

Titulación	Tipo	Curso
4313136 Modelización para la Ciencia y la Ingeniería / Modelling for Science and Engineering	OB	0

## Contacto

Nombre: Anna Barbara Sikora

Correo electrónico: [anna.sikora@uab.cat](mailto:anna.sikora@uab.cat)

## Idiomas de los grupos

Puede consultar esta información al [final](#) del documento.

## Prerrequisitos

No hay prerrequisitos

## Objetivos y contextualización

El objetivo del módulo es:

- Desarrollar programas de c
- Identificar las dificultades relacionadas con la programación paralela.
- Aplicar una metodología adecuada para el desarrollo de aplicaciones paralelas.
- Comprender las diferencias de los enfoques de programación en paralelo: memoria compartida, paso de mensajes
- Determinar el enfoque de programación paralela más conveniente para desarrollar una aplicación particular
- Desarrollar aplicaciones paralelas.
- Evaluar el rendimiento de la aplicación paralela y recopile las medidas necesarias para ajustar la aplicación a fin de mejorar su rendimiento

Al final de este módulo, los estudiantes deben tener conocimientos, métodos y habilidades técnicas suficientes para desarrollar aplicaciones paralelas utilizando un modelo de programación adecuado y evaluar el rendimiento de la aplicación.

## Competencias

- Analizar y evaluar arquitecturas de computadores paralelos y distribuidos, así como desarrollar y optimizar software avanzado para las mismas
- Analizar, sintetizar, organizar y planificar proyectos de su campo de estudio.
- Aplicar la metodología de investigación, técnicas y recursos específicos para investigar en un determinado ámbito de especialización.
- Asegurar, gestionar, auditar y certificar la calidad de los desarrollos, procesos, sistemas y productos informáticos avanzados.
- Innovar en la búsqueda de nuevos espacios / ámbitos en su campo de trabajo.

- Participar en proyectos de investigación y equipos de trabajo en el ámbito de la ingeniería de la información y el cómputo de altas prestaciones.
- Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación.
- Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

## Resultados de aprendizaje

1. Analizar, sintetizar, organizar y planificar proyectos de su campo de estudio.
2. Aplicar la metodología de investigación, técnicas y recursos específicos para investigar en un determinado ámbito de especialización.
3. Desarrollar la solución paralela a un problema computacional eligiendo las herramientas más adecuadas
4. Diseñar la solución paralela a un problema computacional tomando en cuenta las características del hardware disponible
5. Identificar las fuentes de paralelismo en un problema computacional
6. Innovar en la búsqueda de nuevos espacios / ámbitos en su campo de trabajo.
7. Interpretar la información dada por las herramientas de análisis de rendimiento y transformarla en acciones que mejoren la aplicación paralela
8. Planificar y desarrollar proyectos de investigación con contenidos relacionados con la programación paralela
9. Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación.
10. Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.
11. Utilizar las herramientas adecuadas para analizar el rendimiento de una aplicación.

## Contenido

1. Introducción al curso
2. Programación en C
3. Introducción a los sistemas de altas prestaciones, sistemas paralelos, multiprocesadores y multicomputadores. Ejecución de aplicaciones paralelas en sistemas de altas prestaciones.
4. Programación OpenMP - Aplicaciones basadas en memoria compartida. Estándar OpenMP (Open multiprocessing).
5. Programación MPI - Aplicaciones basadas en paso de mensajes. Estándar MPI (Message Passing Interface).
6. Análisis de rendimiento - Análisis de prestaciones de sistemas paralelos. Evaluación de prestaciones de sistemas paralelos. Ejemplos de herramientas de evaluación de prestaciones.
7. Programación GPU

## Actividades formativas y Metodología

Título	Horas	ECTS	Resultados de aprendizaje
Tipo: Dirigidas			
Asistencia a clases de teoría y ejercicios prácticos	14	0,56	2, 3, 4, 5, 8, 9
Prácticas de laboratorio	24	0,96	3, 4, 7, 8

Tipo: Autónomas

Diseño desarrollo de ejercicios prácticos	52	2,08	1, 3, 4, 5, 7, 8, 10, 9, 11
Estudio	36	1,44	1, 2, 3, 4, 5, 6, 7, 8, 10, 9, 11

La asignatura se desarrollará en clases teóricas y ejercicios prácticos.

Es recomendable que el alumnado asista a todas las clases de la asignatura con un ordenador portátil con la batería bien cargada.

El planteamiento del trabajo está orientado a promover un aprendizaje activo y desarrollar las competencias de capacidad de organización y planificación, comunicación oral y escrita, trabajo en equipo y razonamiento crítico. La calidad de los ejercicios realizados, de su presentación y de su funcionamiento se valorará especialmente.

