



Universitat de Lleida

GUÍA DOCENTE  
**COMPUTACIÓN DISTRIBUIDA Y  
APLICACIONES**

Coordinación: CORES PRADO, FERNANDO

Año académico 2023-24

## Información general de la asignatura

<b>Denominación</b>	COMPUTACIÓN DISTRIBUIDA Y APLICACIONES			
<b>Código</b>	102027			
<b>Semestre de impartición</b>	2o Q(SEMESTRE) EVALUACIÓN CONTINUADA			
<b>Carácter</b>	<b>Grado/Máster</b>	<b>Curso</b>	<b>Carácter</b>	<b>Modalidad</b>
	Grado en Ingeniería Informática	3	OBLIGATORIA	Presencial
	Grado en Ingeniería Informática	3	OPTATIVA	Presencial
<b>Número de créditos de la asignatura (ECTS)</b>	6			
<b>Tipo de actividad, créditos y grupos</b>	<b>Tipo de actividad</b>	<b>PRALAB</b>	<b>TEORIA</b>	
	<b>Número de créditos</b>	3	3	
	<b>Número de grupos</b>	1	1	
<b>Coordinación</b>	CORES PRADO, FERNANDO			
<b>Departamento/s</b>	INGENIERÍA INFORMÁTICA Y DISEÑO DIGITAL			
<b>Distribución carga docente entre la clase presencial y el trabajo autónomo del estudiante</b>	6 ECTS = 25x6 = 150 horas de trabajo 40% --> 60 horas presenciales 60% --> 90 horas de trabajo autónomo del estudiante			
<b>Información importante sobre tratamiento de datos</b>	Consulte <a href="#">este enlace</a> para obtener más información.			
<b>Idioma/es de impartición</b>	La asignatura se imparte 40% en Catalán y 60% en Castellano.			
<b>Distribución de créditos</b>	Fernando Cores 6			

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	6	A concertar por correo electrónico

## Información complementaria de la asignatura

La asignatura es inminentemente práctica, por lo cual las prácticas y los trabajos tendrán un peso importante. Básicamente trabajaremos con dos lenguajes, C para la programación paralela y Java para los objetos distribuidos (RMI). Sin embargo, cuando hablamos de aplicaciones distribuidas, tan importante como la propia programación es el diseño de la aplicación, por lo cual también aplicaremos de forma intensiva los conceptos de ingeniería del software.

Para poder seguir la asignatura es imprescindible que los estudiantes tengan unos buenos fundamentos de programación en C y Java. Es mucho más difícil, aprender a desarrollar programas distribuidos y paralelos si no se domina previamente la programación secuencial. 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

- Introducir los conceptos básicos sobre la computación distribuida y la organización de los sistemas distribuidos.
- Proporcionar una visión general de las principales arquitecturas de computación distribuidas y su impacto sobre las tecnologías de la información.
- Asimilar los principios fundamentales y los distintos tipos de modelos subyacentes a los aspectos de funcionamiento de la computación distribuida.
- Conocer los principales paradigmas de computación distribuida y entender sus puntos fuertes, sus desventajas y principales ámbitos de aplicación.
- Comprender los retos tecnológicos que representan la utilización, el diseño y la implementación de los sistemas distribuidos.
- Proporcionar una visión general de los sistemas distribuidos, analizando diferentes casos de estudio y aplicándolos para solventar problemas reales en diferentes ámbitos de la computación distribuida
- Desarrollar las habilidades de diseño y análisis de los sistemas distribuidos que ayuden a comprender, evaluar la calidad las soluciones propuestas
- Incentivar la adopción del modelo distribuido para la compartición de recursos a gran escala de forma transparente e independiente de su ubicación física.

## Competencias

### Competencias estratégicas de la UdL:

- CT2. Adquirir un dominio significativo de una lengua extranjera, especialmente del inglés
- CT3. Adquirir capacitación en el uso de las nuevas tecnologías y de las tecnologías de la información y la comunicación

### Competencias transversales EPS:

- EPS11. Capacidad de comprender las necesidades del usuario expresadas en un lenguaje no técnico.

