Guía docente de la asignatura

Sistemas Concurrentes y Distribuidos (2161134)

Fecha de aprobación: **25/06/2024**

Grado	Grado en Ingeniería Informática y Administración y Dirección de Empresas	**Rama**	Ciencias Sociales y Jurídicas
Módulo	Formación Específica de Rama	**Materia**	Sistemas Operativos, Sistemas Distribuidos y Redes

Curso	3º	**Semestre**	1º	**Créditos**	6	**Tipo**	Obligatoria

PRERREQUISITOS Y/O RECOMENDACIONES

Los alumnos no tendrán que tener asignaturas, materias o módulos aprobados como requisito indispensable para cursar el módulo. No obstante se recomienda la superación de los contenidos y adquisición de competencias de las materias de formación básica, teniendo especial importancia la superación de las materias de:

- Fundamentos de Programación.
- Fundamentos de software.
- Metodología de la Programación.

BREVE DESCRIPCIÓN DE CONTENIDOS (Según memoria de verificación del Grado)

- Exclusión mutua, sincronización y comunicación entre procesos.
- Propiedades de seguridad y vivacidad.
- Algoritmos para modelos basados en memoria compartida y paso de mensajes.
- Semáforos y monitores.
- Bibliotecas de programación concurrente y distribuida.
- Técnicas para el diseño de aplicaciones de tiempo real.

COMPETENCIAS ASOCIADAS A MATERIA/ASIGNATURA

COMPETENCIAS GENERALES

- CG06 - Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes.

COMPETENCIAS ESPECÍFICAS

- CE12 - Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
- CE14 - Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
- CE17 - 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.
- CE20 - Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.

COMPETENCIAS TRANSVERSALES

- CT03 - Capacidad para el uso y aplicación de las TIC en el ámbito académico y profesional.

RESULTADOS DE APRENDIZAJE (Objetivos)

- Comprender la importancia de la programación concurrente en las aplicaciones de hoy en día.
- Identificar las principales características de los distintos tipos de sistemas concurrentes que existen.
- Conocer y entender los problemas que plantea el desarrollo de programas concurrentes y que no aparecen en la programación secuencial.
- Entender los conceptos de sincronización y exclusión mutua entre procesos.
- Identificar las propiedades de seguridad y vivacidad que un sistema concurrente debe cumplir y ser capaz de razonar si dichas propiedades se cumplen.
- Conocer los principales modelos de programación concurrente, paralela y distribuida.
- Adquirir experiencia y conocimiento en los mecanismos de sincronización y comunicación que se utilizan en la actualidad para desarrollar programas concurrentes tanto para sistemas de memoria compartida como para sistemas distribuidos.
- Entender el funcionamiento de semáforos y monitores como mecanismos de sincronización para memoria compartida y comprender cómo se pueden resolver problemas de programación concurrente usando monitores.
- Ser capaz de desarrollar algoritmos para sistemas basados en memoria compartida y para sistemas distribuidos que resuelvan problemas modelo en programación concurrente.
- Conocer y ser capaz de usar bibliotecas y plataformas estandarizadas para la implementación de programas concurrentes basados en memoria compartida y para sistemas distribuidos.
- Conocer las técnicas más destacadas para el diseño de sistemas de tiempo real.

PROGRAMA DE CONTENIDOS TEÓRICOS Y PRÁCTICOS

TEÓRICO

Tema 1. Introducción a la Programación Concurrente

1. Conceptos básicos y motivación.

2. Modelo abstracto de la Programación Concurrente. Consideraciones sobre el hardware.
3. Exclusión mutua y sincronización.
4. Propiedades de sistemas concurrentes.
5. Verificación de programas concurrentes.
6. Resolución de ejercicios.

Tema 2. Algoritmos y mecanismos de sincronización basados en memoria compartida

1. Introducción a la sincronización en sistemas con memoria compartida
2. Algoritmos básicos de exclusión mutua en sistemas con memoria compartida.
3. Soluciones hardware para la exclusión mutua. Cerrojos.
4. Semáforos. Estructura y operaciones.
5. Monitores como mecanismo de alto nivel. Definición y características. Semántica de las señales de los monitores. Implementación de monitores.
6. Resolución de ejercicios.

