



Universitat de Lleida

GUÍA DOCENTE
**SISTEMAS CONCURRENTES Y
PARALELOS**

Coordinación: CORES PRADO, FERNANDO

Año académico 2023-24

Información general de la asignatura

Denominación	SISTEMAS CONCURRENTES Y PARALELOS			
Código	102022			
Semestre de impartición	1R Q(SEMESTRE) EVALUACIÓN CONTINUADA			
Carácter	Grado/Máster	Curso	Carácter	Modalidad
	Doble titulación: Grado en Ingeniería Informática y Grado en Administración y Dirección de Empresas	4	OBLIGATORIA	Presencial
	Grado en Ingeniería Informática	3	OBLIGATORIA	Presencial
	Máster Universitario en Ingeniería Informática		COMPLEMENTOS DE FORMACIÓN	Presencial
Número de créditos de la asignatura (ECTS)	6			
Tipo de actividad, créditos y grupos	Tipo de actividad	PRALAB	TEORIA	
	Número de créditos	3	3	
	Número de grupos	2	1	
Coordinación	CORES PRADO, FERNANDO			
Departamento/s	INGENIERÍA INFORMÁTICA Y DISEÑO DIGITAL			
Distribución carga docente entre la clase presencial y el trabajo autónomo del estudiante	6 ECTS = 25x6 = 150 horas de trabajo 40% --> 60 horas presenciales 60% --> 90 horas de trabajo autónomo del estudiante			
Información importante sobre tratamiento de datos	Consulte este enlace para obtener más información.			
Idioma/es de impartición	Preferentemente en Castellano, en Inglés si hay algún estudiante extranjero.			
Distribución de créditos	Fernando Cores 9.0			

Profesor/a (es/as)	Dirección electrónica\nprofesor/a (es/as)	Créditos impartidos por el profesorado	Horario de tutoría/lugar
CORES PRADO, FERNANDO	fernando.cores@udl.cat	9	

Información complementaria de la asignatura

La asignatura es inminentemente práctica, por lo que se dará más peso a las prácticas y la programación. Básicamente trabajaremos con dos lenguajes, C para la programación concurrente con hilos de ejecución y Java para introducir las Apis de alto nivel para la sincronización y los patrones concurrentes. Ninguno de estos lenguajes se explicará desde cero, debido a que tanto C como Java ya se han visto en asignaturas previas de la titulación. Para poder seguir la asignatura es imprescindible que los estudiantes tengan unos buenos fundamentos de programación en C y Java. En la asignatura se da por supuesto que los estudiantes son capaces de diseñar, desarrollar y depurar aplicaciones secuenciales de dificultad media sin muchos problemas.

Objetivos académicos de la asignatura

- Entender la importancia de la programación concurrente en las aplicaciones actuales.
- 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.
- 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.
- Aplicar metodologías de la Ingeniería del Software en el desarrollo de aplicaciones concurrentes y paralelas.

Competencias

Competencias transversales EPS:

- EPS7. Capacidad de trabajar en situaciones de falta de información y/o bajo presión.

Competencias transversales globales:

- GII-CRI11. 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.
- GII-CRI14. Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.
- GII-CRI16. Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.

Contenidos fundamentales de la asignatura

1. Introducción a la concurrencia

1. Definición concurrencia
2. Necesidad y beneficios de la programación concurrente
3. Arquitecturas hardware concurrentes
4. Características sistemas concurrentes
5. Especificación programas Concurrentes
 1. Condiciones de Bernstein
 2. Tabla de concurrencia
 3. Grafos de precedencia
6. Formas de expresar la concurrencia
 1. Cobegin/Coend
 2. Fork/Join
 3. Unix+C
 4. Ejemplos
7. Caso de estudio: Hilos de Ejecución en Java
2. **Diseño aplicaciones concurrentes/paralelas**
 1. Modelo de programación concurrente/paralela
 2. Eficiencia Programas Concurrentes/Paralelos
 1. Alcance del paralelismo
 2. Granularidad
 3. Localidad
 3. Diseño programas concurrentes
 1. Etapas de Diseño
 2. Técnicas de Descomposición de tareas
 3. Patrones de diseño paralelos
 4. Distribución y comunicación de tareas
 4. Casos de estudio: Java Concurrent API
3. **Sincronización de procesos concurrentes**
 1. Introducción Sincronización
 1. Condiciones de Carrera
 2. El problema de la exclusión mutua
 1. Soluciones Software
 2. Soluciones Hardware
 3. Sincronización condicional
 4. Problemas clásicos de sincronización
 5. Caso de estudio: Sincronización en Java
 1. Concepto de Monitores
 2. Sincronización en Java.
4. **Concurrencia y sincronización en C**
 1. Hilos de Ejecución en Linux (Pthreads)
 2. Pthread mutex y variables de condición

Ejes metodológicos de la asignatura

Grupos Grandes: Clases Teoría (3 créditos)

- Lección magistral: clases basadas en apuntes y transparencias en donde se presentarán los conceptos de la asignatura.
- Problemas: Los conceptos de la asignatura se trabajarán mediante una serie de problemas que se resolverán de forma colaborativa y que ayudarán a asimilar los conceptos clave.
- Casos de uso: Se aplicarán las técnicas vistas en clase a ejemplos reales y se analizará su impacto sobre el rendimiento de las aplicaciones.

