



Técnicas de Computación Científica

Guía de Aprendizaje – Información al estudiante

1. Datos Descriptivos

Asignatura	Técnicas de Computación Científica
Materia	Ingeniería
Departamento responsable	Lenguajes y Sistemas Informáticos en Ingeniería del Software
Créditos ECTS	3
Carácter	Optativa
Titulación	Graduado/a en Ingeniería Informática por la Universidad Politécnica de Madrid
Curso	4º
Especialidad	No aplica

Curso académico	2011-2012
Semestre en que se imparte	Sept.-Enero (puede cambiar según planificación docente)
Semestre principal	Sept.-Enero
Idioma en que se imparte	Español
Página Web	



2. Profesorado

NOMBRE Y APELLIDO	DESPACHO	Correo electrónico
Vicente Martín Ayuso	5209	vicente@fi.upm.es

3. Conocimientos previos requeridos para poder seguir con normalidad la asignatura

Asignaturas superadas	<ul style="list-style-type: none">•
Otros resultados de aprendizaje necesarios	<ul style="list-style-type: none">• Conocimientos básicos de algorítmica y arquitectura de ordenadores.



4. Objetivos de Aprendizaje

COMPETENCIAS ASIGNADAS A LA ASIGNATURA Y SU NIVEL DE ADQUISICIÓN		
Código	Competencia	Nivel
CE	Capacidad para usar consistentemente los recursos computacionales.	3
CE-2	Formalización y la especificación de problemas reales cuya solución requiere el uso de la informática	4
CE-3	Capacidad de elegir y usar los métodos analíticos y de modelización relevantes	3
CE-12	Adquisición de conocimientos avanzados en el campo de ciencias de la computación	4
CE-53	Capacidad de trabajar de forma efectiva como individuo y como miembro de un equipo	3
CE-15	Capacidad de leer y comprender así como de catalogar y clasificar científicamente publicaciones dentro de su campo de estudio.	3

LEYENDA: Nivel de competencia: conocimiento (1), comprensión (2), aplicación (3) y análisis y síntesis (4),



RESULTADOS DE APRENDIZAJE DE LA ASIGNATURA			
Código	Resultado de aprendizaje	Competencias asociadas	Nivel de adquisición
RA1	Los estudiantes usan eficientemente los recursos computacionales	CU , CE3, CE53-54	4
RA2	Los estudiantes comprenden y evalúan las distintas técnicas que influyen en el uso eficiente de un ordenador, secuencial y paralelo.	CU,CE2-4,C54	
RA3	Los estudiantes son capaces de aplicar técnicas de optimización y paralelización para resolver problemas reales.	CU, CE2-3, CE 53-54	



5. Sistema de evaluación de la asignatura

INDICADORES DE LOGRO		
Ref	Indicador	Relaciona-do con RA
I1	Diseñar e implementar programas que usen de manera eficiente los recursos computacionales de ordenadores secuenciales.	RA1, RA3
I2	Conocer las implicaciones de rendimiento que conlleva la implementación en máquina de los algoritmos.	RA1, RA3
I3	Usar ordenadores de alto rendimiento para ejecutar aplicaciones en ciencia e ingeniería.	RA2
I4	Diseñar e implementar una aplicación paralela para resolver un problema real.	RA2, RA3

EVALUACION SUMATIVA			
Breve descripción de las actividades evaluables	Momento	Lugar	Peso en la calif.
Proyecto 1: Problemas secuenciales.	Durante el curso (ver CRONOGAMA) Fecha definitiva a confirmar en clase	Entrega a través de Moodle.	50% Se precisa superar un 3.0
Proyecto 2: Problemas paralelos.	Durante el curso (ver CRONOGRAMA) fecha límite exacta a confirmar en clase)	Se entregan a través de Moodle desde cualquier lugar.	50.00% Se precisa superar un 3.0
			Total: 100%



POLITÉCNICA

UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA
Campus de Montegancedo
Boadilla del Monte. 28660 Madrid

CRITERIOS DE CALIFICACIÓN

El método de evaluación estará basado en la ejecución de dos proyectos durante el curso. Uno de ellos estará centrado en arquitecturas secuenciales y el otro en arquitecturas paralelas. Ambos proyectos tienen el mismo peso. Previamente a cada proyecto habrá una fase de definición del mismo en el que se discutirá el trabajo a realizar.



