

## PLAN DOCENTE DE LA ASIGNATURA

Curso académico: 2025/2026

Identificación y características de la asignatura					
Código	501279				
Denominación (español)	<b>Administración y Organización de Computadores</b>				
Denominación (inglés)	Management and Organization of Computers				
Titulaciones	Grado en Ingeniería Informática en Ingeniería de Computadores Grado en Ingeniería Informática en Ingeniería del Software				
Centro	Escuela Politécnica				
Módulo	Común a la Rama de Informática				
Materia	Ingeniería de Computadores				
Carácter	Obligatorio	ECTS	6	Semestre	3
Profesorado					
Nombre		Despacho		Correo-e	
Pilar Bachiller Burgos		Laboratorio de Robótica		pilarb@unex.es	
Francisco M. Andrés Hernández		Laboratorio de Robótica		pacoan@unex.es	
José María Granado Criado		Dirección TC2		granado@unex.es	
Área de conocimiento	Arquitectura y Tecnología de Computadores				
Departamento	Tecnología de los Computadores y de las Comunicaciones				
Profesor/a coordinador/a (si hay más de uno)	Pilar Bachiller Burgos				
Competencias					
<p><b>Competencias básicas</b></p> <p>CB1: Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.</p> <p>CB2: Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.</p> <p>CB3: Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.</p> <p>CB4: Que los estudiantes puedan transmitir información, ideas, problemas y</p>					

soluciones a un público tanto especializado como no especializado.

CB5: Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

### **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, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 del anexo II de la resolución de la Secretaría General de Universidades de 8 de junio de 2009 (BOE de 4 de agosto de 2009) para la tecnología específica de Ingeniería del Software e Ingeniería de Computadores.

CG06: Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 del anexo II de la resolución antes mencionada para la tecnología específica de Ingeniería del Software e Ingeniería de Computadores.

### **Competencias específicas**

CI04: Capacidad para elaborar el pliego de condiciones técnicas de una instalación informática que cumpla los estándares y normativas vigentes.

CI05: Conocimiento, administración y mantenimiento de sistemas, servicios y aplicaciones informáticas.

CI06: Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.

### **Competencias transversales**

CT02: Habilidades de gestión de recursos de información.

CT03: Capacidad para resolver problemas.

## **Contenidos**

Descripción general del contenido: Organización de un computador desde el punto de vista del programador del lenguaje máquina y ensamblador. Arquitecturas del Conjunto de Instrucciones (ISA), formatos de instrucción y modos de direccionamiento. Administración de sistemas operativos.

### **Temario**

**Denominación del tema 1:** El lenguaje de programación C

**Contenidos del tema 1:**

- 1.1. Introducción
- 1.2. Tipos de datos simples
- 1.3. Estructuras de control
- 1.4. Funciones
- 1.5. Arrays
- 1.6. Estructuras de datos heterogéneas

**Descripción de las actividades prácticas del tema 1:** sesiones de introducción al sistema operativo Linux sobre el que se desarrollan las prácticas de la asignatura.

**Denominación del tema 2:** Familia de microprocesadores de Intel

**Contenidos del tema 2:**

- 2.1. Arquitectura CISC versus RISC
- 2.2. Generaciones de microprocesadores de Intel
- 2.3. Modos de operación
- 2.4. Registros básicos
- 2.5. Modelos de memoria
- 2.6. Memoria virtual
- 2.7. Intel versus AMD

**Descripción de las actividades prácticas del tema 2:** sesiones prácticas de comandos en Linux.

**Denominación del tema 3:** Programación en bajo nivel de los microprocesadores de Intel

**Contenidos del tema 3:**

- 3.1. Lenguaje ensamblador
- 3.2. Modos de direccionamiento
- 3.3. Repertorio de instrucciones

**Descripción de las actividades prácticas del tema 3:** programación en bajo nivel y herramientas de generación de código.

**Denominación del tema 4:** Representación de programas a nivel de máquina

**Contenidos del tema 4:**

- 4.1. Representación de datos
- 4.2. Estructuras de control
- 4.3. Llamadas a procedimientos. Paso de parámetros
- 4.4. Gestión del bloque de activación
- 4.5. Ensamblador en línea

**Descripción de las actividades prácticas del tema 4:** ejercicios prácticos de representación de programas a nivel de máquina.

**Denominación del tema 5:** Aspectos avanzados de los microprocesadores de Intel

**Contenidos del tema 5:**

- 5.1 FPU

- 5.2 Tecnología MMX
- 5.3 Tecnología SSE
- 5.4 Tecnología AVX
- 5.5 Autovectorización

**Descripción de las actividades prácticas del tema 5:** desarrollo del proyecto final de prácticas que incluye aspectos avanzados de programación de bajo nivel.

