



Guía docente de la asignatura

## Programación Paralela (Especialidad Ingeniería del Software) (29611BC)

Fecha de aprobación: 20/06/2022

<b>Grado</b>	Grado en Ingeniería Informática	<b>Rama</b>	Ingeniería y Arquitectura				
<b>Módulo</b>	Complementos de Ingeniería del Software	<b>Materia</b>	Complementos de Programación Paralela y Sistemas Operativos				
<b>Curso</b>	4 <sup>o</sup>	<b>Semestre</b>	2 <sup>o</sup>	<b>Créditos</b>	6	<b>Tipo</b>	Optativa

### PRERREQUISITOS Y/O RECOMENDACIONES

Se recomienda tener cursadas las asignaturas básicas y obligatorias relativas a programación de ordenadores y sistemas concurrentes.

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

- Sistemas paralelos y modelos de programación paralela.
- Metodología de diseño de algoritmos paralelos.
- Análisis de algoritmos paralelos.
- Esquemas algorítmicos paralelos.
- Lenguajes y bibliotecas de programación paralela.

### 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.
- CG09 - Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.

### RESULTADOS DE APRENDIZAJE (Objetivos)

#### Objetivos formativos particulares



- Conocer la importancia y aplicaciones de la programación paralela en la actualidad.
- Adquirir los conceptos básicos de la computación paralela, incluyendo el conocimiento de los modelos de computación paralela más destacados.
- Conocer los modelos de programación paralela más extendidos y las principales herramientas software que los sustentan.
- Ser capaz de evaluar el rendimiento de un algoritmo paralelo, en función de su interés en aplicaciones concretas, utilizando técnicas de modelado propias del ámbito de la computación paralela.
- Conocer las principales técnicas de descomposición y asignación de tareas, y saber usarlas para diseñar algoritmos paralelos.
- Identificar los factores de diseño que más influyen en el rendimiento de una aplicación paralela.
- Conocer y saber usar los esquemas algorítmicos más frecuentes en el desarrollo de software paralelo.
- Adquirir las habilidades propias de la programación paralela en un entorno de memoria compartida y en un entorno de memoria distribuida.
- Conocer y adquirir experiencia en el uso de los lenguajes y bibliotecas más usados para la implementación de programas paralelos.
- Ser capaz de escribir, compilar, ejecutar y evaluar experimentalmente programas paralelos en un ordenador paralelo.

### Objetivos formativos de carácter general

- Ser capaz de desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la Ingeniería del Software.
- Ser capaz de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.

## PROGRAMA DE CONTENIDOS TEÓRICOS Y PRÁCTICOS

### TEÓRICO

#### Tema 1. Introducción a los sistemas paralelos y a la programación paralela

- Motivación y aspectos de la programación paralela.
- Introducción a los sistemas de cómputo paralelo y a los modelos de programación paralela.
- Estudio del rendimiento de un sistema paralelo.

#### Tema 2. Lenguajes e Interfaces para programación paralela

- Lenguajes e interfaces para la programación de Procesadores Gráficos (GPUs).
- Programación paralela y distribuida usando la interfaz de paso de mensajes, MPI.

#### Tema 3. Diseño de algoritmos paralelos

- Técnicas de descomposición en tareas.
- Técnicas de asignación de tareas y equilibrado de carga



## PRÁCTICO

### Seminarios/Talleres

- Seminario 1: Introducción a la programación de GPUs usando CUDA C.
- Seminario 2: Introducción al desarrollo de programas paralelos usando MPI.

### Prácticas de Laboratorio

- Práctica 1. Implementación eficiente en GPU de algoritmos de cálculo intensivo usando CUDA C.
- Práctica 2. Implementación en memoria distribuida (usando MPI) de un algoritmo paralelo de datos.
- Práctica 3. Implementación usando MPI de un algoritmo de equilibrado dinámico de la carga.

