *Sistemas Operativos Distribuidos*
- *Algoritmos Paralelos*

Además, es **imprescindible** tener sólidos conocimientos del lenguaje de programación Java, así como experiencia y dominio del sistema operativo Linux

En la página web de la asignatura, aparecen enlaces a páginas de interés para esta asignatura sobre fuentes bibliográficas y tutoriales, así como información donde obtener software gratuito o con licencia del Departamento para el alumno sobre: PVM, EJB, CORBA, Java/RMI, y software de desarrollo bajo .NET.

Horas de estudio recomendadas

Se recomienda asistir a clase de teoría. La asistencia a clase de prácticas es obligatoria (ver apartado relativo a los criterios de evaluación). Se recomienda una dedicación no presencial de, aproximadamente, 2 horas semanales. Durante la preparación del trabajo habrá que aumentar la dedicación hasta, aproximadamente, unas 10 horas semanales.

NORMAS GENERALES:

- ❖ Durante los primeros 15 días de octubre, los alumnos deberán actualizar, en su caso, el perfil del estudiante en el aula virtual, incluyendo fotografía actualizada y clara
- ❖ Cada grupo no tendrá más de 15 alumnos.
- ❖ La asistencia a las clases prácticas es obligatoria.

CRITERIOS DE EVALUACIÓN:

- ❖ Los alumnos tendrán que realizar la exposición de un trabajo durante los meses de diciembre y enero (ver bloque proyecto en la sección metodología/actividades). La exposición constará de dos partes:
 - *teórica*, donde se presente la tecnología en cuestión. Es aconsejable que las exposiciones teóricas tengan, al menos, tres partes diferenciadas: introducción, desarrollo, conclusiones. El apartado de desarrollo puede (y suele) dividirse en subapartados
 - *práctica*, donde se presenta de forma real la tecnología elegida. Debe realizarse la instalación en los equipos para que el resto de alumnos puedan también practicar con la plataforma en cuestión. Hay que organizar una demo y, al menos, un ejercicio a resolver por el resto de alumnos. Es muy aconsejable realizar las pruebas prácticas en la sala NORBA con antelación suficiente a la exposición, de manera que el software a instalar y las demos estén completamente probadas para el día de la exposición.
- ❖ Para aprobar la asignatura es obligatoria la asistencia a todas las exposiciones realizadas y no tener más de tres faltas a las sesiones de prácticas

BIBLIOGRAFÍA:

Básica

- ❖ Michel Raynal, *Distributed Algorithms and Protocols*, Ed. John Wiley & Sons, 1992.
- ❖ M. Ben-Ari, *Principles of Concurrent and Distributed Programming*, Prentice-Hall, 1990
- ❖ Nancy Lynch, *Distributed Algorithms*, Ed. Morgan Kaufmann Publishers, 1996
- ❖ Randy Chow et al., *Distributed Operating Systems and Algorithms*, Addison-Wesley, 1997
- ❖ Al Geist et al., *PVM: Parallel Virtual Machine*, 1991
- ❖ Robert Orfali & Dan Harkey, *Client/Server programming with Java and CORBA*, (2ª edición), John Wiley and Sons, 1998
- ❖ SCA - OASIS Tutorial. Anish Karmarkar, Ashok Malhotra & David Booz

Adicional

- ❖ Aula virtual de la asignatura
- ❖ <http://www.eclipse.org>
- ❖ <http://java.sun.com/>
- ❖ <http://www.zeroc.com/ice.html>