## Competencias específicas:

- GII-T12. Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados.
- GII-T5. Capacidad para seleccionar, desplegar, integrar y gestionar sistemas de información que satisfagan las necesidades de la organización, con los criterios de coste y calidad identificados.
- GII-T16. Capacidad de concebir sistemas, aplicaciones y servicios basados ??en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil.

## Contenidos fundamentales de la asignatura

### 1. Introducción a la computación distribuida

1. Definiciones y conceptos
2. Retos Computación Distribuida
3. Tipo Sistemas Distribuidos
  1. Sistemas de Computación Distribuida
  2. Sistemas de Información Distribuidos
  3. Sistemas Empotrado Distribuidos
4. Arquitecturas Sistemas Distribuidos
5. Paradigmas Computación Distribuida
  1. Paso de mensajes
  2. Cliente-servidor
  3. Peer-to-Peer
  4. Sistemas de Mensajes
  5. RPC (Remote Procedure Call)
  6. Objetos distribuidos
  7. Agentes Móviles

### 2. Objetos distribuidos

1. Introducción
2. Paradigma de objetos distribuidos
3. RMI: Invocación remota a métodos.
4. RMI Avanzado
  1. Callbacks
  2. Gestión Seguridad en RMI

### 3. Hardware y Software de Computación Paralela

1. Arquitecturas paralelas
2. Redes de interconexión
3. Clusters
4. Colas de trabajos

### 4. Diseño algoritmos paralelos

1. Introducción programación paralela
2. Rendimiento Aplicaciones paralelas
3. Diseño aplicaciones paralelas
4. Casos de Estudio

### 5. Programación Paralela

1. El modelo de paso de mensajes
2. La interfaz de paso de mensajes
3. Rutinas básicas
4. Comunicación Punto a punto
5. Comunicación y operaciones colectivas
6. Tipos de datos derivados
7. Extensiones MPI-2
  1. Entrada/Salida Paralela
  2. Gestión dinámica de procesos

## Ejes metodológicos de la asignatura

### 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.

### Clases Laboratorio (3 créditos)

- Clases dirigidas y seguimiento personalizado por grupos de prácticas.
- Prácticas de laboratorio: se presentarán las tecnologías y APIs de programación distribuida 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.
- Espacio Foro. En este espacio el alumno puede plantear dudas relativas a los contenidos vistos en las sesiones de Teoría y Laboratorio, así como plantear todo tipo de dudas sobre el proyecto. Se anima a todos los estudiantes a participar en la solución de las dudas de sus compañeros. El profesorado participa para hacer aclaraciones o resolver aquellas dudas que no tienen ninguna respuesta por parte de los estudiantes.

## Plan de desarrollo de la asignatura

Sem	Descripción	Actividad Presencial GG1 (Martes)	Actividad Presencial GG2 (Miércoles)	Actividad trabajo autónomo
1	Presentación Introducción	Presentación Asignatura	T1: Introducción a la computación distribuida	<b>Estudiar bibliografía y programa</b>
2	Introducción	T1: Introducción a la computación distribuida Problemas: Retos Computación distribuida	T1: Introducción a la computación distribuida	Problemas: Retos
3	Introducción	T1: Introducción a la computación distribuida	T1: Introducción a la computación distribuida	Problemas: Retos
4	Introducción Objetos distribuidos	Problemas: Retos y Paradigmas sistemas distribuidos	T2: Objetos distribuidos e invocación remota	Problemas: Paradigmas RMI
5	Objetos distribuidos	T2: Objetos distribuidos e invocación remota Práctica1: Enunciado RMI	T2: Objetos distribuidos e invocación remota <b>Tutorial: RMI</b>	Tutorial: RMI Práctica1
6	Objetos distribuidos Computación Paralela	T2: Objetos distribuidos e invocación remota <b>Tutorial: RMI</b>	T3: Hardware y Software de Computación Paralela	Tutorial: RMI Práctica1
7	Computación Paralela	T3: Hardware y Software de Computación Paralela <b>Tutorial: SGE</b>	T3: Hardware y Software de Computación Paralela <b>Tutorial: SGE</b>	Tutorial: Colas de trabajos Práctica1
8	Computación Paralela	T3: Hardware y Software de Computación Paralela <b>Tutorial: SGE</b>	T3: Hardware y Software de Computación Paralela <b>Tutorial: SGE</b>	Tutorial: Colas de trabajos Práctica1

