

## Programa de la asignatura

### *“Arquitectura e Ingeniería de Computadores”*

#### 1. Identificación y características de la asignatura

Denominación	Arquitectura e Ingeniería de Computadores			
Curso y Titulación	4º Curso. Ingeniería en Informática (66 ctos. LRU)			
Área de conocimiento	Arquitectura y Tecnología de Computadores			
Departamento	Tecnología de los Computadores y de las Comunicaciones			
Carácter	Troncal (6 ctos. LRU teoría + 3 ctos. LRU prácticas)			
Duración ECTS (créditos)	Anual		8,2 ECTS (205 horas)	
Distribución ECTS (rangos)	Grupo Grande: 30%	Seminario-Lab.: 15%	Tutoría ECTS: 0%	No presenciales: 55%
	61-62 horas	30-31 horas	0 horas	112-113 horas
Descriptores (según BOE)	Arquitecturas paralelas. Arquitecturas orientadas a aplicaciones y lenguajes			
Coordinador-Profesor/es	Miguel Ángel Vega Rodríguez, Juan Manuel Sánchez Pérez, Julio Ballesteros Rubio			
Tutorías complementarias (Miguel A. Vega Rodríguez)	Despacho ARCO, planta baja, edificio de Investigación		Telf.: 927-257-263	Correo-e: <a href="mailto:mavega@unex.es">mavega@unex.es</a>
	• Las tutorías se publicarán en la web del Centro y en la puerta del despacho del profesor en los plazos previstos por la Normativa vigente de Tutorías.			
Tutorías complementarias (Juan M. Sánchez Pérez)	Despacho 6, planta baja, pabellón de Informática		Telf.: 927-257-246	Correo-e: <a href="mailto:sanperez@unex.es">sanperez@unex.es</a>
	• Las tutorías se publicarán en la web del Centro y en la puerta del despacho del profesor en los plazos previstos por la Normativa vigente de Tutorías.			
Tutorías complementarias (Julio Ballesteros Rubio)	Despacho 23, planta baja, pabellón de Informática		Telf.: 927-257-247	Correo-e: <a href="mailto:julioba@unex.es">julioba@unex.es</a>
	• Las tutorías se publicarán en la web del Centro y en la puerta del despacho del profesor en los plazos previstos por la Normativa vigente de Tutorías.			

## 2. Objetivos/competencias de la asignatura

### Objetivo principal:

- Formar al alumno en los fundamentos arquitectónicos de los computadores actuales.

<i>Relacionados con competencias académicas y disciplinares</i>	
Descripción	
1.	Conocer y comprender el concepto de modelo computacional y los distintos modelos computacionales existentes, profundizando en el modelo de von Neumann.
2.	Dominar los fundamentos de la Arquitectura e Ingeniería de Computadores, comprendiendo la estrecha interrelación entre arquitectura de computadores y tecnología de computadores.
3.	Ser capaz de analizar el rendimiento de distintas arquitecturas de computadores.
4.	Comprender y conocer el concepto de paralelismo a distintos niveles, la clasificación de las arquitecturas paralelas existentes, las técnicas paralelas básicas, y la relación entre lenguajes y arquitecturas paralelas.
5.	Manejar en profundidad las técnicas de segmentación usadas en procesadores para mejora de su rendimiento.
6.	Comprender el concepto de procesador vectorial y su modo de operación, siendo capaz de comparar el rendimiento de estos procesadores con el de los escalares.
7.	Conocer y dominar las características de los sistemas multiprocesadores actuales. Dentro de estas características son de especial interés las redes de interconexión, los mecanismos de sincronización y la organización de la memoria en multiprocesadores.
8.	Comprender y dominar el concepto de multicomputador, entendiendo las distintas redes de interconexión, los mecanismos de paso de mensajes y las estrategias de encaminamiento.

