

Grado en Ingeniería Informática

30212 - Programación de sistemas concurrentes y distribuidos

Guía docente para el curso 2014 - 2015

Curso: 2, Semestre: 1, Créditos: 6.0

Información básica

Profesores

- **Joaquín Antonio Ezpeleta Mateo** ezpeleta@unizar.es
- **Pedro Javier Álvarez Pérez-Aradros** alvaper@unizar.es
- **Jesús Gallardo Casero** jesus.gallardo@unizar.es
- **Raquel Trillo Lado** raqueltl@unizar.es

Recomendaciones para cursar esta asignatura

La asignatura es la continuación natural de las asignaturas de programación presentadas en el primer año de estudios, Programación I y Programación II. Es necesario que el alumno haya adquirido soltura en el desarrollo de programas secuenciales correctos. Es también recomendable que se encuentre cursando las asignaturas de Sistemas Operativos y Teoría de la Computación, ya que comparten algunos temas clave.

Actividades y fechas clave de la asignatura

El calendario detallado de las diversas actividades a desarrollar se establecerá una vez que la Universidad haya aprobado el calendario académico del curso correspondiente. En cualquier caso, las fechas importantes serán anunciadas con la suficiente antelación.

Inicio

Resultados de aprendizaje que definen la asignatura

El estudiante, para superar esta asignatura, deberá demostrar los siguientes resultados...

1:

El estudiante terminará con un conocimiento profundo de cuáles son las características específicas de los sistemas concurrentes y distribuidos.

2:

Conocerá los problemas generados por el acceso concurrente a datos y recursos, así como las soluciones

conceptuales y tecnológicas que se han dado a los mismos

3:

Conocerá las características de los sistemas distribuidos, los retos que plantea y las soluciones que se han planteado para los mismos.

4:

Tendrá nociones de qué son los sistemas tiempo real, y sistemas basados en eventos.

5:

Conocerá herramientas para el diseño y programación de programas con características concurrentes y/o distribuidas.

Introducción

Breve presentación de la asignatura

En la mayoría de los sistemas reales vamos a encontrar distintas entidades que tienen que convivir, lo que supone interaccionar. En el caso de las aplicaciones informáticas ocurre lo mismo: distintos programas tienen que ejecutarse en un mismo entorno, compitiendo por recursos comunes (como ancho de banda o memoria del computador) o colaborando para conseguir objetivos conjuntos (como navegadores accediendo a un sitio web o consultando horarios de trenes, por ejemplo). Diseñar este tipo de programas tiene la dificultad añadida, en comparación con la programación secuencial vista en primer curso, de que hay que considerar esas interacciones. El objetivo de la asignatura es poner de relieve cuáles son los nuevos problemas que aparecen en esta clase de sistemas así como presentar las técnicas utilizadas para el desarrollo de aplicaciones correctas.

Contexto y competencias

Sentido, contexto, relevancia y objetivos generales de la asignatura

La asignatura y sus resultados previstos responden a los siguientes planteamientos y objetivos:

En esta asignatura el alumno aprenderá a enfrentarse al diseño de programas en los que un conjunto de procesos deben sincronizarse, ya sea mediante mecanismos de memoria compartida como a través de redes de comunicaciones, llegando a poder razonar sobre la corrección de la solución propuesta.

Contexto y sentido de la asignatura en la titulación

Esta asignatura, junto con las de programación del primer año y la asignatura de Teoría de la Computación son la plataforma sobre la que se va a asentar todo el desarrollo de las asignaturas posteriores que trabajen en el desarrollo de sistemas. Actualmente es difícil encontrar una aplicación “software” que no incluya características propias de sistemas concurrentes, distribuidos y de tiempo real.

Al superar la asignatura, el estudiante será más competente para...

1:

Contribuye a que el estudiante adquiera las siguientes competencias generales comunes a la rama de informática:

- Conocer y aplicar los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
- Conocer y aplicar las características, funcionalidades y estructura de los Sistemas Distribuidos, las Redes de Computadores e Internet y diseñar e implementar aplicaciones basadas en ellas.
- Conocer y aplicar los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.

- Conocer y aplicar los principios, metodologías y ciclos de vida de la ingeniería de software.

2:

Adicionalmente, contribuye a adquirir las siguientes competencias generales/transversales:

- Capacidad para resolver problemas y tomar decisiones con iniciativa, creatividad y razonamiento crítico.
- Capacidad para usar las técnicas, habilidades y herramientas de la Ingeniería necesarias para la práctica de la misma.
- Capacidad para aprender de forma continua y desarrollar estrategias de aprendizaje autónomo.
- Capacidad para aplicar las tecnologías de la información y las comunicaciones en la Ingeniería.

Importancia de los resultados de aprendizaje que se obtienen en la asignatura:

En la época en que nos encontramos, todo ingeniero informático debe ser capaz de diseñar sistemas concurrentes y distribuidos. Debe también entender cuáles son las características importantes a considerar en sistemas en que el tiempo real es un requisito. Una base sólida en estos aspectos es imprescindible para poder desenvolverse en el mundo profesional.

