

**Guía docente de la asignatura Computación Paralela**

Asignatura	COMPUTACION PARALELA		
Materia	COMPUTACION		
Módulo	TECNOLOGIAS ESPECIFICAS		
Titulación	GRADO EN INGENIERÍA INFORMÁTICA		
Plan	545	Código	46929
Periodo de impartición	2do CUATRIMESTRE	Tipo/Carácter	OPTATIVA (Mención IS, Mención TI, Mención CO)
Nivel/Ciclo	GRADO	Curso	3º
Créditos ECTS	6 ECTS		
Lengua en que se imparte	CASTELLANO		
Profesor/es responsable/s	ARTURO GONZALEZ ESCRIBANO		
Datos de contacto (E-mail, teléfono...)	TELÉFONO: 983 423000 ext. 5613 E-MAIL: arturo@infor.uva.es		
Horario de tutorías	Véase www.inf.uva.es → Alumno → Apoyo → Tutorías		
Departamento	INFORMÁTICA (ATC, LSI, CCIA)		



1. Situación / Sentido de la Asignatura

1.1 Contextualización

Esta asignatura es parte de la materia “Complementos de Ingeniería de Computadores”, compuesta además por las asignaturas Arquitecturas de Computación Avanzadas”, “Sistemas Empotrados”, “Rendimiento y Evaluación de Computadoras”, “Hardware Empotrado”, y “Diseño de Sistemas Digitales”. Se trata de ofrecer al alumno una visión unificada de la programación de sistemas que permiten explotar paralelismo.

1.2 Relación con otras materias

Está relacionada con la asignatura “Arquitecturas de Computación Avanzada”, que estudia los componentes hardware de los sistemas cuya programación es objeto de estudio en esta asignatura.

1.3 Prerrequisitos

Se presupone que el alumno dispone de experiencia previa en el manejo del lenguaje de programación C y el interfaz de comandos de UNIX. Se presupone que tiene conocimientos generales sobre el funcionamiento y arquitectura de sistemas de computo y sistemas operativos.

2. Competencias

2.1 Generales y transversales

Código	Descripción
CG6	Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación especificadas a continuación en esta sección de la memoria.
CG8	Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.
CT1	Capacidad de análisis y de síntesis.
CT3	Comunicación oral y escrita en la lengua propia.
CT5	Habilidades de gestión de la información
CT7	Toma de decisiones
CT9	Trabajo en equipo
CT13	Capacidad de aplicar los conocimientos en la práctica
CT14	Capacidad de aprender
CT15	Capacidad de adaptarse a nuevas situaciones
CT16	Habilidad para trabajar de forma autónoma

2.2 Específicas

Código	Descripción
CO1	Capacidad para tener un conocimiento profundo de los principios fundamentales y modelos de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la informática.
CO3	Capacidad para evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.
IC3	Capacidad de analizar y evaluar arquitecturas de computadores, incluyendo plataformas paralelas y distribuidas, así como desarrollar y optimizar el software de para las mismas.
IC5	Capacidad de analizar, evaluar y seleccionar las plataformas hardware y software más adecuados para el soporte de aplicaciones empujadas y de tiempo real.

3. Objetivos

- Comprender los modelos y técnicas empleados para evaluar la mejora del rendimiento en las soluciones que explotan paralelismo.
- Conocer los principios de diseño y la estructura de los sistemas de memoria compartida, distribuida, los sistemas multinúcleo y los híbridos y heterogéneos.
- Ser capaz de implementar soluciones paralelas en programación multihilo.
- Ser capaz de implementar soluciones paralelas en programación con paso de mensajes.
- Ser capaz de implementar soluciones paralelas en programación de GPUs.

4. Bloques temáticos

Bloque 1: Sistemas de computación paralela.