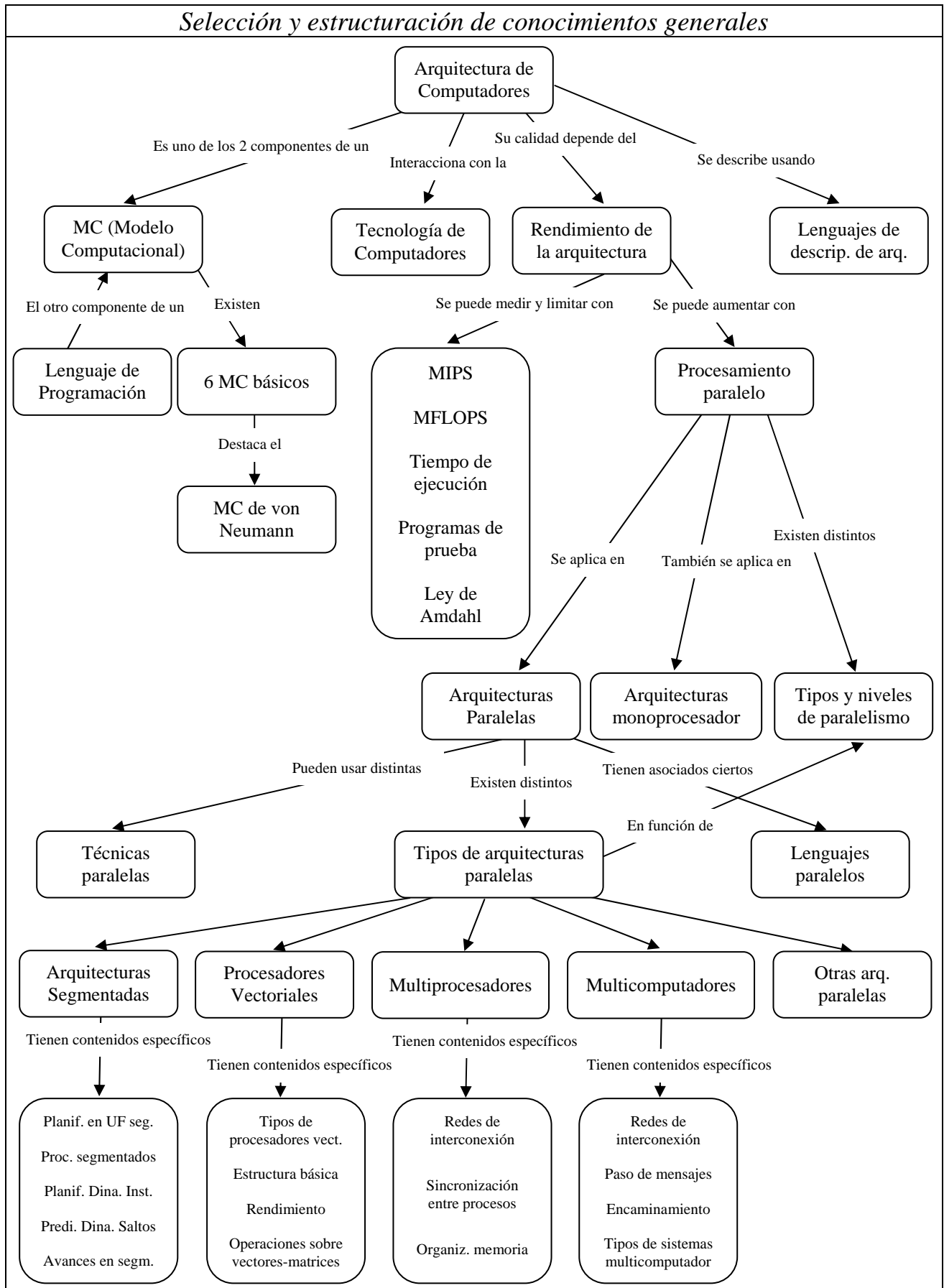
<i>Relacionados con otras competencias personales y profesionales</i>		Competencias Transversales Genéricas (CG)
Descripción		
9.	Ser capaz de desarrollar proyectos, relativos a Arquitectura de Computadores, de dimensiones importantes.	2, 3, 5, 7, 8, 9, 11, 12, 13, 14, 15
10.	Resolver problemas, de Arquitectura de Computadores, con creatividad y confianza en los propios conocimientos.	1, 3, 5, 7, 11, 13, 15
11.	Desarrollar habilidades de síntesis y análisis de la información, combinación de información de diversas fuentes (no sólo de fuentes en español, sino también en lengua inglesa), y gestión de un gran volumen de información. En particular, para información sobre Arquitectura de Computadores.	1, 3, 4, 5, 6, 10, 11, 15

La siguiente tabla explica con más detalle las competencias transversales genéricas -CG- (propias de muchas titulaciones) que se intentan potenciar en los alumnos que cursan la asignatura “Arquitectura e Ingeniería de Computadores” (AIC). Además de estas competencias, en esta asignatura también se potencian otras muchas competencias específicas o genéricas de la titulación.

