

## Programación Paralela

Código: 104356  
Créditos ECTS: 6

Titulación	Tipo	Curso	Semestre
2503758 Ingeniería de Datos	OB	2	2

### Contacto

Nombre: Eduardo Cesar Galobardes

Correo electrónico: eduardo.cesar@uab.cat

### Uso de idiomas

Lengua vehicular mayoritaria: catalán (cat)

Algún grupo íntegramente en inglés: No

Algún grupo íntegramente en catalán: No

Algún grupo íntegramente en español: No

### Equipo docente

Eduardo Cesar Galobardes

Cristina Peralta Quesada

Christian Guzman Ruiz

Anna Bàrbara Sikora

### Prerequisitos

Aunque no hay prerequisites formalmente establecidos y es responsabilidad de la propia asignatura proporcionar a los alumnos un medio para adquirir los conocimientos descritos en el apartado de contenidos de la asignatura, es recomendable: un buen conocimiento de programación, del funcionamiento de un computador y del sistema operativo a nivel de usuario programador (Fundamentos de Informática, Programación Avanzada, Ingeniería de Rendimiento).

### Objetivos y contextualización

El objetivo de esta asignatura es conocer los sistemas de cómputo de altas prestaciones y paralelos, sistemas multiprocesador y multicomputadores, paradigmas de programación paralela, aprender a desarrollar aplicaciones con memoria compartida, introducir el concepto de paso de mensajes y aplicaciones en memoria distribuida y analizar las prestaciones de ejecución de estas aplicaciones.

Los conceptos teóricos sobre paradigmas de programación, memoria compartida y paso de mensajes, se refuerza con las sesiones de problemas y prácticas en las que los alumnos aprenden a programar utilizando lenguajes de programación paralela.

Todos los componentes descritos en esta asignatura deben permitir al alumno comprender el funcionamiento de los sistemas de altas prestaciones y paralelos y, hasta cierto punto, ser capaz de realizar un diseño sencillo de una aplicación paralela y evaluar sus prestaciones.

### Competencias

- Buscar, seleccionar y gestionar de manera responsable la información y el conocimiento.

- Desarrollar un pensamiento y un razonamiento crítico y saber comunicarlo de manera efectiva, tanto en las lenguas propias como en inglés.
- Diseñar soluciones algorítmicas eficientes para problemas computacionales, implementarlas en forma de desarrollo de software robustos, estructurados y fáciles de mantener, y verificar su validez.
- Evaluar de manera crítica el trabajo realizado.
- Planificar y gestionar el tiempo y los recursos disponibles.
- Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.

## Resultados de aprendizaje

1. Analizar los modelos, paradigmas y lenguajes de programación paralela disponibles para determinar el que mejor se adecua a las necesidades de una aplicación.
2. Buscar, seleccionar y gestionar de manera responsable la información y el conocimiento.
3. Desarrollar aplicaciones paralelas basadas en los paradigmas existentes más habituales.
4. Desarrollar un pensamiento y un razonamiento crítico y saber comunicarlo de manera efectiva, tanto en las lenguas propias como en inglés.
5. Evaluar de manera crítica el trabajo realizado.
6. Planificar y gestionar el tiempo y los recursos disponibles.
7. Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.
8. Usar las herramientas y las metodologías más adecuadas para evaluar la funcionalidad y el rendimiento de las aplicaciones paralelas / distribuidas desarrolladas.

## Contenido

Tema 1: Introducción a sistemas de altas prestaciones.

Programación C avanzada e integración de código C en Python. Concurrencia: concepto, condición de carrera, región crítica y mecanismos d'exclusión mutua.

Introducción a los sistemas de altas prestaciones, sistemas paralelos, multiprocesadores y multicomputadores. Ejecución de aplicaciones paralelas en sistemas de altas prestaciones.

Tema 2: Introducción a los sistemas y algoritmos paralelos

SIMD (Single Instruction, Multiple Data), MIMD (Multiple Instruction, Multiple Data). Modelos de aplicaciones paralelas.

Definición de algoritmos paralelos. Ejemplos de algoritmos paralelos. Desarrollo de algoritmos paralelos.

Tema 3: Programación paralela

Paradigmas de programación paralela. Aplicaciones basadas en paso de mensajes. Estándar MPI (Message Passing Interface). Aplicaciones basadas en memoria compartida. Estándares OpenMP (Open multiprocessing), OpenACC (Open Accelerators) y Cuda. Paralelismo en Python. Desarrollo de aplicaciones paralelas utilizando MPI, OpenMP, OpenACC, Cuda y Python.

