

PLAN DOCENTE DE LA ASIGNATURA

Curso académico: 2025/2026

Identificación y características de la asignatura					
Código	501273				
Denominación (español)	Análisis y Diseño de Algoritmos (ADA)				
Denominación (inglés)	Algorithm Design and Analysis				
Titulaciones	Grado en Ing. Informática en Ingeniería de Computadores Grado en Ing. Informática en Ingeniería del Software				
Centro	Escuela Politécnica				
Módulo	Común a la rama de Informática				
Materia	Programación				
Carácter	Obligatoria	ECTS	6	Semestre	3
Profesorado					
Nombre	Despacho	Correo-e			
Álvaro Rubio Largo	I.2 (15)	arl@unex.es			
José Ángel Barriga Corchero	Lab. Quercus	jose@unex.es			
Félix Expósito Durán	I.2 (15)	felixed@unex.es			
Área de conocimiento	Lenguajes y Sistemas Informáticos				
Departamento	Ingeniería de Sistemas Informáticos y Telemáticos				
Profesor coordinador	Álvaro Rubio Largo				
Competencias					
COMPETENCIAS BÁSICAS					
<i>Establecidas para el Grado en el ANEXO I 3.2 del RD 861/2010</i>					
<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 soluciones a un público tanto especializado como no especializado.</p> <p>CB5. Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.</p>					

COMPETENCIAS GENERALES

CG08. Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.

CG09. Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.

CG10. Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática, 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.

COMPETENCIAS ESPECÍFICAS

Según los planes de estudio aprobados, esta asignatura debe cubrir, total o parcialmente, las siguientes competencias específicas y sus resultados de aprendizaje

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.

CI07. Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.

COMPETENCIAS TRANSVERSALES

Según los planes de estudio aprobados y los acuerdos de la comisión de calidad de las titulaciones, esta asignatura debe cubrir, total o parcialmente, las siguientes competencias transversales y sus resultados de aprendizaje en un nivel intermedio

CT03. Capacidad para resolver problemas.

CT07. Capacidad de análisis y síntesis.

Contenidos

Análisis de la complejidad y computabilidad. Utilización de lenguajes estructurados y orientados a objeto para el desarrollo de sistemas software. Estructuras de datos básicas, sus aplicaciones y propiedades.

La materia comprende tres partes; cada una de ellas se desarrolla tanto en el aspecto teórico como en el práctico: (i) se estudian los diferentes aspectos que comprende el análisis de algoritmos en cuanto a eficiencia temporal, espacial y eficacia, (ii) se estudian diferentes estructuras de datos continuando las del curso anterior y (iii) se estudian diferentes esquemas de diseño de algoritmos.

Temario

TEMA 1: COMPLEJIDAD ALGORÍTMICA

Contenidos: Este tema aborda la eficiencia de los algoritmos, tanto espacial como temporal, y la noción de complejidad. Se estudia cómo determinar el tiempo de ejecución en los casos peor, mejor y medio, y se presentan medidas relevantes para problemas comunes. Se explican reglas para calcular el tiempo de ejecución de algoritmos y se introduce la notación asintótica con sus cotas: superior (O), inferior (Ω) y orden exacto (Θ), junto con sus propiedades.

También se analizan medidas frecuentes mediante ejemplos concretos. Se estudian algoritmos de búsqueda y ordenación, y se profundiza en la complejidad de algoritmos recursivos a través de casos de estudio.

Descripción de las actividades prácticas: Se incluyen ejercicios de resolución de cuestiones multirrespuesta relacionadas con los contenidos tratados, análisis de algoritmos iterativos evaluando los casos peor, mejor y medio, y análisis de algoritmos recursivos utilizando el método de sustitución de recurrencias y los teoremas Maestro (por sustracción y división).

TEMA 2: ESTRUCTURAS DE ALMACENAMIENTO DE DATOS NO LINEALES Y ÁRBOLES

Contenidos: Este tema se centra en las estructuras de almacenamiento no lineales, como las tablas, las colas de prioridad y los árboles, incluyendo los montículos. Se abordan su descripción, formas de implementación y los métodos asociados a cada una.

Descripción de las actividades prácticas: Las actividades prácticas comprenden la resolución de cuestiones multirrespuesta relacionadas con los conceptos teóricos, así como el diseño, implementación y análisis de problemas utilizando estructuras no lineales y árboles.

TEMA 3: GRAFOS COMO ESTRUCTURA DE DATOS

Contenidos: Se estudian los grafos como estructuras de almacenamiento y gestión de información, considerando distintas formas de implementación y los métodos aplicables. Además, se analiza la complejidad de los principales algoritmos sobre grafos.

Descripción de las actividades prácticas: Las actividades incluyen cuestiones multirrespuesta, resolución de problemas sobre conceptos básicos de grafos y aplicación de algoritmos como Prim y Kruskal para árboles de expansión mínima. Se trabaja en la detección de caminos en grafos valuados y no-valuados mediante multiplicación de matrices y el algoritmo de Warshall, así como en la obtención de caminos mínimos entre pares de vértices usando los algoritmos de Warshall y Floyd. También se utiliza el algoritmo de Dijkstra para encontrar caminos mínimos entre vértices específicos y se practican otros algoritmos relevantes sobre grafos.