Competencias Transversales Genéricas (CG)	
Competencia	Breve explicación
<b>INSTRUMENTALES</b>	
1. Capacidad de análisis y síntesis	Fundamental para el desarrollo de los problemas prácticos propuestos en la asignatura, e incluso para el estudio de la misma, puesto que su temario es amplio.
2. Capacidad de organización y planificación	Competencia importante para la realización de las prácticas de AIC, puesto que éstas proponen al alumno el desarrollo de diversos proyectos de dimensión importante.
3. Comunicación oral y escrita en la lengua nativa	Capacidad importante para la realización de los exámenes, y para la redacción de la documentación técnica asociada a los proyectos propuestos en las prácticas de AIC.

4. Conocimiento de una lengua extranjera	AIC es una asignatura avanzada con un marcado carácter tecnológico, por lo que muchos de los mejores libros de texto únicamente se encuentran en lengua inglesa. Aunque al alumno se le ofrece en español todo el material necesario para superar la asignatura, lo cierto es que en alguna ocasión quizás le sea necesario acudir a estos textos en lengua inglesa.
5. Conocimientos de Informática relativos al ámbito de estudio	La relación con AIC es directa.
6. Capacidad de gestión de la información (captación y análisis de la información)	Competencia necesaria para sacar el máximo partido de la teoría, el estudio del temario, etc.
7. Resolución de problemas	El examen de la asignatura siempre incluye diversos problemas prácticos a resolver. Otros problemas aparecen durante el desarrollo de los proyectos propuestos en las prácticas.
8. Toma de decisiones	Las prácticas de AIC están abiertas a optimizaciones/mejoras propuestas por el propio alumno. El alumno decide si presenta o no optimizaciones y cuáles. Más aún, las herramientas usadas dentro de las prácticas (p.e. Visual C++ .NET) son muy amplias, con múltiples alternativas de uso. El alumno debe decidir qué alternativas prefiere usar.
<b>PERSONALES</b>	
9. Trabajo en equipo	Las prácticas de la asignatura AIC se realizan en equipos de 2 alumnos.
<b>SISTEMÁTICAS</b>	
10. Razonamiento crítico	El examen de la asignatura siempre incluye alguna pregunta teórica que el alumno debe razonar. Esta competencia también es útil durante el estudio de la asignatura, sobre todo si se complementa el temario con otras fuentes de información.
11. Aprendizaje autónomo	Las prácticas de AIC proponen al alumno el desarrollo de proyectos en diversos lenguajes de programación (como VHDL o Visual C++ .NET). El alumno no tiene por qué conocer esos lenguajes, por lo que debe aprenderlos en parte de forma autónoma, aunque los profesores de la asignatura también proporcionen cierta ayuda. Por otro lado, la asignatura se complementa con el uso de varios simuladores, que permiten que los alumnos practiquen en casa (a su propio ritmo) los conceptos enseñados en clase.
12. Adaptación a nuevas situaciones	Como hemos comentado, las prácticas de AIC incluyen el desarrollo de proyectos en diversos lenguajes de programación (como VHDL o Visual C++ .NET). Y el alumno no tiene por qué conocer, de partida, esos lenguajes. Por tanto, se enfrenta a nuevas situaciones.
13. Creatividad	Durante las prácticas de AIC, especialmente las del 2º semestre, los alumnos deben desarrollar una aplicación visual en entorno Windows. La creatividad es una competencia importante para mejorar la interfaz de dicha aplicación. También es importante la creatividad a la hora de enfrentarse a muchos problemas prácticos.
14. Iniciativa y espíritu emprendedor	Fundamental para proponer mejoras/optimizaciones en las prácticas de la asignatura.
15. Motivación por la calidad	Las prácticas (tanto la aplicación desarrollada como la documentación técnica entregada) de la asignatura deben cumplir unos mínimos de calidad para ser aceptadas. En la misma línea se puede hablar de los exámenes de la asignatura, y en particular, de los problemas prácticos a resolver en ellos.