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación

Este bloque temático comprende una descripción de los conceptos básicos de funcionamiento del hardware y software paralelo, así como una visión general de los mecanismos de programación de sistemas paralelos basados en el uso conjunto de múltiples CPUs, tanto en su vertiente de memoria compartida como distribuida, y la programación de aplicaciones masivamente paralelas para su ejecución en unidades de procesamiento gráfico de propósito general (GP-GPUs).

b. Objetivos de aprendizaje

- Comprender los modelos y técnicas empleados para evaluar la mejora del rendimiento en las soluciones que explotan paralelismo.
- Conocer los principios de diseño y la estructura de los sistemas de memoria compartida, distribuida, los sistemas multinúcleo y los híbridos y heterogéneos.
- Ser capaz de implementar soluciones paralelas en programación multihilo.
- Ser capaz de implementar soluciones paralelas en programación con paso de mensajes.
- Ser capaz de implementar soluciones paralelas en programación de GPUs.

c. Contenidos

Tema 1: Introducción a la computación paralela

- Presentación del profesorado, descripción general del temario, mecanismos de evaluación.
- Concurrencia y Paralelismo: conceptos fundamentales.
- Complejidad y paralelismo
- Arquitecturas paralelas
 - Arquitecturas de memoria compartida.
 - Arquitecturas de memoria distribuida.
 - Arquitecturas GP-GPU.
 - Arquitecturas heterogéneas.
 - Otras arquitecturas emergentes y de propósito específico: FPGA, ASIC, Xeon Phi.
- Modelos de cómputo
 - Modelos de computo paralelo.
 - Análisis de algoritmos en PRAM.
 - Paralelización de código vs. Programación en paralelo.
 - Modelos de programación.
- Algoritmos y diseño de aplicaciones paralelas
 - Paralelismo de datos y de tareas.
 - Particiones y patrones estáticos.
 - Estructuras dispersas.
 - Particiones y patrones dinámicos (Branch & Bound, Pipelines, Granjas)
 - No determinismo

Tema 2: Programación de sistemas de memoria compartida con OpenMP

- ¿Qué es OpenMP?
- Modelo de ejecución y memoria compartida de OpenMP.
- Componentes de OpenMP: directivas y cláusulas.
- Directivas para la construcción de paralelismo.
- Funciones y variables de entorno.
- Directivas de sincronización.
- Tareas OpenMP.

Tema 3: Programación de sistemas de memoria distribuida con MPI

- Introducción a MPI: Motivación, objetivos, historia y terminología
- Interfaz MPI: inicialización y finalización, mensajes MPI, Datatypes
- Comunicaciones punto a punto
- Modos de comunicación y comunicaciones no bloqueantes
- Comunicaciones colectivas: broadcast, scatter/gather, reducciones
- Tipos de datos derivados
- Creación y manejo de comunicadores
- Tratamiento y depuración de errores

Tema 4: Programación GP-GPUs con CUDA

- Introducción a las GPUs.
- Qué es CUDA y para qué se utiliza.
- Modelo arquitectónico CUDA: recursos y jerarquía de memoria.
- Modelo de programación CUDA: estructura, terminología, warps.
- Entorno de ejecución CUDA: compilación, sistemas multi-GPUs
- Sintaxis CUDA: variables, funciones, inicializaciones.
- Consideraciones sobre optimización.

d. Métodos docentes

- Clase magistral participativa
- Estudio de casos en aula y en laboratorio
- Resolución de problemas
- Desarrollo de proyectos
- Metodologías de gamificación competitiva y colaborativa

e. Plan de trabajo

Se publicará en la plataforma de docencia virtual

f. Evaluación

- Evaluación continua basada en exámenes de preguntas cortas o tipo test, y desarrollo de supuestos prácticos.

