

Programación Paralela

Código: 106557
Créditos ECTS: 6

2025/2026

Titulación	Tipo	Curso
Inteligencia Artificial / Artificial Intelligence	OT	3
Inteligencia Artificial / Artificial Intelligence	OT	4

Contacto

Nombre: Carlos Carrillo Jordan

Correo electrónico: carles.carrillo@uab.cat

Equipo docente

Christian Guzman Ruiz

Idiomas de los grupos

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

Prerrequisitos

Aunque no hay prerrequisitos formalmente establecidos y es responsabilidad de la propia asignatura proporcionar a los alumnos y alumnas 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 Programación I y II, Fundamentos de Computación).

Objetivos y contextualización

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

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

Todos los componentes descritos en esta asignatura deben permitir al alumnado 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

Inteligencia Artificial / Artificial Intelligence

- Analizar y resolver problemas de forma efectiva, generando propuestas innovadoras y creativas para alcanzar los objetivos.
- Conceptualizar y modelar alternativas de soluciones complejas a problemas de aplicación de la inteligencia artificial en diferentes ámbitos, y planificar y gestionar proyectos para el diseño y desarrollo de prototipos que demuestren la validez del sistema propuesto.
- Desarrollar pensamiento crítico para analizar de forma fundamentada y argumentada alternativas y propuestas tanto propias como ajenas.
- Diseñar, implementar, analizar y validar soluciones algorítmicas eficientes y robustas a problemas computacionales derivados del diseño de sistemas inteligentes.
- Introducir cambios en los métodos y los procesos del ámbito de conocimiento para dar respuestas innovadoras a las necesidades y demandas de la sociedad.
- Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.
- Trabajar cooperativamente para la consecución de objetivos comunes, asumiendo la propia responsabilidad y respetando el rol de los diferentes miembros del equipo.

Resultados de aprendizaje

1. Analizar el rendimiento de aplicaciones, detectar cuellos de botella y aplicar posibles optimizaciones.
2. Analizar los modelos, paradigmas y lenguajes de programación paralela disponibles para determinar el que mejor se adapta a las necesidades de una aplicación.
3. Analizar y evaluar arquitecturas de computadores en plataformas paralelas y distribuidas.
4. Analizar y resolver problemas de forma efectiva, generando propuestas innovadoras y creativas para alcanzar los objetivos.
5. Conocer y aplicar los principios fundamentales y técnicas básicas de la programación paralela, concurrente y distribuida.
6. Desarrollar pensamiento crítico para analizar de forma fundamentada y argumentada alternativas y propuestas tanto propias como ajenas.
7. Desarrollar y optimizar software para las plataformas paralelas y distribuidas basadas en los paradigmas existentes más habituales.
8. Implementar y optimizar aplicaciones basadas en las funcionalidades y estructura de los sistemas paralelos, distribuidos y cloud.
9. Ponderar los riesgos y las oportunidades de las propuestas de mejora tanto propias como ajenas.
10. Proponer nuevos métodos o soluciones alternativas fundamentadas.
11. Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.
12. Trabajar cooperativamente para la consecución de objetivos comunes, asumiendo la propia responsabilidad y respetando el rol de los diferentes miembros del equipo.
13. 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 avanzada en C e integración de código C con Python. Concurrencia: concepto condiciones de carrera, región crítica y mecanismos de exclusión mutua.

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

Tema 2: Clasificación de paralelismo

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

Tema 3: Algoritmos paralelos

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

Tema 4: Programación paralela

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

Tema 5: Análisis de prestaciones

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

Actividades formativas y Metodología

Título	Horas	ECTS	Resultados de aprendizaje
Tipo: Dirigidas			
Práctica (Resolución de proyectos prácticos de laboratorio)	16	0,64	1, 3, 4, 7, 6, 8, 10, 12, 13
Problemas (Resolución de ejercicios)	7	0,28	1, 2, 3, 4, 5, 6, 10, 11, 12
Teoría	12	0,48	1, 2, 3, 5, 7, 8
Tipo: Autónomas			
Estudio autónomo	45	1,8	1, 2, 4, 5, 7, 8, 13
Preparación de problemas i ejercicios prácticos	15	0,6	1, 2, 4, 5, 7, 6, 8, 12
Preparación de proyectos	40	1,6	1, 4, 7, 6, 8, 10, 12, 13

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 desgranarán 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 evaluaciones continuas de grupos de temas.
- Clases de problemas. Todos los temas irán acompañados de una relación de problemas que el alumno debe intentar resolver. En este sentido, y a medida que el/la alumno/a 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 de problemas es hacer de puente entre las clases teóricas y las clases en laboratorio, que promoverá la capacidad de análisis y síntesis, el razonamiento crítico, y entrenar al o la estudiante en la resolución de problemas. Aquellos que el/la profesor/a considere de mayor interés o en los que los/las estudiantes encuentren mayor dificultad serán corregidos en la pizarra. Antes del comienzo de cada temario de problemas el/la profesor/a podrá proponer una lista de ejercicios que los/las alumnos/as deberán resolver.