### Actividades formativas

Horas de trabajo del alumno/a por tema		Horas Gran grupo	Actividades prácticas				Actividad de seguimiento	No presencial
			GG	CH	L	O		
Tema	Total	GG	CH	L	O	S	TP	EP
1	18,5	5		1,5			0	12
2	15	3		1,5			0	10,5
3	27,5	6		1,5			0	20
4	40,5	9		6,5			1	24
5	40,5	12		6			0,5	22
<b>Evaluación</b>	8	2,5		5,5				
<b>TOTAL</b>	150	37,5		22,5			1,5	88,5

GG: Grupo Grande (85 estudiantes).

CH: Actividades de prácticas clínicas hospitalarias (7 estudiantes)

L: Actividades de laboratorio o prácticas de campo (15 estudiantes)

O: Actividades en sala de ordenadores o laboratorio de idiomas (20 estudiantes)

S: Actividades de seminario o de problemas en clase (40 estudiantes).

TP: Tutorías Programadas (seguimiento docente, tipo tutorías ECTS).

EP: Estudio personal, trabajos individuales o en grupo, y lectura de bibliografía.

### Metodologías docentes

- En Clases teórico-prácticas en el aula, se realizarán clases expositivas para el desarrollo de los contenidos teóricos. Asimismo, se realizarán clases de explicación y resolución de ejercicios y problemas. Por último, se llevarán a cabo actividades individuales o en grupo destinadas a aplicar los conceptos expuestos a la resolución de problemas.
- En sesiones de laboratorio, se dedicarán varias sesiones prácticas a conocer el sistema operativo y las herramientas software que se utilizarán a lo largo de la asignatura. Se propondrán problemas de programación que habrá que resolver durante la sesión. Se planteará una práctica de programación que se desarrollará tanto dentro como fuera del laboratorio, realizando actividades de seguimiento durante las sesiones prácticas correspondientes.
- En tutorías programadas, individuales o en grupos pequeños se realizará el seguimiento de las actividades planteadas a lo largo del semestre y la evaluación individual de los objetivos alcanzados.
- Realización de actividades, trabajos y estudio por parte del estudiante, de manera autónoma. Las actividades no presenciales necesarias para alcanzar los objetivos de aprendizaje se resumen en las siguientes: estudio individual, búsqueda de información, desarrollo de programas, resolución de problemas, realización de cuestionarios.

### Resultados de aprendizaje

- Comprende la organización de un computador desde el punto de vista del programador en lenguaje máquina y ensamblador, conociendo las distintas

alternativas para el conjunto de instrucciones, los formatos de instrucción y modos de direccionamiento.

- Conoce y aplica en actividades de nivel medio las competencias transversales fundamentales de la profesión.

## Sistemas de evaluación

### Instrumentos de evaluación

Se utilizarán los siguientes instrumentos de evaluación:

- ✓ Portafolio de actividades: conjunto de actividades realizadas por el estudiante a lo largo del semestre. Se realizarán varias actividades de este tipo a lo largo del curso.
- ✓ Prácticas de programación: resolución de problemas de administración de sistemas operativos y de programación de bajo nivel aplicando los distintos conocimientos adquiridos a lo largo de la asignatura.
- ✓ Pruebas escritas: preguntas de tipo test, preguntas cortas, resolución de problemas, etc.

### Criterios de evaluación

Tal y como se contempla en la normativa de evaluación vigente, esta asignatura puede superarse siguiendo el sistema de evaluación continua o mediante el sistema de evaluación global.

Como se indica en esta normativa, la elección del sistema de evaluación global corresponde a los estudiantes. Así aquellos estudiantes que deseen acogerse a este sistema, deberán solicitarlo a los profesores de la asignatura en el plazo establecido según la normativa. Si un estudiante no solicita expresamente el sistema de evaluación global en el plazo indicado, se supondrá que opta por la evaluación continua.

### **Evaluación continua**

Para superar la asignatura por el sistema de evaluación continua será necesario obtener una calificación superior o igual a 5 (sobre 10) en las prácticas de programación y en la prueba de evaluación del temario teórico que se realizará en las convocatorias oficiales. La primera parte de dicha prueba podrá convalidarse con el resultado de un examen parcial que tendrá lugar durante la primera mitad del semestre, siempre que dicho resultado supere el aprobado. Esta opción sólo podrá aplicarse en la convocatoria de enero.

La evaluación de las prácticas de programación considerará la resolución de ejercicios propuestos antes y durante las sesiones (30%), así como la realización de una práctica final (70%). Para aprobar la práctica final será necesario superar un examen de modificación propuesto. La calificación de dicho examen será "Apto" o "No apto". Una calificación de "No apto" conlleva el suspenso de la práctica final y de la parte práctica de la asignatura.