Tema 3. Sistemas basados en paso de mensajes

1. Mecanismos básicos en sistemas basados en paso de mensajes.
2. Modelos y lenguajes de programación distribuida.
3. Bibliotecas de paso de mensajes y patrones de interacción.
4. Mecanismos de alto nivel en sistemas distribuidos. RPC y RMI.
5. Resolución de ejercicios.

Tema 4. Introducción a los sistemas de tiempo real

1. Concepto de sistema de tiempo real. Medidas de tiempo y modelo de tareas.
2. Planificación de tareas periódicas con asignación de prioridades.
3. Modelos generales y específicos de tareas.
4. Resolución de ejercicios.

PRÁCTICO

Seminarios/Talleres:
- Seminario práctico 1: Introducción a la programación multihebra usando semáforos.
- Seminario práctico 2: Introducción a la programación multihebra con monitores.
- Seminario práctico 3: Introducción al uso de una interfaz de paso de mensajes.

Prácticas de Laboratorio:
- Práctica 1: Resolución de problemas de sincronización con semáforos.
- Práctica 2: Programación de monitores con hebras.
- Práctica 3: Programación de aplicaciones distribuidas.
- Práctica 4: Programación de tareas periódicas con prioridades.

BIBLIOGRAFÍA

BIBLIOGRAFÍA FUNDAMENTAL

- J. T. Palma, C. Garrido, F. Sánchez, A. Quesada. Programación Concurrente. Thomson-Paranifo. 2003.
- G. R. Andrews. Foundations of Multithreaded, Parallel, and Distributed Programming. Addison Wesley, 2000.

- F. Almeida, D. Gimenez, J. M. Mantas, A.M. Vidal . Introduccion a la Programacion Paralela. Paraninfo Cengage Learning, 2008.
- A. Burns, A. Wellings. Sistemas de Tiempo Real y Lenguajes de Programación (3ª Edición). Addison Wesley, 2003.
- A. Wiliams. C++ Concurrency in Action, 2nd Edition. Manning Publications. 2019
- M. Capel. Programación Concurrente y en Tiempo Real. Fundamentos y aplicaciones. Garceta grupo editorial, 2022

BIBLIOGRAFÍA COMPLEMENTARIA

- C. Breshears. The Art of Concurrency: A Thread Monkey's Guide to Writing Parallel Applications. O'Reilly Media. 2009.
- N.A. Lynch. Distributed Algorithms. Morgan Kaufmann. 1996.
- George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair. Distributed Systems: Concepts and Design (5ª Edición). Addison-Wesley, 2011.
- V. Kumar , A. Grama, A. Gupta, G. Karypis. Introduction to Parallel Computing. Benjamin/Cummings Publishing Company, 2003

ENLACES RECOMENDADOS

A principio de curso se avisará de la plataforma web y páginas web auxiliares donde se encontrarán los enlaces recomendados para la asignatura.

METODOLOGÍA DOCENTE

- MD01 - Lección Magistral (Clases Teóricas-Expositivas)
- MD02 - Actividades Prácticas (Resolución de Problemas, Resolución de Casos Prácticos, Desarrollo de Proyectos, Prácticas en Laboratorio, Taller de Programación, Aula de Informática, Prácticas de Campo).
- MD03 - Seminarios (Debates, Demos, Exposición de Trabajos Tutelados, Conferencias, Visitas Guiadas, Monografías).
- MD04 - Actividades no presenciales Individuales.
- MD05 - Actividades no presenciales Grupales.
- MD06 - Tutorías Académicas.

EVALUACIÓN (instrumentos de evaluación, criterios de evaluación y porcentaje sobre la calificación final)

EVALUACIÓN ORDINARIA

Todo lo relativo a la evaluación se regirá por la Normativa de evaluación y calificación de los estudiantes vigente en la Universidad de Granada.
Preferentemente, la evaluación se ajustará al sistema de evaluación continua del aprendizaje del estudiante siguiendo el artículo 7 de la anterior Normativa.
Se utilizarán alguna o algunas de las siguientes técnicas de evaluación:

- Para la parte teórica se realizarán un mínimo de dos sesiones de evaluación y un máximo de tres, así como entregas de ejercicios sobre el desarrollo y los resultados de las

actividades propuestas.
- Para la parte práctica se realizarán sesiones de evaluación prácticas en el laboratorio, resolución de problemas y se podrá evaluar el desarrollo de proyectos (individuales o en grupo), así como el trabajo desarrollado por los alumnos teniendo en cuenta las entregas de sus informes/memorias, o en su caso, las entrevistas personales con los mismos sobre dicho trabajo.
- En el caso de la evaluación continua, los seminarios se podrán evaluar teniendo en cuenta la asistencia, los problemas propuestos que hayan sido resueltos y entregados por los alumnos, en su caso, las entrevistas efectuadas durante el curso y la presentación oral de los trabajos desarrollados.