g. Bibliografía básica

- *Introducción a la programación paralela*, Francisco Almeida et al., Paraninfo, ISBN 8497326741

h. Bibliografía complementaria

- *Fundamentos de Informática y Programación en C*, Diego R. Llanos Ferraris, primera edición, editorial Paraninfo, 2010, ISBN 978-84-9732-792-3.
- *Multicore and GPU Programming: An Integrated Approach*. Gerassimos Barlas. Morgan Kaufmann, 2015. ISBN: 978-0-12-417137-4
- *Using OpenMP: portable shared memory parallel programming*, Barbara Chapman, Gabriele Jost, Ruud van der Pas. MIT press, 2008, ISBN: 978-0-262-53302-7
- *Using MPI: Portable Parallel Programming with the Message-passing Interface*, 3º ed. W. Gropp, E. Lusk, and A. Skjellum, MIT Press, 2015, ISBN 978-0262527392
- *Programming Massively Parallel Processors: A Hands-on Approach*, 2ª edición, David Kirk and Wen-mei W. Hwu, Morgan Kaufmann, 2013, ISBN 978-0-12-415992-1
- *CUDA By Example: An Introduction to General-Purpose GPU*, Jason Sanders and Edward Kandrot, Addison-Wesley, ISBN 013138768

i. Recursos necesarios

- Laboratorio de la Escuela de Informática asignado al programa GPU Education Center.
- Acceso remoto a las instalaciones y servidores de cómputo paralelo del Departamento y Escuela de Informática.
- Acceso al aula virtual

5. Temporalización (por bloques temáticos)

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1: Sistemas de computación paralela	6 ECTS	Semanas 1 a 15

6. Métodos docentes y principios metodológicos

Actividad	Metodología
Clase de teoría	<ul style="list-style-type: none">Clase magistral participativaEstudio de casos en aulaResolución de problemas
Clase práctica	<ul style="list-style-type: none">Clase magistral participativaRealización de proyectos guiado por el profesor, que encargará y guiará el trabajo que se realizará en grupos (2/3 alumnos), siguiendo un enfoque colaborativo. Técnicas de gamificación y participación en concursos de programación.
Seminarios	<ul style="list-style-type: none">Talleres de aprendizaje
Tutoría	<ul style="list-style-type: none">Evaluación de los contenidos teóricos y de los proyectos

7. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	30	Estudio y trabajo autónomo individual	45
Clases prácticas de aula (A)	0	Estudio y trabajo autónomo grupal	45
Laboratorios (L)	30		
Prácticas externas, clínicas o de campo	0		
Seminarios (S)	6		
Tutorías grupales (TG)	0		
Evaluación	0		
Total presencial	60	Total no presencial	90

8. Sistema y características de la evaluación

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Práctica: Evaluación del desarrollo e informes de prácticas.	65	Prácticas a desarrollar en semanas concretas, concursos de programación (ver cronograma de actividades).
Teoría: Evaluaciones intermedias de carácter teórico.	35	Cuatro breves evaluaciones de teoría en semanas concretas al final de cada tema (ver cronograma de actividades)
Bonificaciones relacionadas con gamificación.	10	Participación en foros, tareas de apoyo técnico a compañeros, hitos en los concursos de programación.

Notas importantes:

1. La nota de teoría se obtendrá como la media aritmética simple de las calificaciones obtenidas en cuatro breves cuestionarios de teoría basados en preguntas cortas o de tipo test.
2. La nota de prácticas se obtendrá como una media ponderada de las calificaciones obtenidas en tres prácticas.
3. Para aprobar la asignatura a través del mecanismo de evaluación continua, hay que obtener:
 - Al menos un 4 sobre 10 en cada práctica.
 - Al menos un 3 sobre 10 de media en los exámenes teóricos.
 - Un 5 de nota media en la asignatura.
4. Los alumnos que no hayan obtenido un 5 de nota media ponderada durante el curso (siguiendo los criterios anteriores) deberán presentarse al examen final de la asignatura, tanto en su convocatoria ordinaria como extraordinaria.
5. La calificación obtenida en los exámenes finales constituirá el 100% de la nota del alumno, no computándose en este caso los porcentajes anteriores.
6. Se considerarán como "no presentados" a los alumnos que no se presenten a ninguna evaluación de teoría o de prácticas.

9. Consideraciones finales