

Guía docente de la asignatura

## Arquitectura de Computadores (496112A)



Fecha de aprobación: 28/06/2024

<b>Grado</b>	Grado en Ingeniería Informática (Ceuta)	<b>Rama</b>	Ingeniería y Arquitectura				
<b>Módulo</b>	Formación Específica de Rama	<b>Materia</b>	Estructura y Arquitectura de Computadores				
<b>Curso</b>	2º	<b>Semestre</b>	2º	<b>Créditos</b>	6	<b>Tipo</b>	Obligatoria

### PRERREQUISITOS Y/O RECOMENDACIONES

- Se recomienda la superación de los contenidos y adquisición de competencias de las materias de formación básica y de la asignatura de rama Estructura de Computadores.

### BREVE DESCRIPCIÓN DE CONTENIDOS (Según memoria de verificación del Grado)

- Estructura y clasificación de arquitecturas paralelas (procesadores, multiprocesadores, multicomputadores y sistemas distribuidos).
- Clasificación del paralelismo de una aplicación.
- Arquitecturas con paralelismo a nivel de instrucción (ILP).
- Programación eficiente de ILP (mecanismos y algoritmos básicos de optimización de código).
- Arquitecturas multihebra, multinúcleo y multiprocesadores.
- Programación paralela.
- Evaluación de prestaciones.

### COMPETENCIAS ASOCIADAS A MATERIA/ASIGNATURA

#### COMPETENCIAS GENERALES

- CG04 - Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.
- CG06 - Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes.

#### COMPETENCIAS ESPECÍFICAS



- CE14 - Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
- CE15 - Capacidad de conocer, comprender y evaluar la estructura y arquitectura de los computadores, así como los componentes básicos que los conforman.
- CE20 - Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.

### COMPETENCIAS TRANSVERSALES

- CT02 - Capacidad para tomar decisiones basadas en criterios objetivos (datos experimentales, científicos o de simulación disponibles) así como capacidad de argumentar y justificar lógicamente dichas decisiones, sabiendo aceptar otros puntos de vista.

### RESULTADOS DE APRENDIZAJE (Objetivos)

- Explicar las diferentes clasificaciones de arquitecturas paralelas.
- Distinguir entre procesamiento paralelo y procesamiento distribuido, y asociarlos con las arquitecturas que se utilizan para implementarlos.
- Relacionar el paralelismo implícito en una aplicación con las arquitecturas que lo aprovechan.
- Afrontar el análisis y el diseño de un núcleo con paralelismo a nivel de instrucción (ILP).
- Describir lo que hace un compilador y el programador para aprovechar una arquitectura ILP. Implementar código que aproveche la arquitectura ILP. Distinguir entre las prestaciones del procesador, las del compilador y las del programa que ejecute el computador.
- Explicar los conceptos de ganancia en prestaciones o velocidad y escalabilidad y las leyes relacionadas con estos conceptos.
- Describir la estructura y organización de arquitecturas multihebra, multinúcleo y multiprocesador.
- Explicar lo que hace un compilador para aprovechar una arquitectura multinúcleo y multiprocesador.
- Expresar un algoritmo de forma apropiada para que se pueda ejecutar en arquitecturas multinúcleos y multiprocesadores. Escribir código que aproveche la arquitectura multinúcleo y multiprocesador.
- Explicar la necesidad de mantener coherencia entre caches y entre cache y memoria principal. Afrontar el análisis y diseño de protocolos de mantenimiento de coherencia en multicores y multiprocesadores.
- Distinguir entre los diferentes tipos de modelos de consistencia de memoria. Explicar la influencia en las prestaciones de un computador del modelo de consistencia de memoria.
- Implementar código que aproveche el modelo de consistencia de memoria y las instrucciones máquina de sincronización. Implementar mecanismos básicos de sincronización.

### PROGRAMA DE CONTENIDOS TEÓRICOS Y PRÁCTICOS

#### TEÓRICO



- Arquitecturas paralelas: clasificación y prestaciones
- Programación paralela
- Arquitecturas con paralelismo a nivel de hebra (TLP)
- Arquitecturas con paralelismo a nivel de instrucción (ILP)
- Arquitecturas con paralelismo de datos (DLP)

## PRÁCTICO

- Programación Paralela con Memoria Compartida: distribución de trabajo, comunicación y sincronización entre hebras.
- Optimización de Código para microarquitecturas ILP
- Evaluación de Prestaciones

## BIBLIOGRAFÍA

### BIBLIOGRAFÍA FUNDAMENTAL

- J. Ortega, M. Anguita, A. Prieto. Arquitectura de Computadores. Thomson, 2005. ISSN/ISBN: 84-9732-274-6. ESIIT/C.1 ORT arq
- M. Anguita, J. Ortega. Fundamentos y Problemas de Arquitectura de Computadores, Editorial Técnica Avicam. 2016. ISSN/ISBN: 978-84-16535-52-1. ESIIT/C.1 ANG fun
- Barbara Chapman, Gabriele Jost, Ruud van der Pas, Using OpenMP. Portable Shared Memory Parallel Programming, The MIT Press. 2008. ISSN/ISBN: 9780262533027. ESIIT/D.1 CHA usi