Grupos Medios: Clases Laboratorio (3 créditos)

- Clases dirigidas y seguimiento personalizado por grupos de prácticas.
- Prácticas de laboratorio: se presentarán las tecnologías y APIs de programación concurrente y se trabajaran de forma práctica mediante tutoriales y ejemplos.

- Problemas: Realización y corrección de problemas relacionados tanto con la parte teórica como práctica de la asignatura.

Trabajo autónomo (no presencial):

- Las prácticas y los problemas se completarán fuera del horario de clase.

Plan de desarrollo de la asignatura

S	Descripción	Actividad Presencial GG (Jueves)	Actividad Presencial GM	Actividad trabajo autónomo
1	Presentación Introducción Concurrencia	T1: Introducción Concurrencia	Martes Lab1 y Lab2 Presentación Asignatura L1: Java Threads	Estudiar bibliografía y programa
2	Introducción Concurrencia	T1: Introducción Concurrencia	L1: Java Threads	T1: Introducción Concurrencia
3	Introducción Concurrencia	Jueves Fiesta (Teoría)	L1: Java Threads Problemas: Programació Concurrent	T1: Introducción Concurrencia
4	Introducción Concurrencia	T1: Introducción Concurrencia	Prácticas 1: Presentación L2: Java Concurrent API	Práctica 1 T1: Introducción Concurrencia
5	Introducción Concurrencia	Jueves Fiesta (Teoría)	L2: Java Concurrent API	Práctica 1 T1: Introducción Concurrencia
6	Diseño aplicaciones concurrentes	T2: Diseño aplicaciones concurrentes	Problemas: Programació Concurrent Problema P1 Eval. Cont.	Práctica 1 Problemas: Concurrencia
7	Diseño aplicaciones concurrentes	T2: Diseño aplicaciones concurrentes	Corrección P1 Eval. Cont. Prácticas 2: Presentación	Práctica 2
8	Diseño aplicaciones concurrentes	T2: Diseño aplicaciones concurrentes	Problemas: Diseño	Práctica 2 Problemas: Diseño
9		1^{er} Parcial		Estudiar
10	Sincronización	T3: Sincronización procesos concurrentes	L3: Sincronización Java	T3: Sincronización
11	Sincronización	T3: Sincronización procesos concurrentes	L3: Sincronización Java Práctica 3: Presentación	Práctica 3 T3: Sincronización
12	Sincronización	T3: Sincronización procesos concurrentes	L4: Concurrencia y Sincronización Pthreads	Práctica 3 T3: Sincronización
13	Sincronización	Jueves Fiesta (Teoría)	L4: Concurrencia y Sincronización Pthreads	Práctica 3 T3: Sincronización
14	Sincronización PThreads	T3: Sincronización procesos concurrentes	Problemas: Sincronización Problema P2 Eval. Cont. Práctica 4: Presentación	Práctica 4 Problemas: Sincronización
15	Sincronización PThreads	Problemas: Sincronización	Corrección P2 Eval. Cont. Problemas: Sincronización	Práctica 4 Problemas: Sincronización
16		2^{on} Parcial		Estudiar
17		2^{on} Parcial		Estudiar

18	TUTORIAS	
19	Recuperación	Estudiar

Sistema de evaluación

EVALUACIÓN

La asignatura se aprueba con una nota final superior o igual 5, obteniendo una nota promedio de 4 en los exámenes y habiendo realizado correctamente las prácticas de laboratorio.

La nota final de la asignatura se obtiene a partir de la suma ponderada de las notas de los dos exámenes, las prácticas y las notas de la evaluación continuada.