La gestión de la docencia de la asignatura se hará a través del Campus Virtual (<https://cv.uab.cat/>), que servirá para poder ver los materiales, gestionar los grupos de prácticas, hacer las entregas correspondientes, ver las notas, comunicarse con los/las profesores/as, etc.

Nota: se reservarán 15 minutos de una clase dentro del calendario establecido por el centro o por la titulación para que el alumnado rellene las encuestas de evaluación de la actuación del profesorado y de evaluación de la asignatura o módulo.

## Evaluación

### Actividades de evaluación continuada

Título	Peso	Horas	ECTS	Resultados de aprendizaje
Ejercicios prácticos con GPUs	20	6	0,24	3, 4, 5
Ejercicios prácticos en C	15	6	0,24	10, 9
Ejercicios prácticos en MPI	20	6	0,24	1, 2, 3, 4, 5, 6, 7, 8, 10, 9, 11
Ejercicios prácticos en OpenMP	15	4	0,16	1, 2, 4, 5, 6, 8
Examen Final	30	2	0,08	3, 4, 5, 10, 9

El objetivo del proceso de evaluación es verificar que el alumno ha adquirido los conocimientos y habilidades de:  
Se evaluarán cinco tipos de actividades de forma independiente donde la

1. Programación en C (15%)
2. Programación en OpenMP (15%)
3. Programación en MPI (20%)
4. Programación con GPUs (20%)
5. Examen final individual (30%)

Las evaluaciones de la programación serán realizadas en grupos de 2 personas. Examen final será individual y €

Nota sobre plagios:

Sin perjuicio de otras medidas disciplinarias que se estimen oportunas, y de acuerdo con la normativa académica vigente, las irregularidades cometidas por un estudiante que puedan conducir a una variación de la calificación en una actividad evaluable se calificarán con un cero (0). Las actividades de evaluación calificadas de esta forma y por este procedimiento no serán recuperables. Si es necesario superar cualquiera de estas actividades de evaluación para aprobar la asignatura, esta asignatura quedará suspendida directamente, sin oportunidad de recuperarla en el mismo curso. Estas irregularidades incluyen, entre otros:

- la copia total o parcial de una práctica, informe, o cualquier otra actividad de evaluación;
- dejar copiar;
- presentar un trabajo de grupo no hecho íntegramente por los y las miembros del grupo (aplicado a todos los y las miembros, no sólo a los que no han trabajado);
- presentar como propios materiales elaborados por un tercero, aunque sean traducciones o adaptaciones, y en general trabajos con elementos no originales y exclusivos del estudiante;
- el uso no autorizado de la IA (p. ej., Copilot, ChatGPT o equivalentes);
- tener dispositivos de comunicación (como teléfonos móviles, smart watches, bolígrafos con cámara, etc.) accesibles durante las pruebas de evaluación teórico-prácticas individuales (exámenes);
- hablar con compañeros o compañeras durante las pruebas de evaluación teórico-prácticas individuales (exámenes);
- copiar o intentar copiar de otros alumnos durante las pruebas de evaluación teórico-prácticas (exámenes);
- usar o intentar usar escritos relacionados con la materia durante la realización de las pruebas de evaluación teórico-prácticas (exámenes), cuando éstos no hayan sido explícitamente permitidos.

## Bibliografía

Parallel Programming : Techniques and Applications using Networked Workstations and Parallel Computers. Barry Wilkinson. Prentice Hall, 1999.

Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering. Ian Foster . Addison Wesley, 1995.

Introduction to Parallel Computing. A. Grama et al. Addison Wesley, Second Edition, 2003.

Parallel Program Development For Cluster Computing: Methodology, Tools and Integrated Environments. Edited by J. C. Cunha, P. Kacsuk, S. C. Winter. Nova Science Publishers, Inc., 2001.

High Performance Cluster Computing (Vols. 1 y 2), Rajkumar Buyya ed., Prentice Hall, 1999.

Parallel Programming with MPI, Peter Pacheco, Morgan Kaufman, 1996

Using MPI-2, William Gropp, Ewing Lusk and Rajeev Thakur, MIT Press, 1999.

Programming Massively Parallel Processors: A Hands-on Approach. David Kirk and Wen-mei Hwu. ISBN: 978-0-12-381472-2. Published by Elsevier Inc. 2010.

## Software

C

OpenMP

MPI

OpenACC

gestor de colas (SLURM)

herramientas para evaluación de rendimiento (perf, TAU, nvcc)

conexión remota con laboratorio (e.g. MobaTex, Eclipse, Visual Studio, etc.)

## Lista de idiomas

Nombre	Grupo	Idioma	Semestre	Turno
(PLABm) Prácticas de laboratorio (máster)	1	Inglés	primer cuatrimestre	tarde
(PLABm) Prácticas de laboratorio (máster)	2	Inglés	primer cuatrimestre	tarde
(TEm) Teoría (máster)	1	Inglés	primer cuatrimestre	tarde