Evaluación

Actividades de evaluación

El estudiante deberá demostrar que ha alcanzado los resultados de aprendizaje previstos mediante las siguientes actividades de evaluación

1:

En la Escuela de Ingeniería y Arquitectura de Zaragoza:

Prueba práctica en el laboratorio (20%): El objetivo de esta prueba es evaluar los conocimientos y destrezas que han adquirido los alumnos en las sesiones prácticas de laboratorio. Estas sesiones consistirán en la realización utilizando el computador de una serie de ejercicios relacionados con los contenidos de la asignatura. Opcionalmente, un alumno podrá ir entregando durante el semestre, en las fechas indicadas por los profesores, las soluciones implementadas para cada uno de los ejercicios planteados para estas sesiones. Estas soluciones serán evaluadas, proporcionando al alumno la correspondiente realimentación. La entrega en fecha de estas soluciones eximirá al estudiante de realizar la prueba final práctica en el laboratorio. Su correspondiente calificación será obtenida a partir de las evaluaciones de los ejercicios entregados.

2:

Realización y defensa de trabajos/proyectos prácticos en grupo (15%): Durante esta actividad se les planteará a los alumnos un proyecto de programación, relacionado con los contenidos de la asignatura, que deberán realizar en grupo. Cada grupo deberá comprender los requisitos del proyecto, evaluar posibles alternativas de solución y, finalmente, implementar la que considere más adecuada. El proyecto resultado de cada grupo deberá ser entregado y defendido por sus integrantes en las fechas establecidas por los profesores. Se valorará la corrección de la solución respecto a los requisitos iniciales y el grado de justificación de la solución alcanzada.

La defensa en fecha del proyecto eximirá al estudiante de realizar una prueba final práctica equivalente en el laboratorio.

3:

Prueba escrita (65%): En esta prueba se plantearán cuestiones y/o problemas relacionados con el programa impartido en la asignatura. Su tipología y complejidad será similar a los presentados en las sesiones de aula y laboratorio. En general, se valorará la calidad y claridad de las respuestas, así como las estrategias de solución planteadas por los alumnos.

4:

La **calificación final** de la asignatura se obtendrá como la media ponderada de las tres partes, teniendo en cuenta que es necesario llegar a obtener, al menos, 5.0 puntos sobre 10.0 en cada una de ellas. En caso de no alcanzar ese mínimo en alguna de las partes, la calificación global de la asignatura será la mínima entre 4.0 y el resultado de ponderar con los porcentajes de cada parte.

5:

Validez de las calificaciones. La calificación obtenida en cualquiera de las partes superadas en la primera convocatoria tendrá validez para todas las convocatorias del curso.

6: En la Escuela Universitaria Politécnica de Teruel:

La nota final de la asignatura en la convocatoria ordinaria se divide de la siguiente forma:

- **Trabajo en grupo.** **10%** de la nota final. Se realizará en parejas o, excepcionalmente, grupos de 3 personas. Consistirá en proponer un problema de programación concurrente, resolverlo y exponerlo en clase. En el examen final habrá una parte de recuperación de esta actividad.
- **Prácticas.** **20%** de la nota final. Habrá 4 prácticas entregables a lo largo del curso, tres referidas a la parte de concurrentes y una a la de distribuidos. En caso de no superar las prácticas, deberá hacerse una parte adicional de prácticas en el examen final.
- **Teoría y ejercicios.** **70%** de la nota final. Evaluable en el examen final, que constará de una parte de teoría y otra de problemas. A mitad del curso habrá un examen parcial, que valdrá un 15% de la nota final, y que liberará de materia para el final en caso de superarse. Si no se supera, esa nota no cuenta y toda la nota de teoría será la del examen final.

Quien, habiendo superado el trabajo en grupo, las prácticas o el parcial, quiera mejorar en el examen final la nota obtenida, puede hacer las preguntas correspondientes de recuperación. En ese caso, la nota que quedará al final será la mayor entre la obtenida en su momento y la obtenida en el examen final.

En cuanto a la **convocatoria extraordinaria (septiembre)**, la nota final será la nota del examen extraordinario, teniendo en cuenta que ese examen tendrá una parte de prácticas que valdrá el 20% de la nota total. Aquellos que hubieran aprobado la parte de prácticas en la convocatoria ordinaria mantendrán la nota y no tendrán que hacer dicha parte de prácticas en el examen extraordinario. En el examen extraordinario, la nota del parcial y del trabajo no se mantiene. En cuanto a las prácticas, rige el mismo mecanismo que en el examen ordinario.

Actividades y recursos

Presentación metodológica general

El proceso de aprendizaje que se ha diseñado para esta asignatura se basa en lo siguiente:

El aprendizaje se obtendrá a partir de tres entradas: la sesiones explicativas del profesorado, los trabajos desarrollados en las sesiones prácticas y el trabajo del alumno (individual o en grupo).

Para el desarrollo de los dos primeros tipos de actividades, el alumno deberá haber hecho un trabajo previo. En el primer caso, el repaso y estudio de los contenidos planteados en sesiones anteriores. En el caso de las sesiones prácticas, el alumno deberá acudir con el enunciado del trabajo meditado y trabajado, así como la lista de dudas o aclaraciones que requieran la intervención del profesor. Además, será también en una sesión de prácticas donde el alumno deberá presentar y defender ante el profesor su trabajo realizado en sesiones de laboratorio anteriores.