Tabla. Actividades de evaluación

Bloque	Acr	Actividad Evaluación	Ponderación	Nota Mínima	En grupo	Obligatoria	Recuperable
Teoría	P1	Examen 1 ^{er} Parcial	20%	4	NO	SI	SI
	P2	Examen 2 ^o Parcial	20%	4	NO	SI	SI
PRA1	PRA1	Práctica 1	10%	NO	SI (<=2)	SI	SI/NO
PRA2	PRA2	Práctica 2	8%	NO	SI (<=2)	SI	SI/NO
PRA3	PRA3	Práctica 3	10%	NO	SI (<=2)	SI	SI/NO
PRA4	PRA4	Práctica 4	12%	NO	SI (<=2)	SI	SI/NO
Eval Continua	EC	Problemas Eval. Cont.	10%	NO	SI (<=2)	NO	NO
	EP	Ejercicios prácticos	10%	NO	NO	NO	NO
	PA	Participación Clase	5%	NO	NO	NO	NO
Existe nota mínima de 4 en la media de las pruebas escritas para poder aprobar la asignatura.							
Nota Final = 0,20*(P1+P2) + 0,10*(PRA1+ PRA2+ PRA3+PRA4) + 0,10*(EC+EP) + 0,05*PA							

La asignatura tiene dos parciales, cada uno de ellos con un peso del 20% en la nota final. Estas pruebas son obligatorias y eliminan materia. Existe una nota mínima (4) para la nota de teoría (promedio de los dos exámenes parciales). En el caso que la ponderación de las diferentes notas sea mayor o igual a 5, pero no se haya alcanzado la nota mínima del bloque de teoría, entonces la asignatura se considerará suspendida con un 4.9.

La realización de las prácticas de laboratorio es obligatoria para aprobar la asignatura. Las prácticas se evaluarán con una nota que representará el 40% de la nota final de la asignatura. De las cuatro prácticas de la asignatura, solo se podrán recuperar dos en la segunda convocatoria. Las prácticas recuperadas tendrán un 20% de penalización en la nota obtenida.

La copia de cualquiera de las prácticas de la asignatura implicará suspender todas las prácticas y con ello la asignatura.

EVALUACIÓN ALTERNATIVA

En el caso de solicitar la evaluación alternativa (por conciliación familiar o laboral), esta consistirá en la realización de un examen final, más la entrega de las 4 prácticas de la asignatura. Los requisitos para estas actividades son los mismos aplicados en la evaluación normal.

Tabla. Actividades de evaluación alternativa

Bloque	Acr	Actividad Evaluación	Ponderación	Nota Mínima	En grupo	Obligatoria	Recuperable
Teoría	EX	Examen Final	50%	4	NO	SI	SI
PRA1	PRA1	Práctica 1	12,5%	NO	NO	SI	SI/NO
PRA2	PRA2	Práctica 2	12,5%	NO	NO	SI	SI/NO
PRA3	PRA3	Práctica 3	12,5%	NO	NO	SI	SI/NO
PRA4	PRA4	Práctica 4	12,5%	NO	NO	SI	SI/NO
<i>Existe nota mínima de 4 en la media de las pruebas escritas para poder tener nota final de la asignatura.</i>							
Nota Final = $0,5 \cdot EX + 0,125 \cdot (PRA1 + PRA2 + PRA3 + PRA4)$							

La asignatura se aprueba con una nota final superior o igual a 5 y obteniendo al menos un 4 en el examen final y habiendo realizado correctamente las prácticas de laboratorio.

La nota final de la asignatura se obtiene a partir de la suma ponderada de la nota del examen final y las notas de las prácticas.

Existe una nota mínima (4) para la nota de teoría. En el caso que la ponderación de las diferentes notas sea mayor o igual a 5, pero no se haya alcanzado la nota mínima del bloque de teoría, entonces la asignatura se considerará suspendida con un 4.9.

Las prácticas se evaluarán con una nota que representará el 50% de la nota final de la asignatura. Las prácticas se entregarán todas de forma conjunta después de realizar el examen. De las cuatro prácticas de la asignatura, solo se podrán recuperar dos en la segunda convocatoria. Las prácticas recuperadas tendrán un 20% de penalización en la nota obtenida.

Bibliografía y recursos de información

Bibliografía Básica:

- José Tomás Palma Méndez, M. C. Garrido Carrera, F. Sanchez Figueroa, A. Quesada Arencibia, "Programación Concurrente", Thomson, 2003.
- Maurice Herlihy, Nir Shavit, "The Art of Multiprocessor Programming", Morgan Kaufmann, 2008
- Douglas Lea, "Concurrent Programming in Java: Design Principles and Patterns", Addison-Wesley Professional, 2000.

Bibliografía Ampliada:

- Gadi Taubenfeld, "Synchronization Algorithms and Concurrent Programming", Pearson / Prentice Hall, 2006
- M. Ben-Ari, "Principles of Concurrent and Distributed Programming", Addison-Wesley, 2nd Edition, 2006
- KayA. Robbins, Steven Robbins, "UNIX Programación Práctica. Guía para la Concurrencia, la Comunicación y los Multihilos", Edt.Prentice-Hall, 1997.