6. Contenidos y Actividades de Aprendizaje

CONTENIDOS ESPECÍFICOS		
Bloque / Tema / Capítulo	Apartado	Indicadores Relacionados
Tema 1: optimización en arquitecturas secuenciales.	<ul style="list-style-type: none">• Overview de arquitecturas y subsistemas de memoria.• Optimización del subsistema de memoria.• Optimización de la CPU	11,16
Tema 2: Introducción al paralelismo y arquitecturas paralelas	Aarquitecturas paralelas. Evolución de las mismas, expectativas y herramientas.	12,15
Tema 3: Programación paralela.	Principales paradigmas de programación paralela: HPF, OpenMp, MPI y UPC. Benchmarking.	13,14,15,16
Tema 4: Green computing	Eficiencia energética en instalaciones y ordenadores de alto rendimiento.	13,14,15,16

7. Breve descripción de las modalidades organizativas utilizadas y de los métodos de enseñanza empleados

Tabla 7. Modalidades organizativas de la enseñanza

MODALIDADES ORGANIZATIVAS DE LA ENSEÑANZA		
Escenario	Modalidad	Finalidad
	Clases Teóricas	<i>Hablar a los estudiantes</i>
	Seminarios-Talleres	<i>Construir conocimiento a través de la interacción y la actividad de los estudiantes</i>
	Clases Prácticas	<i>Mostrar a los estudiantes cómo deben actuar</i>
	Prácticas Externas	<i>Completar la formación de los alumnos en un contexto profesional</i>
	Tutorías	<i>Atención personalizada a los estudiantes</i>
	Trabajo en grupo	<i>Hacer que los estudiantes aprendan entre ellos</i>
	Trabajo autónomo	<i>Desarrollar la capacidad de autoaprendizaje</i>

Tabla 8. Métodos de enseñanza

MÉTODOS DE ENSEÑANZA		
	Método	Finalidad
	Método Expositivo/Lección Magistral	Transmitir conocimientos y activar procesos cognitivos en el estudiante
	Estudio de Casos	Adquisición de aprendizajes mediante el análisis de casos reales o simulados
	Resolución de Ejercicios y Problemas	Ejercitar, ensayar y poner en práctica los conocimientos previos
	Aprendizaje Basado en Problemas (ABP)	Desarrollar aprendizajes activos a través de la resolución de problemas
	Aprendizaje orientado a Proyectos	Realización de un proyecto para la resolución de un problema, aplicando habilidades y conocimientos adquiridos
	Aprendizaje Cooperativo	Desarrollar aprendizajes activos y significativos de forma cooperativa
	Contrato de Aprendizaje	Desarrollar el aprendizaje autónomo

Se conoce como método expositivo "la presentación de un tema lógicamente estructurado con la finalidad de facilitar información organizada siguiendo criterios adecuados a la finalidad pretendida". Esta metodología -también conocida como lección (lecture)- se centra fundamentalmente en la exposición verbal por parte del profesor de los contenidos sobre la materia objeto de estudio. El término "lección magistral" se suele utilizar para denominar un tipo específico de lección impartida por un profesor en ocasiones especiales.

Análisis intensivo y completo de un hecho, problema o suceso real con la finalidad de conocerlo, interpretarlo, resolverlo, generar hipótesis, contrastar datos, reflexionar, completar conocimientos, diagnosticarlo y, en ocasiones, entrenarse en los posibles procedimientos alternativos de solución.

Situaciones en las que se solicita a los estudiantes que desarrollen las soluciones adecuadas o correctas mediante la ejercitación de rutinas, la aplicación de fórmulas o algoritmos, la aplicación de procedimientos de transformación de la información disponible y la interpretación de los resultados. Se suele utilizar como complemento de la lección magistral.

Método de enseñanza-aprendizaje cuyo punto de partida es un problema que, diseñado por el profesor, el estudiante ha de resolver para desarrollar determinadas competencias previamente definidas.

Método de enseñanza-aprendizaje en el que los estudiantes llevan a cabo la realización de un proyecto en un tiempo determinado para resolver un problema o abordar una tarea mediante la planificación, diseño y realización de una serie de actividades, y todo ello a partir del desarrollo y aplicación de aprendizajes adquiridos y del uso efectivo de recursos.

Enfoque interactivo de organización del trabajo en el aula en el cual los alumnos son responsables de su aprendizaje y del de sus compañeros en una estrategia de corresponsabilidad para alcanzar metas e incentivos grupales. Es tanto un método, a utilizar entre otros, como un enfoque global de la enseñanza, una filosofía.

