



Guía docente de la asignatura

Asignatura	COMPUTACION PARALELA		
Materia	COMPLEMENTOS DE INGENIERÍA DE COMPUTADORES		
Módulo	(vacío)		
Titulación	GRADO EN INGENIERÍA INFORMÁTICA		
Plan	464	Código	45218
Periodo de impartición	1 ^{er} . CUATRIMESTRE	Tipo/Carácter	OPTATIVA
Nivel/Ciclo	GRADO	Curso	4º
Créditos ECTS	6 ECTS		
Lengua en que se imparte	CASTELLANO		
Profesor/es responsable/s	DIEGO RAFAEL LLANOS FERRARIS, ARTURO GONZALEZ ESCRIBANO		
Datos de contacto (E-mail, teléfono...)	TELÉFONO: 983 423000 ext. 5642 (Diego), ext. 5623 (Arturo) E-MAIL: { diegoarturo }@infor.uva.es		
Horario de tutorías	Véase www.uva.es → Centros → Campus de Valladolid → Escuela Técnica Superior de Ingeniería Informática → Tutorías		
Departamento	INFORMÁTICA (ATC, LSI, CCIA)		



1. Situación / Sentido de la Asignatura

1.1 Contextualización

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 los sistemas que requieren una programación paralela.

1.2 Relación con otras materias

Presenta relación 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 conoce el lenguaje C, lenguaje básico en la programación de sistemas paralelos, y que tiene conocimientos generales sobre el funcionamiento y arquitectura de los sistemas operativos.





2. Competencias

2.1 Generales

Código	Descripción
G03	Capacidad de análisis y de síntesis
G05	Comunicación oral y escrita en la lengua propia.
G08	Habilidades de gestión de la información
G10	Toma de decisiones
G12	Trabajo en equipo
G16	Capacidad de aplicar los conocimientos en la práctica
G18	Capacidad de aprender
G19	Capacidad de adaptarse a nuevas situaciones
G20	Capacidad de generar nuevas ideas.
G21	Habilidad para trabajar de forma autónoma

2.2 Específicas

Código	Descripción
IC3	Capacidad de analizar y evaluar arquitecturas de computadores, incluyendo plataformas paralelas y distribuidas, así como desarrollar y optimizar el software de dichos sistemas.
IC5	Capacidad de analizar, evaluar y seleccionar las plataformas hardware y software más adecuados para el soporte de aplicaciones empotradas y de tiempo real.

3. Objetivos

Código	Descripción
IC3.1	Comprender en qué circunstancias la programación paralela puede acortar el tiempo de resolución de problemas de cálculo.
IC3.2	Construir aplicaciones paralelas para su ejecución en un sistema de memoria compartida o distribuida, cubriendo las fases de compilación, desarrollo, depuración e inicialización.
IC3.3	Conocer el procedimiento de programación de aplicaciones para su ejecución en sistemas de procesamiento masivamente paralelo utilizando GPUs.



4. 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)	24		
Prácticas externas, clínicas o de campo	0		
Seminarios (S)	6		
Tutorías grupales (TG)	0		
Evaluación	0		
Total presencial	60	Total no presencial	90





5. Bloques temáticos

Bloque 1: Sistemas paralelos de memoria compartida y distribuida

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación

Este bloque temático se corresponde con los mecanismos de programación de sistemas paralelos basados en el uso conjunto de múltiples CPUs, tanto en su vertiente de memoria compartida como distribuida.

b. Objetivos de aprendizaje

Código	Descripción
CI14.1	Comprender en qué circunstancias la programación paralela puede acortar el tiempo de resolución de problemas de cálculo.
CI14.2	Construir aplicaciones paralelas para su ejecución en un sistema de memoria compartida o distribuida, cubriendo las fases de compilación, desarrollo, depuración e inicialización.

c. Contenidos

Tema 1: Fundamentos de la Programación Paralela

1. Introducción
2. Arquitectura de sistemas de memoria compartida y distribuida
3. Arquitectura de sistemas basados en el uso de GPUs
4. Análisis de costes
5. Introducción a los modelos de programación paralela.

Tema 2: Programación de sistemas de memoria compartida: OpenMP

1. ¿Qué es OpenMP?
2. Modelo de ejecución y memoria compartida de OpenMP.
3. Componentes de OpenMP: directivas y cláusulas.
4. Directivas para la construcción de paralelismo.
5. Funciones y variables de entorno.
6. Directivas de sincronización.
7. Tareas OpenMP.

Tema 3: Programación de sistemas de memoria distribuida: MPI

1. Introducción a MPI: Motivación, objetivos, historia y terminología
2. Interfaz MPI: inicialización y finalización, mensajes MPI, Datatypes
3. Implementaciones MPI.
4. Comunicaciones punto a punto: síncronas y asíncronas
5. Tipos de datos derivados
6. Comunicaciones colectivas: broadcast, scatter/gather, reducciones.



d. Métodos docentes

- Clase magistral participativa
- Estudio de casos en aula y en laboratorio
- Resolución de problemas
- Desarrollo de proyectos

e. Plan de trabajo

Además de conocer los fundamentos en los que se basa la asignatura, se propondrán dos trabajos de carácter práctico, junto con tres tests para reforzar los conocimientos adquiridos. Las fechas de realización de los tests se corresponderán con las semanas cuatro, siete y diez, y la de los trabajos con las semanas siete y diez.