## BIBLIOGRAFÍA

### BIBLIOGRAFÍA FUNDAMENTAL

- G. Barlas. Multicore and GPU Programming. An Integrated Approach. Morgan Kaufmann. 2015.
- John Cheng, Max Grossman, Ty McKercher. Professional CUDA C Programming. John Wiley & Sons, 2014.
- F. Almeida, D. Gimenez, J. M. Mantas, A.M. Vidal . Introduccion a la Programacion Paralela. Paraninfo Cengage Learning, 2008.
- David B. Kirk, Wen-meí W. Hwu. Programming Massively Parallel Processors, Second Edition: A Hands-on Approach. Morgan Kaufmann. 2012.
- V. Kumar , A. Grama, A. Gupta, G. Karypis. Introduction to Parallel Computing. Benjamin/Cummings Publishing Company, 2003.

### BIBLIOGRAFÍA COMPLEMENTARIA

- Nicholas Wilt . CUDA Handbook: A Comprehensive Guide to GPU Programming. Addison-Wesley, 2013.
- Storti Duane, Yurtoglu Mete. CUDA for Engineers: An Introduction to High-Performance Parallel Computing, Prentice Hall. 2015.
- Thomas Rauber, Gudula Rünger. Parallel Programming: for Multicore and Cluster Systems. Springer Verlag. 2010.

## ENLACES RECOMENDADOS

- [Página web de documentación de la asignatura](#)
- Plataforma [PRADO](#)

## 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).

## 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 Normativa.

**Para la parte teórica, se valorarán las siguientes tareas:**

- Resolución de ejercicios/problemas, tanto mediante entregas de ejercicios resueltos como mediante su resolución en clase, así como la participación activa del estudiante en clase. Se exigirán al menos tres aportaciones del estudiante en este tipo de tareas. Cada aportación individual podrá suponer una ponderación máxima del 20% en la evaluación de la parte teórica.
- Entrega y presentación de trabajos por parte del estudiante. Esta tarea podrá suponer una ponderación de hasta el 30% en la evaluación de la parte teórica.

**Para la parte práctica se realizarán prácticas de programación y desarrollo de miniproyectos (individuales o en grupo), y se valorarán las entregas de los informes realizados por los estudiantes y las entrevistas personales con los mismos. La ponderación de cada una de las prácticas en la evaluación será la siguiente:**

Porcentaje de evaluación

Práctica	1	2	3
Ponderación	35%	25%	40%

En su caso, los seminarios se evaluarán teniendo en cuenta los problemas propuestos que hayan sido resueltos y entregados por los alumnos, su participación activa en clase y la presentación de los trabajos desarrollados.

La calificación global corresponderá 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 su caso, una parte relacionada con el trabajo autónomo de los alumnos y los seminarios impartidos. 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:

Porcentaje de evaluación

Actividad Formativa	Ponderación
Parte teórica	40 %



Parte práctica	60 %
Otros (seminarios, ...)	Hasta un 10%

En cualquiera de los modalidades de evaluación (ordinaria o única final), 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 3 (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 de los estudiantes en la convocatoria extraordinaria se regirá por los mismos criterios y constará de las mismas pruebas que las indicadas en este documento para la evaluación única final (ver más abajo).

### EVALUACIÓN ÚNICA FINAL

Para los estudiantes que se acojan a la evaluación única final o que se presenten a la convocatoria extraordinaria, la evaluación se llevará a cabo mediante dos sesiones:

- Una sesión de evaluación para la parte teórica basada en resolución de ejercicios y problemas, que permitirá al estudiante obtener un máximo de 4 puntos sobre 10 en la calificación numérica.
- Una sesión de evaluación para la parte práctica, consistente en la entrega y defensa, por parte del estudiante, de los informes realizados en sus prácticas, que permitirá al estudiante obtener un máximo de 6 puntos sobre 10 en la calificación numérica.

La calificación global corresponderá a la suma de las puntuaciones obtenidas en ambas sesiones de evaluación, siempre que se haya obtenido al menos el 30% del máximo de puntos en cada parte.