### BIBLIOGRAFÍA COMPLEMENTARIA

- T. Rauber, G. Runder. [Parallel Programming: for Multicore and Cluster Systems](#). Springer 2023. ISSN/ISBN: 978-3-031-28923-1.
- Barry Wilkinson. Parallel programming: techniques and applications using networked workstations and parallel computer, 2005. ISSN/ISBN: 9780131405639. ESIIT/D.1 WIL par
- R. Gerber, A. J.C. Bik, K. B. Smith and X. Tian. The Software Optimization Cookbook. High Performance Recipes for the IA-32 Platforms. Intel Press, 2006. ESIIT/C.1 SOF sof
- A. Fog. [How to Optimize for the Pentium family of microprocessors](#), 2004.
- R. Gerber. The Software Optimization Cookbook. High Performance Recipes for the Intel Architecture. Intel Press, 2002. ESIIT/C.1 GER sof
- Ortega, J.; González, J Problemas de Ingeniería de Computadores. Cien problemas resueltos de procesadores paralelos Ed. Copicentro Granada, 2008. ESIIT/C.1 ORT pro

## ENLACES RECOMENDADOS

- [Procesadores de Intel](#)
- [Procesadores de AMD](#)
- [Gcc/g++](#)
- [OpenMp](#)



## METODOLOGÍA DOCENTE

- MD01 - Lección Magistral (Clases Teóricas-Expositivas)
- MD02 - Actividades Prácticas (Resolución de Problemas, Resolución de Casos Prácticos, Desarrollo de Proyectos, Prácticas en Laboratorio, Taller de Programación, Aula de Informática, Prácticas de Campo).
- MD03 - Seminarios (Debates, Demos, Exposición de Trabajos Tutelados, Conferencias, Visitas Guiadas, Monografías).
- MD04 - Actividades no presenciales Individuales.
- MD05 - Actividades no presenciales Grupales.
- MD06 - Tutorías Académicas.

## EVALUACIÓN (instrumentos de evaluación, criterios de evaluación y porcentaje sobre la calificación final)

### EVALUACIÓN ORDINARIA

Todo lo relativo a la evaluación se regirá por la Normativa de evaluación y calificación de los estudiantes vigente en la Universidad de Granada.

La calificación final que aparecerá en el Acta será un número comprendido entre 0 y 10 con precisión de un decimal (de acuerdo con el art. 5 del R. D 1125/2003). Para aprobar oficialmente una asignatura se ha de obtener una puntuación mínima de 5 (de acuerdo con el art. 5 del R. D 1125/2003).

En función de la convocatoria (ordinaria o extraordinaria), y del tipo de evaluación escogida, la calificación se obtendrá como se detalla a continuación.

El número de estudiantes tan elevado, a pesar de ser una asignatura de ingeniería y tanto por grupo de teoría como por grupo prácticas, condiciona los métodos y herramientas de evaluación aplicables.

La metodología de evaluación por defecto según la normativa de la Universidad de Granada es la evaluación continua, que en el caso de esta asignatura se compone de las siguientes actividades:

Porcentajes de evaluación

Actividades formativas evaluación ordinaria		Ponderación	Mínimo	Máximo
Actividades de teoría	Actividades	60%	2,4	3
	Prueba final			3
Actividades de prácticas y seminarios	Actividades	40%	1,6	2
	Prueba final			2
Calificación en Acta		100%	5	10

Obsérvese que tener un mínimo supone que es obligatorio hacer la prueba o actividad para poder aprobar la asignatura. En la prueba final de evaluación continua se evaluarán todos los contenidos de teoría y prácticas. Esta prueba tendrá lugar al final del cuatrimestre en la fecha que fije el centro. La calificación en Acta, para quien, no habiendo superando algunos de los mínimos de teoría o prácticas, alcance una puntuación superior a 5 al sumar la nota de teoría y prácticas, será igual a 4.9 (suspenseo).

### EVALUACIÓN EXTRAORDINARIA

Para las convocatorias extraordinarias se utilizará la evaluación única final.



**EVALUACIÓN ÚNICA FINAL**

La siguiente tabla muestra las actividades y la contribución de las mismas a la nota final de la asignatura y la nota mínima exigida, en su caso, para cada una de ellas:

Porcentajes de evaluación

Prueba única final evaluación extraordinaria	Ponderación	Mínimo	Máximo
Prueba escrita de la parte teórica	60%	2,4	6
Prueba escrita de la parte de prácticas y seminarios	40%	1,6	4
<b>Calificación en Acta</b>	100%	5	10

Obsérvese que tener un mínimo supone que es obligatorio hacer la prueba o actividad para poder aprobar la asignatura. Esta prueba tendrá lugar al final del cuatrimestre en la fecha que fije el centro. La calificación en Acta, para quien, no habiendo superando algunos de los mínimos de teoría o prácticas, alcance una puntuación superior a 5 al sumar la nota de teoría y prácticas, será igual a 4.9 (suspenso).

**INFORMACIÓN ADICIONAL**

Información de interés para estudiantado con discapacidad y/o Necesidades Específicas de Apoyo Educativo (NEAE): [Gestión de servicios y apoyos \(https://ve.ugr.es/servicios/atencion-social/estudiantes-con-discapacidad\)](https://ve.ugr.es/servicios/atencion-social/estudiantes-con-discapacidad).