### 3. Contenidos



<i>Secuenciación de bloques temáticos y temas (programa teórico)</i>
1. Fundamentos de la Arquitectura e Ingeniería de Computadores
<p>1.1.- Concepto de modelo computacional y modelos computacionales básicos. El modelo de von Neumann y sus extensiones.</p> <p>1.2.- Concepto de Arquitectura de Computadores: Evolución e interpretación del concepto, interacción entre arquitectura y tecnología, lenguajes de descripción de arquitecturas.</p> <p>1.3.- Análisis del rendimiento de arquitecturas: Definiciones, tiempo de ejecución, MIPS y MFLOPS, programas de prueba, limitaciones del rendimiento.</p>
2. Introducción al procesamiento paralelo
<p>2.1.- Conceptos básicos.</p> <p>2.2.- Tipos y niveles de paralelismo.</p> <p>2.3.- Mecanismos de paralelismo en monoprocesadores.</p> <p>2.4.- Clasificación de las arquitecturas paralelas.</p> <p>2.5.- Técnicas paralelas básicas.</p> <p>2.6.- Relación entre lenguajes y arquitecturas paralelas.</p>
3. Segmentación
<p>3.1.- Introducción a la segmentación.</p> <p>3.2.- Planificación en unidades funcionales segmentadas.</p> <p>3.3.- Introducción a los procesadores segmentados.</p> <p>3.4.- Procesadores segmentados lineales.</p> <p>3.5.- Procesadores segmentados con unidades funcionales multiciclo.</p> <p>3.6.- Planificación dinámica de instrucciones.</p> <p>3.7.- Predicción dinámica de saltos.</p> <p>3.8.- Avances en arquitecturas segmentadas.</p>
4. Procesadores vectoriales
<p>4.1.- Tipos de procesadores vectoriales.</p> <p>4.2.- Estructura básica de los procesadores vectoriales.</p> <p>4.3.- Modelo de rendimiento: escalar y vectorial.</p> <p>4.4.- Ampliación del repertorio de instrucciones. Operaciones sobre vectores y matrices.</p>
5. Multiprocesadores
<p>5.1.- Introducción.</p> <p>5.2.- Redes de interconexión en multiprocesadores.</p> <p>5.3.- Mecanismos de sincronización entre procesos.</p> <p>5.4.- Organización de la memoria en los multiprocesadores.</p>
6. Multicomputadores
<p>6.1.- Introducción.</p> <p>6.2.- Redes de interconexión estáticas.</p> <p>6.3.- Mecanismos de paso de mensajes.</p> <p>6.4.- Estrategias de encaminamiento de mensajes.</p> <p>6.5.- Sistemas multicomputador.</p>

<i>Contenido práctico</i>
Fuertemente ligado al programa teórico explicado en cada semestre, con el objetivo de reforzar dicho programa mediante la experimentación.
PRIMER SEMESTRE
<ul style="list-style-type: none"><li>• Modelado y simulación de un procesador segmentado sin detección de riesgos ni anticipación.</li><li>• Modelado y simulación de un procesador segmentado con detección de riesgos y anticipación.</li></ul>
SEGUNDO SEMESTRE
<ul style="list-style-type: none"><li>• Modelado y simulación del problema de la coherencia caché en un multiprocesador simétrico (SMP) con un sistema jerárquico de memoria basado en bus.</li></ul>

