

Curso : 2018/19

30212 - Programación de sistemas concurrentes y distribuidos

Información del Plan Docente

Año académico:	2018/19
Asignatura:	30212 - Programación de sistemas concurrentes y distribuidos
Centro académico:	110 - Escuela de Ingeniería y Arquitectura 326 - Escuela Universitaria Politécnica de Teruel
Titulación:	439 - Graduado en Ingeniería Informática 443 - Graduado en Ingeniería Informática
Créditos:	6.0
Curso:	443 - Graduado en Ingeniería Informática: 2 439 - Graduado en Ingeniería Informática: 2
Periodo de impartición:	Primer Semestre
Clase de asignatura:	Obligatoria
Módulo:	---

Información Básica

Objetivos 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.

Recomendaciones para cursar la 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.

Competencias y resultados de aprendizaje

Competencias

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

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.

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 continuada y desarrollar estrategias de aprendizaje autónomo.
- Capacidad para aplicar las tecnologías de la información y las comunicaciones en la Ingeniería.

Resultados de aprendizaje

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

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

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

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

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

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

Importancia de los resultados de aprendizaje

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

Tipo de pruebas y su valor sobre la nota final y criterios de evaluación para cada prueba

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

En la Escuela de Ingeniería y Arquitectura de Zaragoza:

- Prueba práctica en el laboratorio (15%): El objetivo de esta prueba es evaluar los conocimientos y destrezas que han adquirido los alumnos en las sesiones prácticas de laboratorio.
- Realización y defensa de un trabajo práctico 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.

- Prueba escrita (70%): 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.

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, 4.0 puntos sobre 10.0 en la prueba práctica en el laboratorio y 5.0 sobre 10.0 en el trabajo en grupo y en la prueba escrita. 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.

La evaluación de la asignatura en la convocatoria de septiembre constará de dos pruebas:

- Prueba escrita (70%), de características similares a la de la primera convocatoria
- Prueba en el laboratorio (30%), en la que se evaluarán los conocimientos y destrezas de naturaleza práctica en la implementación de sistemas concurrentes y distribuidos.

La calificación final de la asignatura en esta convocatoria se obtendrá como la media ponderada de las dos partes.

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. 15% de la nota final. 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 varias prácticas entregables a lo largo del curso, referidas a distintos temas del mismo. En el examen final habrá una parte de recuperación de esta actividad.
- Teoría y ejercicios. 65% de la nota final. Evaluable en el examen final, que constará de una parte de teoría aplicada y otra de problemas.

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, 4 puntos sobre 10 en las prácticas y 4 puntos sobre 10 en la prueba escrita. En caso de no alcanzar ese mínimo en alguna de esas partes, la calificación global de la asignatura será la mínima entre 4.0 y el resultado de ponderar con los porcentajes de cada parte.

En cuanto a la convocatoria de 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 la convocatoria extraordinaria, la nota del trabajo no se mantiene.

Metodología, actividades de aprendizaje, programa 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: las 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

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

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.

Por tipos de actividad, la dedicación del alumno se organiza en torno a:

- * 60 horas de actividades presenciales (sesiones en el aula teóricas y de problemas y sesiones en el laboratorio), aproximadamente
- * 30 horas de trabajo en grupo, aproximadamente
- * 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.), aproximadamente
- * 5 horas dedicadas a distintas pruebas de evaluación, aproximadamente

Programa

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)

Planificación de las actividades de aprendizaje y calendario de fechas clave

El calendario concreto para cada una de las actividades será detallado antes del inicio del curso de acuerdo con el calendario académico y la distribución de laboratorios para la realización de las sesiones prácticas. Esta planificación será acorde con la duración de las actividades detalladas en el apartado 4.3 de esta guía docente.

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.

Bibliografía y recursos recomendados

Zaragoza:

BB

Ben-Ari, M.. Principles of concurrent and distributed programming / M. Ben-Ari . - 2nd ed. Harlow (England) : Pearson Education, 2006

BB

Williams, Anthony. C++ Concurrency in action : practical multithreading / Anthony Williams . Shelter Island (NY) : Manning, cop. 2012

BC

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

BB

Ben-Ari, M.. Principles of concurrent and distributed programming / M. Ben-Ari . - 2nd ed. Harlow (England) : Pearson Education, 2006

BB

Liu, M. L. (Mei-Ling L.). Computación distribuida : fundame
aplicaciones / M. L. Liu ; traducción, José María Peña Sán
[et al.] . Madrid [etc.] : Pearson Educación, cop. 2004

BB

Magee, Jeff. Concurrency : state models & Java programs
Magee & Jeff Kramer Chichester [etc.] : John Wiley & Sons
1999