Un acuerdo establecido entre el profesor y el estudiante para la consecución de unos aprendizajes a través de una propuesta de trabajo autónomo, con una supervisión por parte del profesor y durante un período determinado. En el contrato de aprendizaje es básico un acuerdo formalizado, una relación de contraprestación recíproca, una implicación personal y un marco temporal de ejecución.



	<p>Durante una clase de teoría o lección magistral, el profesor realiza una exposición verbal de los contenidos sobre la materia objeto de estudio, mediante la cual suministra a los alumnos información esencial y organizada procedente de diversas fuentes con unos objetivos específicos predefinidos (motivar al alumno, exponer los contenidos sobre un tema, explicar conocimientos, efectuar demostraciones teóricas, presentar experiencias, etc.) pudiendo utilizar para ello, además de la exposición oral, otros recursos didácticos (audiovisuales, documentos, etc).</p>
	<p>Este método de enseñanza se utiliza como complemento de la clase de teoría (lección magistral) y se basa en solicitar a los estudiantes que desarrollen las soluciones adecuadas o correctas mediante la ejercitación de rutinas, la aplicación de fórmulas o algoritmos, la aplicación de procedimientos de transformación de la información disponible y la interpretación de los resultados.</p> <p>La intención principal es la de aplicar lo ya aprendido para favorecer la comprensión tanto de la importancia como del contenido de un nuevo tema, afianzar conocimientos y estrategias y su aplicación en las situaciones prácticas que se planteen.</p>
	<p>Clases que se imparten en aula informática con acceso a los recursos computacionales que se necesitan para el desarrollo de la materia. El alumno trabaja individualmente o en grupos muy reducidos (2-3 estudiantes) en la implementación y aplicación de un algoritmo bajo la supervisión del profesor.</p>

	...
	...
	...

8. Recursos didácticos

RECURSOS DIDÁCTICOS	
BIBLIOGRAFÍA	<ul style="list-style-type: none"> • High Performance Cluster Computing. R. Buyya. Ed. Prentice Hall. 1999 • Cluster Computing White Paper. M. Baker, et al. 2001. • Using MPI, Portable Parallel Programming with the Message Passing Interface. W. Gropp, E. Lusk, A. Skjellum. Ed. MIT Press. 1999 • Message Passing Interface Forum http://www.mpi-forum.org/ • The High Performance Fortran Handbook. Scientific and Engineering Computation Series C.H. Koelbel et al. The MIT Press, 1994. • UPC: Distributed Shared Memory Programming. T. El-Ghazawi et al. Wiley Series on Parallel and Distributed Computing, Wiley Interscience. 2005 • Berkeley Unified Parallel C (UPC) Project. http://upc.lbl.gov • Parallel Programming in OpenMP. R. Chandra et al. Ed. Morgan Kaufmann, 2001.
RECURSOS WEB	Página web de la asignatura (http://)
	<p>Sitio Moodle de la asignatura (http://web3.fi.upm.es/AulaVirtual)</p> <p>Para cada tema se aportará material para guiar y facilitar el estudio que estará disponible en Moodle. Típicamente por cada tema habrá:</p> <ol style="list-style-type: none"> 1) Resumen de los conceptos básicos y objetivos del tema. 2) Hoja de problemas, de los cuales el profesor resolverá algunos durante las horas de clases y el alumno los restantes en sus horas de estudio. 3) Ejercicios de implementación (para resolver usando MATLAB), de cara a preparar las pruebas de laboratorio.
EQUIPAMIENTO	Laboratorio