- Clases de proyectos. La parte práctica de los temas teóricos quedará completada con sesiones en el laboratorio, donde el/la alumno/a desarrollará un conjunto de programas y deberá intentar resolver un problema concreto que recibirá al comenzar el temario. Los programas desarrollados (proyectos) se deberán entregar en la clase en las fechas señaladas. Las prácticas se desarrollarán en grupos de dos alumnos/as. Las clases incluyen 9 sesiones en el laboratorio, de 2 horas de duración, donde el/la alumno/a realizará el desarrollo de los programas.
- Realización de ejercicios. Durante el curso habrá un conjunto de ejercicios básicos, cortos y prácticos relacionados con la teoría y las clases prácticas de problemas. Cada grupo (de 2 personas) solucionará estos ejercicios y realizará la entrega con la fecha correspondiente. Con este ejercicio se busca que el/la estudiante sea capaz de resolver de forma teórica a los conceptos trabajados en teoría y después pueda aplicarlos al caso práctico de laboratorio.

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.

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
Proyectos prácticos 1	10%	1	0,04	2, 3, 4, 6, 8, 9, 10, 12, 13
Proyectos prácticos 2	10%	1	0,04	2, 3, 4, 6, 8, 9, 10, 12, 13
Proyectos prácticos 3	10%	1	0,04	2, 3, 4, 6, 8, 9, 10, 12, 13
Realización de ejercicios prácticos	20%	8	0,32	1, 4, 5, 7, 8, 11, 12, 13
1er Control parcial individual	25%	2	0,08	1, 4, 5, 7
2do Control parcial individual	25%	2	0,08	1, 4, 5, 7

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 cinco actividades son:

1. Teoría (T)
2. Resolución de los proyectos prácticos de laboratorio (PL)
3. Realización de ejercicios prácticos (EXER).

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 tenemos asignado en la semana de exámenes de recuperación de junio. Se podrán recuperar por separado las partes que no se hayan superado en los controles parciales de teoría. La nota mínima para aprobar esta parte de Teoría es ≥ 5 . En la recuperación sólo se pueden recuperar los exámenes suspendidos, no se puede subir la nota obtenida (en el caso de aprobar la parte de Teoría). La nota máxima que puede obtenerse en la recuperación es un 7.

La parte de Resolución de proyectos prácticos de laboratorio (PL) se evaluará de manera grupal. Habrá tres entregas. La nota final saldrá de la suma ponderada de las tres entregas ($X_0 * \text{Entrega 1} + X_1 * \text{Entrega 2} + X_2 * \text{Entrega 3}$, $X_0 + X_1 + X_2 = 1$). Habrá 3 pruebas cortas y escritas de validación individual de las prácticas en el horario de clase (la tercera prueba en el horario establecido por la coordinación para los segundos parciales de junio). Nota proyectos prácticos (PL) = Laboratorio * Validación. Para aprobar las PL la nota mínima deberá ser ≥ 5 . No se podrá recuperar. La asistencia es obligatoria.

Los ejercicios prácticos (EXER) se realizarán en grupos de 2 personas y consistirán en trabajar problemas de programación muy concretos, cortos y 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 en el 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: 50% de Teoría, 20% Resolución de ejercicios prácticos y 30% de Resolución de proyectos prácticos de laboratorio. El resultado deberá ser ≥ 5 .

En caso de no superar la asignatura por no alcanzar la puntuación mínima en alguno de los apartados (Teoría o Proyectos prácticos 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 >= 5) y (PL >= 5)) entonces

NF = 0.5 * T + 0.2 * EXER + 0.3 * PL

Si (NF >= 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.

Evaluación Única: Esta asignatura NO PREVÉ EVALUACIÓN ÚNICA.

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.

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.

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 alter. 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 Kauffman, 1996
- OpenACC Programming and Best Practices Guide (https://www.openacc.org/sites/default/files/inline-files/OpenACC_Programming_Guide_0.pdf)

Software

OpenMP

MPI

OpenACC/CUDA

gestor de colas (SLURM)

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

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

Grupos e idiomas de la asignatura

La información proporcionada es provisional hasta el 30 de noviembre de 2025. A partir de esta fecha, podrá consultar el idioma de cada grupo a través de este [enlace](#). Para acceder a la información, será necesario introducir el CÓDIGO de la asignatura

Nombre	Grupo	Idioma	Semestre	Turno
(PAUL) Prácticas de aula	711	Inglés	segundo cuatrimestre	mañana-mixto
(PLAB) Prácticas de laboratorio	711	Inglés	segundo cuatrimestre	mañana-mixto
(TE) Teoría	71	Inglés	segundo cuatrimestre	mañana-mixto