Actividades de aprendizaje programadas (Se incluye programa)

El programa que se ofrece al estudiante para ayudarle a lograr los resultados previstos comprende las siguientes actividades...

1:

Las actividades se organizarán en base a clase presencial, resolución de problemas (con y sin tutela del profesor), prácticas de laboratorio, trabajo en equipo y actividades de evaluación.

Planificación y calendario

Calendario de sesiones presenciales y presentación de trabajos

El calendario de la asignatura estará definido por el centro en el calendario académico del curso correspondiente.

Programa de la asignatura

En la Escuela de Ingeniería y Arquitectura de Zaragoza:

El programa de la asignatura consta de los siguientes temas (entre paréntesis se especifica el número de horas de trabajo en el aula y en la solución de problemas, respectivamente):

- Introducción a la programación concurrente (4h aprox.)
- Modelado y análisis de sistemas concurrentes (6h aprox.)
- El problema de la sección crítica (5h aprox.)
- Semáforos (8h aprox.)
- Monitores (6h aprox.)
- Introducción a la programación distribuida (3h aprox.)
- Coordinación basada en espacios de tuplas (3h aprox.)
- Algoritmos distribuidos (7h aprox.)
- Introducción a la programación dirigida por eventos y tiempo real (3h aprox.)

También se realizarán las siguientes prácticas:

- La programación concurrente. Threads y datos compartidos. Problemas de interferencias (2h)
- Solución al problema de la sección crítica (2h)
- Programación con semáforos (2h)
- Programación con monitores (2h)
- Programación de sistemas distribuidos (4h)
- Análisis del proyecto en grupo (2h)

En la Escuela Universitaria Politécnica de Teruel:

El programa de la asignatura consta de los siguientes temas (entre paréntesis se especifica el número de horas de trabajo en el aula y en la solución de problemas, respectivamente):

- Introducción a la programación concurrente (2h aprox.)
- El problema de la sección crítica (2h aprox.)
- El lenguaje de programación Ada (2h aprox.)
- Programación concurrente en Java (2h aprox.)
- Semáforos (5h aprox.)
- Regiones críticas condicionales (2h aprox.)

- Monitores (8h aprox.)
- Comunicación directa. Citas (7h aprox.)
- Activación y terminación de procesos (2h aprox.)
- Introducción a los Sistemas Distribuidos (3h aprox.)
- Sockets (2h aprox.)
- Sincronización de Sistemas Distribuidos (3h aprox.)
- Introducción a la programación dirigida por eventos y tiempo real (2h aprox.)
- Exposición de trabajos en grupo (3h aprox.)

También se realizarán las siguientes prácticas:

- Lenguaje Ada básico (2h)
- Programación concurrente en Java y Ada (2h)
- Programación con semáforos (2h)
- Programación con monitores (2h)
- Programación con citas (2h)
- Programación con sockets (2h)
- Defensa de prácticas (3h)

Trabajo del estudiante

- 60 horas, aproximadamente, de actividades presenciales (sesiones en el aula teóricas y de problemas y sesiones en el laboratorio)
- 30 horas de trabajo en grupo
- 55 horas de trabajo y estudio individual efectivo (estudio de apuntes y textos, resolución de problemas, preparación de clases y prácticas, desarrollo de programas, etc.)
- 5 horas dedicadas a distintas pruebas de evaluación

Bibliografía

Bibliografía recomendada

- M. Ben-Ari
Principles of Concurrent and Distributed Programming
Addison-Wesley, 2006
- J. Magee, J. Kramer
Concurrency. State models and Java programming
Wiley, 2006
- M.L. Liu
Computación distribuida. Fundamentos
Ed. Pearson- Addison Wesley, 2004

Referencias bibliográficas de la bibliografía recomendada

Escuela Universitaria Politécnica

- Ben-Ari, M.. Principles of concurrent and distributed programming / M. Ben-Ari . - 2nd ed. Harlow (England) : Pearson Education, 2006
- Liu, M. L. (Mei-Ling L.). Computación distribuida : fundamentos y aplicaciones / M. L. Liu ; traducción, José María Peña Sánchez ... [et al.] . Madrid [etc.] : Pearson Educación, cop. 2004
- Magee, Jeff. Concurrency : state models & Java programs / Jeff Magee & Jeff Kramer Chichester [etc.] : John Wiley & Sons, cop. 1999

Escuela Politécnica Superior

1. Ben-Ari, M.. Principles of concurrent and distributed programming / M. Ben-Ari . - 2nd ed. Harlow (England) : Pearson Education, 2006
2. Magee, Jeff. Concurrency : state models & Java programs / Jeff Magee & Jeff Kramer Chichester [etc.] : John Wiley & Sons, cop. 1999

- 3. Liu, M.L. *Computación distribuida : fundamentos y aplicaciones* / M. L. Liu ; traducción, José María Peña Sánchez ... [et al.] . Madrid [etc.] : Pearson Educación, cop. 2004