Tema 4: Análisis de prestaciones

Análisis de prestaciones de sistemas paralelos. Evaluación de prestaciones de sistemas paralelos. Ejemplos de herramientas de evaluación de prestaciones.

## Metodología

En el desarrollo de la asignatura se podrán diferenciar cuatro tipos de actividades docentes:

- Clases teóricas: exposición en la pizarra de la parte teórica de cada tema del programa. La estructura típica de una clase magistral de este tipo será la siguiente: en primer lugar se hará una introducción donde se presentarán brevemente los objetivos de la exposición y los contenidos a tratar. A continuación se desglosarán los contenidos objeto de estudio, incluyendo exposiciones narrativas, desarrollos formales que proporcionen los fundamentos teóricos, e intercalando ejemplos que ilustren la aplicación de los contenidos expuestos. Finalmente, el/la profesor/a expondrá las conclusiones de los contenidos. Durante todo el curso habrá evaluación continuada de grupos de temas.
- Clases prácticas: Todos los temas irán acompañados de una lista de problemas que el alumno debe intentar resolver. En este sentido, y a medida que el alumno vaya progresando y profundizando en sus conocimientos, estos problemas serán poco a poco más complejos. Los seminarios serán el foro natural en el que se podrá discutir en común el desarrollo del trabajo práctico, aportando los conocimientos que le faltan al estudiante para llevarlo adelante. La misión de las clases prácticas es hacer de puente entre las clases teóricas y las clases de laboratorio, que promoverá la capacidad de análisis y síntesis, el razonamiento crítico, y que entrenará al estudiante en la resolución de problemas. Aquellos que el/la profesor/a considere de mayor interés o en los que los alumnos/as encuentren mayor dificultad serán corregidos en la pizarra. Antes del comienzo de cada tema de problemas el/la profesor/a podrá proponer una lista de ejercicios que los alumnos/as deberán resolver.
- Clases en el laboratorio: La parte práctica de los temas teóricos quedará completada con sesiones en el laboratorio, donde el/la alumno/a desarrollará una serie de programas y deberá intentar resolver un problema concreto que recibirá al comenzar el temario. Algunos de estos ejercicios se deberán entregar a la clase en las fechas especificadas. Las prácticas se desarrollarán en grupos de dos o tres alumnos. Las clases incluyen como mínimo 6 sesiones en el laboratorio, de 2 horas de duración, donde el/la alumno/a realizará el desarrollo de los ejercicios.
- Realización de trabajos: Durante el curso, los estudiantes en grupos de dos o tres alumnos realizará un trabajo. Cada grupo redactará o mejorará un artículo de la Viquipèdia de un tema relacionado con la asignatura. Con este ejercicio se busca que el estudiante sea capaz de presentar por escrito temas relacionados con el cómputo de altas prestaciones para una audiencia genérica.

Este 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.

Competencias transversales:

Como se ha mencionado antes, la asignatura incluye como actividad docente la realización de un trabajo que consiste en el desarrollo de un artículo a la Vikepèdia y está orientado a comunicar eficientemente por escrito, conocimientos, resultados y habilidades, tanto en entornos profesionales como ante públicos no expertos, haciendo un uso eficiente de las TIC en la comunicación y la transmisión de ideas (competencias GT02, GT03 y GT07).

Los estudiantes disponen de material para el desarrollo de artículos en la Vikepèdia (que incluye la página del Viquiprojecte: Còmput d'altas prestacions). Durante el semestre se realizan un mínimo de dos reuniones con los grupos para discutir y evaluar la evolución del trabajo.

La gestión de la docencia de la asignatura se hará a través del Campus Virtual (<https://cv2008.uab.cat/>), que servirá para poder ver los materiales, gestionar los grupos de prácticas, hacer las entregas correspondientes, ver las notas, comunicarse con los profesores, 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.

