



Guía Docente				
Datos Identificativos			2012/13	
Asignatura (*)	Técnicas de Optimización e Paralelización	Código	614473005	
Titulación				
Descritores				
Ciclo	Período	Curso	Tipo	Créditos
Mestrado Oficial	1º cuatrimestre	Primeiro	Obrigatoria	6
Idioma	CastelánGalego			
Prerrequisitos				
Departamento	Electrónica e Sistemas			
Coordinación	Fraguela Rodriguez, Basilio Bernardo	Correo electrónico	basilio.fraguela@udc.es	
Profesorado	Arenaz Silva, Manuel Carlos	Correo electrónico	manuel.arenaz@udc.es	
	Fraguela Rodriguez, Basilio Bernardo		basilio.fraguela@udc.es	
Web				
Descrición xeral	El objetivo principal del curso es proporcionar una visión general de los esquemas básicos de optimización de código y paralelización más usados en aplicaciones de todo tipo, con especial énfasis en las científicas y de ingeniería. Se hace una revisión de los núcleos computacionales más comunes en este tipo de aplicaciones, y de sus versiones paralelas para cada tipo de algoritmo. Con las competencias adquiridas en esta materia, el egresado tendrá la capacidad de tomar decisiones profesionales y empresariales que permitan mejorar la calidad, el rendimiento y la competitividad de los productos de software de su organización.			

Competencias da titulación	
Código	Competencias da titulación

Resultados da aprendizaxe			
Competencias de materia (Resultados de aprendizaxe)	Competencias da titulación		
Aplicar técnicas de optimización para códigos secuenciales que tienen un alto impacto en programas paralelos	AI1 AI6	BI6	
Aplicar los algoritmos de transformación de programas secuenciales en programas paralelos más utilizados	AI1 AI6		
Extraer paralelismo mediante el análisis de las características de una aplicación secuencial con objeto de desarrollar una versión paralela	AI1 AI6	BI6	
Adaptar aplicaciones de forma que tengan un buen rendimiento en computadores paralelos con diferentes arquitecturas	AI1 AI6		
Comparar y evaluar alternativas de diseño e implementación de aplicaciones paralelas para computadores paralelos con diferentes arquitecturas	AI1 AI2 AI6	BI2 BI3 BI9	CM6
Trabajar en equipo		BI12 BI13	CM1
Llevar a cabo un aprendizaje autónomo		BI4	CM2 CM7

Contidos	
Temas	Subtemas
1. Introducción	a. Programa b. Relación con otras asignaturas c. Evaluación d. Consideraciones generales



2. Optimización do procesamento paralelo	<ul style="list-style-type: none"> a. Métricas de rendimento de programas paralelos b. Fases en a paralelización c. Técnicas de distribución/particionamento de datos d. Técnicas de balanceo de a carga computacional
3. Optimización do uso de a jerarquía de memoria	<ul style="list-style-type: none"> a. Técnicas de Programación para a Mejora de a Localidad b. Técnicas de Programación para a Ocultación de a Latencia de los Accesos c. Optimizacións do Uso de a Memoria Específicas de las Aplicacións Paralelas
4. Outras optimizacións secuenciais	<ul style="list-style-type: none"> a. Extensiones al lenguaje b. Condicionales c. Bucles d. Tablas de consulta e. Plantillas f. Inlining g. Outros
5. Optimización baseada en núcleos computacionais	<ul style="list-style-type: none"> a. Caracterización de núcleos computacionais b. Técnicas de paralelización de núcleos computacionais c. Acoplamiento de diversas técnicas de paralelización d. Outros aspectos do rendimento de las técnicas de paralelización de núcleos computacionais e. Paralelización automática

Planificación

Metodoloxías / probas	Horas presenciais	Horas non presenciais / traballo autónomo	Horas totais
Sesión maxistral	22	44	66
Prácticas de laboratorio	26	26	52
Proba obxectiva	2	2	4
Traballos tutelados	2	20	22
Atención personalizada	6	0	6

*Os datos que aparecen na táboa de planificación son de carácter orientativo, considerando a heteroxeneidade do alumnado

Metodoloxías

Metodoloxías	Descrición
Sesión maxistral	Exposición oral complementada con el uso de medios audiovisuales y la introducción de fases de debate con los estudiantes. Todo ello con la finalidad de transmitir conocimientos y facilitar el aprendizaje. Se realizarán sesiones magistrales sobre gran parte de los contenidos del temario, normalmente como punto de partida para el resto de actividades previstas para cada punto.
Prácticas de laboratorio	Actividad que permite a los estudiantes aprender y afianzar los conocimientos ya adquiridos mediante la realización de sesiones prácticas en ordenadores.
Proba obxectiva	Actividad realizada para la evaluación del conocimiento y las capacidades adquiridas por los alumnos con esta materia. Consiste en una prueba escrita con preguntas para la evaluación individual objetiva de cada alumno.
Traballos tutelados	Actividad que permite a los alumnos estudiar en mayor profundidad una aplicación específica directamente relacionada con alguno de los contenidos de la materia. Se le asignará un trabajo tutelado a cada alumno. Una vez desarrollado tendrá que entregar un informe sobre el mismo.

Atención personalizada

Metodoloxías	Descrición
--------------	------------



Prácticas de laboratorio Traballos tutelados	<p>La atención personalizada se desarrolla en la realización de las prácticas de laboratorio, así como en los trabajos tutelados, ya que es imprescindible para dirigir a los alumnos en el desarrollo del trabajo. Además, esta atención sirve para validar y evaluar el trabajo que va realizando cada alumno en distintas fases de su desarrollo hasta llegar a su finalización.</p> <p>Por otro lado, se recomendará a los alumnos la asistencia a tutorías como método de ayuda.</p>
---	---

Avaliación		
Metodoloxías	Descrición	Cualificación
Proba obxectiva	Examen de contenidos teóricos y prácticos	50
Prácticas de laboratorio	Valoración de la aptitud y aprendizaje desarrollados en las actividades acometidas en las sesiones prácticas de la materia	20
Traballos tutelados	Calidad del trabajo y del progreso del alumno durante su realización	30

Observacións avaliación

Fontes de información	
Bibliografía básica	<ul style="list-style-type: none">- M. Wolfe (1995). High-Performance Compilers for Parallel Computing. Addison Wesley- Randy Allen, Ken Kennedy (2001). Optimizing Compilers for Modern Architectures: A Dependence-based Approach . Morgan Kaufmann- Richard Gerber, Aart J. C. Bik , Kevin Smith , Xinmin Tian (2005). The software optimization cookbook. High performance recipes for IA-32 platforms. Intel Press 2nd edition
Bibliografía complementaria	<ul style="list-style-type: none">- M.J. Quinn (2003). Parallel Programming in C with MPI and OpenMP. McGraw-Hill

Recomendacións
Materias que se recomenda ter cursado previamente
Arquitectura e Tecnoloxía de Computadores/614473001 Programación Paralela/614473004
Materias que se recomenda cursar simultaneamente
Programación Paralela/614473004
Materias que continúan o temario
Taller de Proxectos/614473013 Traballo Fin de Mestrado/614473014 Depuración de Programas e Análise do Rendemento/614473008
Observacións

(*)A Guía docente é o documento onde se visualiza a proposta académica da UDC. Este documento é público e non se pode modificar, salvo casos excepcionais baixo a revisión do órgano competente dacordo coa normativa vixente que establece o proceso de elaboración de guías