La calificación global corresponderá por tanto a la puntuación ponderada de los diferentes aspectos y actividades que integran el sistema de evaluación. Por tanto, el resultado de la evaluación será una calificación numérica obtenida mediante la suma ponderada de las calificaciones correspondientes a una parte teórica, una parte práctica y, en el caso de la evaluación continua, una parte relacionada con el trabajo autónomo de los alumnos, los seminarios impartidos y el aprendizaje basado en proyectos. La adaptación del sistema de evaluación general propuesto a las características de esta asignatura, con indicación explícita del peso de la evaluación de cada actividad formativa, se ajustará a lo indicado en la siguiente tabla:

Porcentajes de evaluación

Actividades formativas	Ponderación
Parte Teórica	65%
Parte Práctica	35%
Otros (seminarios, entregas de ejercicios, ...)	Hasta un 5% adicional a la calificación final obtenida

Para aprobar la asignatura es necesario tener una calificación numérica superior o igual a 5 (sobre 10). No obstante, además del requisito anterior, se establece como requisito adicional para superar la asignatura que tanto la calificación correspondiente a la parte teórica como la correspondiente a la parte práctica sean mayores o iguales a 4 (sobre 10).

El sistema de calificaciones se expresará mediante calificación numérica de acuerdo con lo establecido en el art. 5 del R. D 1125/2003, de 5 de septiembre, por el que se establece el sistema europeo de créditos y el sistema de calificaciones en las titulaciones universitarias de carácter oficial y validez en el territorio nacional.

EVALUACIÓN EXTRAORDINARIA

La evaluación extraordinaria constará de dos pruebas de evaluación, una para la parte teórica y otra para la parte práctica, con las características que se indican a continuación:
- Evaluación de teoría: los estudiantes realizarán una única prueba escrita que constará de preguntas de teoría, preguntas tipo test y problemas relacionados con el temario de la asignatura.
- Evaluación de prácticas: los estudiantes realizarán una prueba en el laboratorio que constará de ejercicios o casos prácticos de programación basados en el temario de prácticas de la asignatura.

La ponderación de cada parte en la nota final será del 65% (parte teórica) y el 35% (parte práctica). Para aprobar la asignatura se deben cumplir cada uno de estos tres requisitos:
- La nota de la prueba de teoría es igual o superior al 40% del máximo de dicha prueba.
- La nota de la prueba de prácticas es igual o superior al 40% del máximo de dicha prueba.
- La suma (ponderada) de ambas notas es igual o superior al 50% del máximo posible de dicha suma

EVALUACIÓN ÚNICA FINAL

La evaluación única final constará de dos pruebas de evaluación, una para la parte teórica y otra

para la parte práctica, con las características que se indican a continuación:

- Evaluación de teoría: los estudiantes realizarán una única prueba escrita que constará de preguntas de teoría, preguntas tipo test y problemas relacionados con el temario de la asignatura.
- Evaluación de prácticas: los estudiantes realizarán una prueba en el laboratorio que constará de ejercicios o casos prácticos de programación basados en el temario de prácticas de la asignatura.

La ponderación de cada parte en la nota final será del 65% (parte teórica) y el 35% (parte práctica). Para aprobar la asignatura se deben cumplir cada uno de estos tres requisitos:

- La nota de la prueba de teoría es igual o superior al 40% del máximo de dicha prueba.
- La nota de la prueba de prácticas es igual o superior al 40% del máximo de dicha prueba.
- La suma (ponderada) de ambas notas es igual o superior al 50% del máximo posible de dicha suma

INFORMACIÓN ADICIONAL

Información de interés para estudiantado con discapacidad y/o Necesidades Específicas de Apoyo Educativo (NEAE): Gestión de servicios y apoyos (https://ve.ugr.es/servicios/atencion-social/estudiantes-con-discapacidad).