## Actividades

Título	Horas	ECTS	Resultados de aprendizaje
--------	-------	------	---------------------------

Tipo: Dirigidas

Problemas (Resolución de ejercicios)	9	0,36	1, 5, 2, 3, 4, 6, 8
Práctica (Resolución de las prácticas de laboratorio)	12	0,48	1, 5, 2, 3, 4, 6, 8
Teoría	20	0,8	1, 2, 3, 4, 8

Tipo: Autónomas

Estudio autónomo	30	1,2	1, 2, 3, 6, 7, 8
Investigación	15	0,6	5, 2, 4, 6, 7
Preparación de problemas	20	0,8	1, 2, 3, 6, 7, 8
Preparación de prácticas	30	1,2	1, 5, 2, 3, 6, 8

## Evaluación

El objetivo del proceso de evaluación es verificar que el alumno ha adquirido los conocimientos y habilidades definidos en los objetivos de la asignatura, así como las competencias.

Se evaluarán cuatro tipos de actividades de manera independiente donde la suma ponderada de ellas dará la nota final. Estas actividades son:

1. Teoría (T)
2. Resolución de las prácticas de laboratorio (PL)
3. Redacción de un artículo en la Wikipedia (VA)
4. Realización de ejercicios prácticos individuales (PA).

La parte de Teoría (T) se evaluará con dos controles parciales individuales durante todo el curso. La nota final de Teoría saldrá de la suma ponderada de los dos controles ( $0.5 * \text{Control 1} + 0.5 * \text{Control 2}$ ). Habrá una segunda oportunidad para recuperar esta parte el día que tengamos asignado en la semana de exámenes de junio. Se podrán recuperar las partes que no hayan sido superadas en los controles parciales de teoría. La nota mínima para aprobar esta parte es  $\geq 5$ .

La parte de Resolución de prácticas de laboratorio (PL) se evaluará de manera grupal. Tiene cuatro entregas. La nota final saldrá de la suma ponderada de las cuatro entregas ( $X0 * \text{Entrega 1} + X1 * \text{Entrega 2} + X2 * \text{Entrega 3} + X3 * \text{Entrega 4}$ ,  $X0 + X1 + X2 + X3 = 1$ ). Para aprobar las PL la nota mínima deberá ser 5. Sólo hay una única oportunidad (no se puede recuperar esta parte).

La parte de Redacción de un artículo en la Wikipedia (VA) se evaluará de manera grupal. Cada grupo tendrá un tema relacionado a los contenidos de la asignatura para desarrollar en la Wikipedia (puede que el artículo ya exista o sea de nueva creación). El artículo se irá desarrollando durante el curso y se discutirá con el responsable de la asignatura y los miembros del grupo en horas de tutoría. La nota de cada miembro del grupo se determinará por la calidad del artículo y participación en la discusión. Sólo hay una única oportunidad (no se puede recuperar esta parte). En consecuencia, en caso de que un/a alumno/a no tenga nota de revisión del artículo, no podrá optar al excelente en la recuperación.

Los ejercicios prácticos (PA) consistirán en trabajar problemas de programación muy concretos y en algunos casos relacionados con los que se encontrarán en las prácticas de laboratorio. Se quiere que cada estudiante estudie la resolución de un conjunto de problemas concretos de forma aislada al caso más general de la práctica. El valor de estos ejercicios es del 30% de la nota final y dada su naturaleza y objetivo no son recuperables.

La nota final de la asignatura será la suma ponderada de las notas de cada una de las cuatro actividades: 30% de Teoría, 10% Redacción de un artículo en la Wikipedia, 30% Resolución de ejercicios prácticos individuales y 30% de Resolución de prácticas de laboratorio. El resultado para superarla deberá ser  $\geq 5$ .

En caso de no superar la asignatura por no alcanzar la puntuación mínima en alguno de los apartados (Teoría o Prácticas de Laboratorio), aunque al hacer la media ponderada la nota final fuera igual o superior a 5 la nota que se pondrá en el expediente será de 4,5.

En caso de que la media no llegue a 5 la nota que figurará en el expediente será la nota media obtenida numéricamente.

Si el/la alumno/a entrega cualquier actividad, se entiende que se presenta en la asignatura y será evaluado/a. Si no entrega ninguna actividad, entonces se puede considerar No evaluable.

Otorgar una calificación de matrícula de honor es decisión del profesorado responsable de la asignatura. La normativa de la UAB indica que las MH sólo se podrán conceder a estudiantes que hayan obtenido una calificación final igual o superior a 9.00. Se puede otorgar hasta un 5% de MH del total de estudiantes matriculados.