9		<b>1<sup>er</sup> Parcial</b>		Estudiar
10	Diseño Paralelo	T4: Diseño algoritmos paralelos	T4: Diseño algoritmos paralelos Práctica2: Enunciado	Práctica2
11	Diseño Paralelo	T4: Diseño algoritmos paralelos	T4: Diseño algoritmos paralelos	Práctica2
12	Diseño Paralelo	Problemas: Diseño programas paralelos	<b>Miércoles Fiesta</b>	Práctica2 Problemas: Diseño
13	Programación Paralela	T5: Programación Paralela <b>Tutorial: MPI</b>	T5: Programación Paralela <b>Tutorial: MPI</b> Práctica2: Entrega Diseño	Tutorial: MPI Práctica2
14	Programación Paralela	T5: Programación Paralela <b>Tutorial: MPI</b>	T5: Programación Paralela <b>Tutorial: MPI</b>	Tutorial: MPI Práctica2
15	Programación Paralela	T5: Programación Paralela <b>Tutorial: MPI</b>	Problemas: Implementación programas paralelos	Tutorial: MPI Práctica2
16		<b>2on Parcial</b>		Estudiar
17		<b>2on Parcial</b>		Estudiar
18		TUTORIAS		
19		<b>Recuperación</b>		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 <sup>er</sup> Parcial	20%	4	NO	SI	SI
	P2	Examen 2 <sup>o</sup> Parcial	20%	4	NO	SI	SI
PRA1	PRA1	Práctica 1	16%	NO	SI (<=2)	SI	SI/NO
PRA2	PRA2	Práctica 2	16%	NO	SI (<=2)	SI	SI/NO
Eval. Continua	TR	Trabajos	7%	NO	SI (<=2)	NO	NO
	TU	Tutoriales	21%	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.							
<b>Nota Final</b> = 0,20*(P1+P2) + 0,16*(PRA1+ PRA2) + 0,07*TR + 0,21*TU + 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 32% de la nota final de la asignatura. De las dos prácticas de la asignatura, solo se podrán recuperar una 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 la asignatura.

Respecto a las actividades de evaluación continua, está consta de un trabajo en grupo (7%), más 3 tutoriales (ejercicios básicos de programación) con un peso del 7% cada uno. Por último, también se tendrá en cuenta la participación en clase con un 5% extra sobre la nota final.

## 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 2 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	25%	NO	NO	SI	SI/NO
PRA2	PRA2	Práctica 2	25%	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>							
<b>Nota Final</b> = $0,5 * EX + 0,25 * (PRA1 + PRA2 + PRA3 + PRA4)$							

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 una 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 la asignatura.

## Bibliografía y recursos de información

### Bibliografía Básica:

- Coulouris G, Dollimore J., Kindberg T.: "Sistemas distribuidos: Conceptos y diseño"; Addison-Wesley, 2001.
- M.L. Liu, "Computación distribuida". Edt. Addison Wesley, 2004
- [Peter Pacheco](#), "An Introduction to Parallel Programming", Morgan Kaufmann, 2011.

### Bibliografía Ampliada:

- M. Ben-Ari, "Principles of Concurrent and Distributed Programming", Addison-Wesley, 2nd Edition, 2006
- [Rajkumar Buyya](#): "High Performance Cluster Computing: Architectures and Systems", Edt. Pearson Education; 1st edition 1999
- [Rajkumar Buyya](#): "High Performance Cluster Computing: Programming and Applications", Volume 2, Edt. Prentice Hall, 1st edition 1999.