f. Evaluación

- Evaluación continua
- Exámenes escritos
- 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.
- *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*, 2o ed. W. Gropp, E. Lusk, and A. Skjellum, MIT Press, 1999.

i. Recursos necesarios



Bloque 2: Sistemas Masivamente Paralelos (GPGPU)

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación

Este bloque comprende la programación de aplicaciones masivamente paralelas para su ejecución en unidades de procesamiento gráfico de propósito general (GP-GPUs).

b. Objetivos de aprendizaje

Código	Descripción
CI14.1	Comprender en qué circunstancias la programación paralela puede acortar el tiempo de resolución de problemas de cálculo.
CI14.3	Conocer el procedimiento de programación de aplicaciones para su ejecución en sistemas de procesamiento masivamente paralelo utilizando GPUs.

c. Contenidos

Tema 4: Programación de GP-GPUs: CUDA

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

d. Métodos docentes

- Clase magistral participativa
- Estudio de casos en aula y en laboratorio
- Resolución de problemas
- Desarrollo de proyectos

e. Plan de trabajo

Además de conocer los fundamentos en los que se basa la asignatura, se propondrá un trabajo de carácter práctico, junto con un test para reforzar los conocimientos adquiridos. Las fechas de entrega de los citados trabajos y la realización de los tests se corresponderá con la semana quince.

f. Evaluación

- Evaluación continua



- Exámenes escritos
- Supuestos prácticos

g. Bibliografía básica

- Nvidia CUDA Programming Guide 4.2 (disponible online)

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.
- *Programming Massively Parallel Processors: A Hands-on Approach*, David Kirk and Wen-mei W. Hwu, Morgan Kaufmann, ISBN 978-0-12-381472-2
- *CUDA By Example: An Introduction to General-Purpose GPU*, Jason Sanders and Edward Kandrot, Addison-Wesley, ISBN 013138768

i. Recursos necesarios





6. Temporalización (por bloques temáticos)

El número de semanas de un cuatrimestre son 15. Ejemplo:

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1: Sistemas paralelos de memoria compartida y distribuida	4 ECTS	Semanas 1 a 10
Bloque 2: Sistemas masivamente paralelos (GP-GPUs)	2 ECTS	Semanas 11 a 15

7. Sistema de calificaciones – Tabla resumen

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Evaluación de informes de prácticas (7)	65	informes de práctica a entregar en semanas concretas (ver cronograma de actividades).
Evaluaciones intermedias de carácter teórico y de tipo test (8)	35	Ocho breves evaluaciones de teoría en semanas concretas (ver cronograma de actividades)

Notas importantes:

- Sólo los alumnos que no hayan obtenido un 5 de nota media ponderada durante el curso (siguiendo los criterios de la tabla anterior) deberán presentarse al examen final de la asignatura, tanto en su convocatoria ordinaria como extraordinaria.
- La calificación obtenida en los exámenes finales constituirá el 100% de la nota del alumno, no computándose en este caso los porcentajes de la tabla anterior.
- Se considerarán como “no presentados” a los alumnos que, no habiendo aprobado la asignatura a través de sus evaluaciones intermedias, no se presenten al examen final de la asignatura.

8. Anexo: Métodos docentes

Si no se han incluido en los bloques temáticos, se puede hacer referencia a éste anexo. Ejemplo:

Actividad	Metodología
Clase de teoría	<ul style="list-style-type: none"> Clase magistral participativa Estudio de casos en aula Resolución de problemas
Clase práctica	<ul style="list-style-type: none"> Clase magistral participativa Realización de un proyecto guiado por el profesor, que encargará y guiará el



	trabajo que se realizará en grupos (2/3 alumnos), siguiendo un enfoque colaborativo.
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

9. Anexo: Cronograma de actividades previstas

Semana	Fecha inicio semana	Contenidos	Actividades previstas	Entrega de trabajos	Horas presenciales	Horas no presenciales
1	23 sep	Tema 1	Teoría, problemas y prácticas	No	4	6
2	30 sep	Tema 1	Teoría, problemas y prácticas	No	4	6
3	7 oct	Tema 1	Teoría, problemas y prácticas	No	4	6
4	14 oc	Tema 2	Teoría, problemas y prácticas	Evaluación test	4	6
5	21 oct	Tema 2	Teoría, problemas y prácticas	No	4	6
6	28 oct	Tema 2	Teoría, problemas y prácticas	No	4	6
7	4 nov	Tema 3	Teoría, problemas y prácticas	Evaluación test, entrega trabajo.	4	6
8	11 nov	Tema 3	Teoría, problemas y prácticas	No	4	6
9	18 nov	Tema 3	Teoría, problemas y prácticas	No	4	6
10	25 nov	Tema 3	Teoría, problemas y prácticas	Evaluación test, entrega trabajo.	4	6



11	2 dic	Tema 4	Teoría, problemas y prácticas	No	4	6
12	9 dic	Tema 4	Teoría, problemas y prácticas	No	4	6
13	16 dic	Tema 4	Teoría, problemas y prácticas	No	4	6
14	6 ene	Tema 4	Teoría, problemas y prácticas	No	4	6
15	13 ene	Tema 4	Teoría, problemas y prácticas	Evaluación test; entrega trabajo.	4	6