Cada nota asociada con los bloques de prácticas y de prueba escrita supondrá el 40% de la nota final. El 20% restante estará asociado con el portafolio de actividades (no recuperable). Así, una vez superados los bloques de prácticas de programación y prueba

escrita, la nota final del estudiante se obtendrá de la siguiente forma:

$$\text{Nota\_Final} = 0,2 * \text{Nota\_Portafolio} + 0,4 * \text{Nota\_Prácticas} + 0,4 * \text{Nota\_Teoría}$$

En otro caso, la calificación se calculará según la siguiente tabla:

<b>Nota Prácticas</b>	NP	<5 o NP	≥5	<5 o NP
<b>Nota Teoría</b>	NP	<5 o NP	<5 o NP	≥5
<b>Nota Final</b>	NP	1	2	2

NP: No Presentado

La nota en cada bloque superado se guardará durante todas las convocatorias del curso.

### Evaluación global

Para superar la asignatura será necesario obtener una nota superior o igual a 5 en los bloques de teoría y prácticas de manera individual. Si una de las dos partes no se supera en la misma convocatoria, la parte superada NO se guardará para siguientes convocatorias.

Para superar el bloque de prácticas será necesario realizar la práctica final y superar un examen de modificación propuesto, cuya calificación se aplicará de la misma manera que en el sistema de evaluación continua. La nota obtenida de dicha práctica supondrá el 70% de la nota de todo el bloque de prácticas. El 30% restante se obtendrá de un examen de prácticas en el que se evaluarán contenidos no incluidos en la práctica final. Si la parte asociada con la práctica final no se supera, la calificación final de todo el bloque será de 2.

La evaluación del bloque de teoría se realizará mediante una prueba de tipo test y un examen de problemas. Este último será común para todos los estudiantes con independencia del tipo de evaluación al que se hayan acogido. Para superar el bloque de teoría, será necesario obtener una calificación mayor o igual a 5 en el examen de problemas. Tras superar esta nota, la nota final del bloque se obtendrá como el 30% de la nota de la prueba tipo test más el 70% de la nota del examen de problemas. Si la nota del examen de problemas fuera inferior a 5, la nota de todo el bloque de teoría será de 2.

Una vez superados los bloques de teoría y prácticas, la nota final del estudiante se obtendrá de la siguiente forma:

$$\text{Nota\_Final} = 0,5 * \text{Nota\_Teoría} + 0,5 * \text{Nota\_Prácticas}$$

En otro caso, la calificación se calculará según la siguiente tabla:

<b>Nota Teoría</b>	NP	<5 o NP	≥5	<5 o NP
<b>Nota Prácticas</b>	NP	<5 o NP	<5 o NP	≥5

Nota Final	NP	1	2	2
------------	----	---	---	---

### Bibliografía (básica y complementaria)

- [Angulo03] J.M. Angulo, J.L. Gutiérrez e I. Angulo. *Arquitectura de microprocesadores. Los Pentium a fondo*. Paraninfo, 2003.
- [Brey09] B.B. Brey. *Intel Microprocessors. Architecture, Programming and Interfacing*. Pearson – Prentice Hall, 2009.
- [Bryant15] R.E. Bryant, D.R. O'Hallaron. *Computer Systems: a Programmer's Perspective*. Pearson Education, 2015.
- [Charte09] F. Charte. *Ensamblador (edición 2009)*. Anaya multimedia, 2009.
- [Garcia00] J. García de Jalón, I. Aguinaga, A. Mora. *Aprenda LINUX como si estuviera en primero*. Universidad de Navarra, 2000.
- [Kussworn14] Modern x86 assembly language programming. Apress, 2014.
- [Pardo08] A. Pardo. *Programación en ensamblador de la arquitectura IA-32*. Universidad Carlos III de Madrid, 2008.
- [Patterson14] D.A. Patterson y J.L. Hennessy. *Computer Organization and Design (5th edition). The Hardware/Software Interface*. Morgan Kaufmann (Elsevier), 2014
- [Tanenbaum00] A.S. Tanenbaum. *Organización de computadores. Un enfoque estructurado*. Pearson Educación, 2000.
- [Seyfarth14] R. Seyfarth. *Introduction to 64 bit Intel Assembly Language Programming for Linux and OS X: Third Edition*. CreateSpace Independent Publishing Platform, 2014
- [Ubuntu] <https://ubuntu.com>
- [Ujaldón03] M. Ujaldón. *Arquitectura del PC. Volumen I: Microprocesadores*. Editorial Ciencia-3, 2003.

### Otros recursos y materiales docentes complementarios

El aula virtual de la asignatura contiene recursos relacionados con todos los temas del temario teórico, así como el material necesario para el desarrollo de la parte práctica.