#### 4. Metodologías/Actividades

<i>Actividad formativa</i>	<i>Metodología de enseñanza/aprendizaje (e/a)</i>
Coordinación y evaluación	Metodología e/a: <i>Presentación del plan docente con la clarificación de los objetivos a cumplir con cada actividad. Organización de la participación de los alumnos en las diversas tareas. Enumeración y repaso breve de los conocimientos. Pruebas individuales y exámenes escritos.</i>
Exposición verbal y aprendizaje a partir de documentos	Metodología e/a: <i>Exposición magistral de los contenidos teóricos de la asignatura y realización de ejercicios tipo relacionados con los mismos por parte del profesor (ambos únicamente mientras la asignatura tenga docencia presencial). Algunos contenidos en el material/documentación de apoyo podrían estar en inglés.</i>
Discusión	Metodología e/a: <i>Supervisión (y moderación) del debate sobre determinados conceptos de la asignatura, ayudando a los estudiantes a analizar y a extraer conclusiones.</i>
Solución de problemas	Metodología e/a: <i>Resolución de problemas por parte de los alumnos, así como corrección y discusión de problemas realizados por los alumnos previamente con la supervisión del profesor.</i>
Aplicación práctica y observación dirigida	Metodología e/a: <i>Realización por parte del alumno de simulaciones y montajes experimentales, así como aplicaciones software. Obtención de datos, interpretación de los mismos así como, en algunos casos, la elaboración de un informe.</i>
Diseño de proyectos o trabajos de investigación	Metodología e/a: <i>Recogida, análisis y síntesis de información. Resolución de problemas prácticos o ejecución de procedimientos. Implementación de proyectos de programación. Elaboración final de un informe a defender. Presentación oral de dicho informe. Trabajo en grupo.</i>
Actividades no presenciales (aprendizaje autónomo)	Metodología e/a: <i>Estudio, comprensión y desarrollo de los contenidos teóricos. Resolución de problemas prácticos planteados, para la posterior discusión y/o corrección. Desarrollo de proyectos o trabajos de investigación.</i>

## 5. Requisitos/Recomendaciones

<i>Interrelación</i>		
Requisitos	Temas	<i>Procedencia</i>
Los alumnos deben poseer amplios conocimientos sobre Estructura de Computadores para la correcta comprensión de cualquiera de los temas de la asignatura AIC. Estos conceptos fueron explicados en la asignatura “Estructura de Computadores”, que deberían tener aprobada	Para todos los temas	Estructura de Computadores (2º de Ingeniería en Informática)

<i>Recomendaciones</i>
Además, son deseables conocimientos de nivel avanzado sobre la materia programación (según fueron explicados en los cursos anteriores de la titulación).



## 6. Evaluación

La asignatura está dividida en dos partes: teoría y prácticas; que deben aprobarse por separado.

### TEORÍA

- Se realiza un examen final (junio).
- El alumno que no supere en junio la asignatura se examinará de toda la asignatura en la convocatoria de septiembre.

### PRÁCTICAS

- Se realizan diversas prácticas a lo largo de la asignatura.
- Las prácticas tienen carácter obligatorio.
- El alumno debe superar una prueba relacionada con las prácticas realizadas en cada semestre.
- Para aprobar la asignatura es indispensable tener aprobadas las prácticas.
- Si la nota de prácticas es igual o superior al notable servirá para redondear hacia arriba la nota alcanzada en el examen de teoría.

<i>Criterios de evaluación</i>	<i>Vinculación</i>
Descripción	Objetivo
1. Demostrar la adquisición, comprensión y dominio de los principales conceptos de la asignatura.	1-8, 11
2. Resolver problemas aplicando conocimientos teóricos y basándose en resultados experimentales.	1-3, 5-8, 10
3. Desarrollar y defender adecuadamente un proyecto relativo al temario del primer semestre.	1, 2, 3, 4, 5, 9
4. Desarrollar y defender adecuadamente un proyecto relativo al temario del segundo semestre.	2, 3, 4, 6, 7, 8, 9

<i>Actividades e instrumentos de evaluación</i>	
Seminario-Laboratorio (Prácticas)	<ul style="list-style-type: none"> <li>• Para cada proyecto se evaluará la calidad de la documentación técnica entregada (memoria del proyecto junto con la aplicación en sí).</li> <li>• Además, para cada proyecto el alumno también debe realizar una prueba de ejecución (defensa del proyecto ante ciertas situaciones prácticas delante del ordenador), demostrando el conocimiento y dominio del proyecto desarrollado.</li> <li>• Será necesario tener aprobados ambos proyectos (las prácticas) para aprobar la asignatura.</li> </ul>
Grupo grande (Teoría)	<ul style="list-style-type: none"> <li>• Prueba de desarrollo escrito y resolución de problemas. Para cada semestre incluirá 1 pregunta teórica y 3 problemas.</li> <li>• La pregunta teórica permite valorar la adquisición, comprensión y dominio de los conceptos. Los problemas evalúan la correcta aplicación de estos conceptos, de forma creativa y segura, a supuestos de carácter práctico.</li> </ul>