	Aula XXXX
	Sala de trabajo en grupo

Cronograma de trabajo de la asignatura

Semana	Actividades en Aula	Actividades en Laboratorio	Trabajo Individual	Trabajo en Grupo	Actividades de Evaluación	Otros
Semanas 1	<ul style="list-style-type: none"> Introducción. Arquitecturas secuenciales. 	<ul style="list-style-type: none"> INTRODUCCIÓN (2h) 	<ul style="list-style-type: none"> 			<ul style="list-style-type: none">
Semanas 2 -- 3	<ul style="list-style-type: none"> Perfilado. Uso eficiente de la jerarquía de memoria. (6 horas) 	<ul style="list-style-type: none"> Clase LAB (2 h) Práctica (3h) 	<ul style="list-style-type: none"> 5 horas (conceptos/prob) 1 horas (ejercicios LAB para preparar el 	<ul style="list-style-type: none"> 2 horas (practica) 	<ul style="list-style-type: none"> 	<ul style="list-style-type: none">
Semanas 4 -- 7	<ul style="list-style-type: none"> Optimización en la CPU librerías optimizadas, Benchmarking(12 horas) 	<ul style="list-style-type: none"> Clase LAB (2 h) Práctica (3h) 	<ul style="list-style-type: none"> 10 horas (conceptos/prob) 1 horas (ejercicios LAB) 	<ul style="list-style-type: none"> 2 horas (practica) 	<ul style="list-style-type: none"> 	<ul style="list-style-type: none">
Semanas 8 -- 9	Paralelismo y arquitecturas paralelas (6 horas)	<ul style="list-style-type: none"> Clase LAB (2 h) Práctica (3h) 	<ul style="list-style-type: none"> 5 horas (conceptos/prob) 1 horas (ejercicios LAB) 	<ul style="list-style-type: none"> 2 horas (practica) 	<ul style="list-style-type: none"> Entrega proyecto 1 	<ul style="list-style-type: none">
Semanas 10 --12	<ul style="list-style-type: none"> Programación paralela openMp, HPF (8 horas) 	<ul style="list-style-type: none"> Clase LAB (2 h) Práctica (2h) 	<ul style="list-style-type: none"> 6 horas (conceptos/prob) 1 horas (ejercicios LAB) 	<ul style="list-style-type: none"> 2 horas (practica) 	<ul style="list-style-type: none"> 	<ul style="list-style-type: none">
Semanas 13--16	<ul style="list-style-type: none"> Programación paralela . UPC, MPI (12 horas) 	<ul style="list-style-type: none"> Clase LAB (3 h) Práctica (3h) 	<ul style="list-style-type: none"> 10 horas (conceptos/prob) 2 horas (ejercicios LAB) 	<ul style="list-style-type: none"> 2 horas (practica) 	<ul style="list-style-type: none"> 	<ul style="list-style-type: none">
Semana Exámenes	EXAMEN FINAL TEORIA Solo para alumnos que no superen los proyectos	EXAMEN FINAL LABORATORIO Solo para alumnos que no superen los proyectos	10 horas de preparación examen Se supone un seguimiento previo asignatura. El examen final de LABORATORIO no es obligatorio y puede aprobarse por curso.		<ul style="list-style-type: none"> Entrega proyecto 2 EXAMEN FINAL de PROBLEMAS y LABORATORIO (opcional) 	<ul style="list-style-type: none">

Nota: Para cada actividad se especifica la dedicación en horas que implica para el alumno.

COMENTARIOS al CRONOGRAMA ANTERIOR:

1. Las horas reservadas en nuestro horario para ACTIVIDADES en AULA suman 48, correspondientes a un horario de 3 h/semana x 16 semanas. En el cronograma anterior, las horas listadas suman 44 horas (dejando margen para fiestas, imponderables).
2. Las horas de LABORATORIO (incluyendo exámenes laboratorio) presupuestadas suman un máximo de 30 horas, ajustadas a un horario de 2horas/semana x 16 semanas = 32 horas. De nuevo, para tener en cuenta días no lectivos, etc. se cuenta con la flexibilidad de poder reducir a 4 el número de exámenes de Laboratorio/Prácticas a realizar durante el curso (de un total de 5 presupuestadas).

3. El computo de horas de trabajo del alumno fuera del aula :

• Familiarizarse con MATLAB	4 horas
• Repaso asignatura / conceptos / ejercicios	36 horas
• Repasar / completar ejercicios LABORATORIO	6 horas
• Completar practicas (trabajo en grupo)	10 horas
• Preparación Examen Laboratorio	10 horas
• Preparar examen final	10 horas
• Realización del Examen Final de Problemas	<u>+ 2 horas</u>
• HORAS TOTALES	78 horas

Estas 78 horas, sumadas a las 74 horas presenciales hacen un total de 152 horas de dedicación del alumno, equivalentes a los 6 créditos ECTS de la asignatura.

4. Estas 150 horas estarán repartidas uniformemente durante el semestre a razón de unas 10 horas semanales. Dado que en el horario de la asignatura hay 5 horas de clase presencial, el alumno debe contar con duplicar dichas horas con trabajo propio repasando conceptos, haciendo problemas, resolviendo ejercicios computacionales y de implementación, etc