Las fechas de evaluación continua y entrega de trabajos se publicarán en el campus virtual y pueden estar sujetos a cambios de programación por motivos de adaptación a posibles incidencias; siempre se informará en el campus virtual sobre estos cambios ya que se entiende que el CV es el mecanismo habitual de intercambio de información entre el profesorado y los/las estudiantes.

Para cada actividad de evaluación, se indicará un lugar, fecha y hora de revisión en la que el estudiante podrá revisar la actividad con el/la profesor/a. En este contexto, se podrán hacer reclamaciones sobre la nota de la actividad, que serán evaluadas por el profesorado responsable de la asignatura. Si el/la estudiante no se presenta en esta revisión,

no se revisará posteriormente esta actividad.

## RESUMEN

Si  $((T \geq 5) \text{ y } (P \geq 5))$  entonces

$$NF = 0.3 * T + 0.1 * VA + 0.3 * PA + 0.3 * PL$$

Si  $(NF \geq 5)$  entonces APROBADO

sino SUSPENDIDO

sino SUSPENDIDO

Estudiantes repetidores: Los alumnos repetidores o repetidoras que tengan aprobadas las prácticas de laboratorio pueden pedir la convalidación de esta parte de la asignatura. El resto de actividades de evaluación deberán hacerlas en las mismas condiciones que los/las otros/as estudiantes.

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;

- 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.

En caso de no superar la asignatura debido a que alguna de las actividades de evaluación no alcanza la nota mínima requerida, la nota numérica del expediente será el valor menor entre 4.5 y la media ponderada de las notas. Con las excepciones de que se otorgará la calificación de "No Evaluable" a los y las estudiantes que no participen en ninguna de las actividades de evaluación, y de que la nota numérica del expediente será el valor menor entre 3.0 y la media ponderada de las notas en caso de que el estudiante haya cometido irregularidades en un acto de evaluación (y por tanto no será posible el aprobado por compensación). En ediciones futuras de esta asignatura, el estudiante que haya cometido irregularidades en un acto de evaluación no se le convalidará ninguna de las actividades de evaluación realizadas.

En resumen: copiar, dejar copiar o plagiar (o el intento de) en cualquiera de las actividades de evaluación equivale a un SUSPENSO, no compensable y sin convalidaciones de partes de la asignatura en cursos posteriores.

## Actividades de evaluación

Título	Peso	Horas	ECTS	Resultados de aprendizaje
Artículo Viquipèdia	10%	1	0,04	5, 2, 4, 6, 7
Control individual parcial 1	15%	2	0,08	1, 3, 4, 8
Control individual parcial 2	15%	2	0,08	1, 3, 4, 8
Entrega de prácticas	30%	3	0,12	1, 5, 2, 3, 4, 6, 8
Realización de ejercicios prácticos individuales	30%	6	0,24	1, 5, 2, 3, 4, 6, 8

## Bibliografía

- Parallel Programming Concepts and Practice. Bertil Schmidt, Jorge González-Domínguez, Christian Hundt, Moritz Schlarb. Morgan Kaufmann, 2018
- An Introduction to Parallel Programming, 2nd edition. Peter S. Pacheco. Morgan Kaufmann. 2018
- Programming Massively Parallel Processors, A Hands-on Approach. David B. Kirk, Wen-mei W. Hwu. Morgan Kaufmann. 3th Edition. 2018
- Computer Architecture. A quantitative approach. John L. Hennessy, David A. Patterson. Morgan Kaufmann. 6a edición. 2018
- 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.
- Parallel Programming with MPI, Peter Pacheco, Morgan Kaufman, 1996
- OpenACC Programming and Best Practices Guide ([https://www.openacc.org/sites/default/files/inline-files/OpenACC\\_Programming\\_Guide\\_0.pdf](https://www.openacc.org/sites/default/files/inline-files/OpenACC_Programming_Guide_0.pdf))

## Software

En este curso trabajaréis sobre clusters propios del Departamento de Arquitectura de Computadores y Sistemas Operativos, en consecuencia, necesitaréis un cliente ssh para realizar las conexiones (integrado en

cualquier computador con sistema operativo Linux) y es muy recomendable el uso de un entorno de desarrollo que permita la conexión remota (como el MobaXTerm para Windows [<https://mobaxterm.mobatek.net/>] o el Visual Studio Code para todas las plataformas [<https://code.visualstudio.com/>]).