## 7. Bibliografía

<i>Bibliografía de apoyo seleccionada</i>
<p>[CUL99] Culler-Singh. “<i>Parallel Computer Architecture</i>”. Morgan Kaufmann, 1999.</p> <p>[HEN12] Hennessy-Patterson. “<i>Computer Architecture: A Quantitative Approach</i>”. Morgan Kaufmann, 5ª edición, 2012.</p> <p>[HWA93] Hwang. “<i>Advanced Computer Architecture: Parallelism, Scalability, Programmability</i>”. McGraw-Hill, 1993.</p> <p>[SIM98] Sima-Fountain-Kacsuk. “<i>Advanced Computer Architecture: A Design Space Approach</i>”. Addison-Wesley, 1998.</p> <p>[ZAR96] Zargham. “<i>Computer Architecture: Single and Parallel Systems</i>”. Prentice Hall, 1996.</p>
<i>Bibliografía o documentación de lectura obligatoria</i>
<ul style="list-style-type: none"> <li>• Resumen de cada uno de los temas (elaborado por los profesores).</li> <li>• Relación de problemas de cada tema (elaborada por los profesores).</li> <li>• Enunciados de prácticas donde se describen los distintos proyectos de Seminario-Laboratorio a realizar (elaborados por los profesores).</li> </ul>
<i>Recursos didácticos</i>
<ul style="list-style-type: none"> <li>• Aula virtual de la asignatura (descarga de materiales, programa de la asignatura, foros, noticias, etc.): <a href="http://campusvirtual.unex.es/zonaux/avux/course/view.php?id=5043">http://campusvirtual.unex.es/zonaux/avux/course/view.php?id=5043</a></li> <li>• Sitio web del simulador PipeSim (útil para el “Tema 3. Segmentación”): <a href="http://arco.unex.es/pipesim">http://arco.unex.es/pipesim</a></li> <li>• Sitio web del simulador PDIWeb (útil para el “Tema 3. Segmentación”): <a href="http://arco.unex.es/pdiweb">http://arco.unex.es/pdiweb</a></li> <li>• Sitio web del simulador SMPCache (útil para el “Tema 5. Multiprocesadores” y para el proyecto de Seminario-Laboratorio del 2º semestre): <a href="http://arco.unex.es/smpcache">http://arco.unex.es/smpcache</a></li> </ul>
<i>Bibliografía o documentación de ampliación, sitios web...</i>
<ul style="list-style-type: none"> <li>• Relación de problemas de examen (elaborada por los profesores).</li> <li>• [HOC92] Hockney-Jesshope. “<i>Parallel Computers 2: Architecture, Programming and Algorithms</i>”. Adam Hilger, 2ª edición, 1992.</li> <li>• [ORT05] Ortega-Anguila-Prieto. “<i>Arquitectura de Computadores</i>”. Thomson-Paraninfo, 2005.</li> <li>• [STA12] Stallings. “<i>Computer Organization &amp; Architecture: Designing for Performance</i>”. Prentice-Hall, 9ª edición, 2012.</li> <li>• <i>WWW Computer Architecture Page</i>. University of Wisconsin-Madison, University of Pennsylvania, University of Texas at Austin, EE.UU.: <a href="http://arch-www.cs.wisc.edu">http://arch-www.cs.wisc.edu</a></li> <li>• <i>TOP500 Supercomputer Sites</i>. University of Mannheim, University of Tennessee, NERSC/LBNL, Alemania y EE.UU.: <a href="http://www.top500.org">http://www.top500.org</a></li> <li>• <i>The Green500 Energy-Efficient Supercomputer List</i>: <a href="http://www.green500.org">http://www.green500.org</a></li> <li>• Sitios web de los distintos libros recomendados en la asignatura.</li> </ul>

## 8. Otras consideraciones metodológicas

<i>Horas de estudio y realización de prácticas recomendadas</i>
<ul style="list-style-type: none"><li>• 112-113 horas no presenciales.</li></ul>
<i>Normas</i>
<ul style="list-style-type: none"><li>• El alumno debe incluir su fotografía y datos relevantes en el campus virtual de la UEx.</li><li>• La convocatoria del examen de teoría la fija el Centro. El profesor fija la hora del examen.</li></ul>