TEMA 4: ESQUEMA ALGORÍTMICO DE DIVIDE Y VENCERÁS

Contenidos: En este tema se explora el paradigma de divide y vencerás, aplicándolo a la multiplicación de enteros muy grandes, búsqueda binaria, ordenación, mezcla, quicksort, búsqueda de la mediana, multiplicación de matrices y cálculo de potencias.

Descripción de las actividades prácticas: Las prácticas incluyen la resolución de cuestiones multirrespuesta sobre los contenidos estudiados, el planteamiento y diseño de soluciones a problemas tipo aplicando el esquema divide y vencerás, y el análisis de la complejidad de los algoritmos propuestos.

TEMA 5: ALGORITMOS VORACES

Contenidos: Se estudian las características generales de los algoritmos voraces y su aplicación a problemas clásicos como árboles de recubrimiento mínimo, caminos mínimos, el problema del cambio, el problema de la mochila y la planificación de tareas.

Descripción de las actividades prácticas: Las actividades consisten en la resolución de cuestiones multirrespuesta, el diseño de soluciones a problemas empleando el enfoque voraz y el análisis de su complejidad.

TEMA 6: ESQUEMAS DE VUELTA ATRÁS Y DE RAMIFICACIÓN Y PODA

Contenidos: Este tema cubre las características generales de los esquemas de vuelta atrás y de ramificación y poda. Se aplican estos enfoques a la exploración de grafos (recorridos en

profundidad y en anchura), así como a problemas como el recorrido de Hamilton, el de Euler, la mochila, las n reinas y la asignación de tareas.

Descripción de las actividades prácticas: Las actividades prácticas incluyen la resolución de cuestiones multirrespuesta y el planteamiento y diseño de soluciones a problemas utilizando tanto vuelta atrás como ramificación y poda, junto con el estudio de la complejidad de dichos algoritmos.

TEMA 7: ESQUEMA DE PROGRAMACIÓN DINÁMICA

Contenidos: Se presentan las características generales de la programación dinámica y su aplicación a la multiplicación de matrices, los algoritmos de Warshall y Floyd, el problema del cambio, el problema de la mochila, y los caminos mínimos en grafos. También se abordan enfoques recursivos como el cálculo de coeficientes binomiales y problemas como el del campeonato mundial.

Descripción de las actividades prácticas: Se realizan actividades de resolución de cuestiones multirrespuesta, diseño de soluciones a problemas utilizando programación dinámica y análisis de su complejidad.

Actividades formativas

Horas de trabajo del alumno/a por tema		Horas Gran grupo	Actividades prácticas				Actividad de seguimiento	No presencial
Tema	Total	GG	CH	L	O	S	TP	EP
1	24	6,5		4,5				15
2	16	4		3				9
3	32	8		3			1,5	19,5
4	4	1		1,5				1,5
5	24	6		3				15
6	40	10		5,5				24,5
7	4	1		0,5				2,5
Evaluación	4	1		1,5				1,5
TOTAL	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

Cuando trabajamos en “Grupo Grande”, las actividades expositivas cumplen la función de vertebrar el proceso de aprendizaje del alumno durante el desarrollo de la asignatura, habitualmente esta actividad expositiva se acompaña del uso de herramientas para visualizar el material del curso, además del uso tradicional de la pizarra. En el desarrollo de estas sesiones el estudiante debe resolver problemas y responder a cuestiones planteadas por el profesor. También los estudiantes deberán proponer preguntas y plantear dudas a sus compañeros y al profesor.

Cuando trabajamos en “Grupo Pequeño” se seguirá una metodología de aprendizaje Basado en Problemas basándose sobre todo en la discusión y debate por parte de los alumnos acerca de diferentes soluciones para un mismo problema.

Además, se dispone del Aula Virtual donde también es posible abrir diferentes foros de discusión para diferentes problemas donde se aplican de forma práctica los conceptos teóricos de la asignatura.

Los alumnos que tengan dudas sobre la comprensión de los conceptos y problemas planteados en el aula pueden someter a debate las soluciones de problemas con el profesor utilizando las horas de tutorías tradicionales.

Resultados de aprendizaje

RESULTADOS DE APRENDIZAJE QUE SE CORRESPONDEN CON LA MEMORIA VERIFICADA DEL TÍTULO:

- Justifica la utilización de distintos paradigmas de programación y plataformas de desarrollo de software en un determinado contexto.
- Busca, analiza, sintetiza y critica nueva información para aprender nuevos lenguajes, algoritmos, técnicas, paradigmas y metodologías de programación aplicables a distintas áreas, teniendo como objetivo la actualización continua de los conocimientos y competencias.
- Analiza, planifica, diseña y desarrolla soluciones algorítmicas y programas robustos y correctos a problemas planteados, argumentando las decisiones tomadas, evaluando el resultado final y documentando el código y el proceso.

Al término del periodo de enseñanza de esta asignatura, el estudiante será capaz de:

- Aplicar el concepto de eficiencia temporal y espacial a los algoritmos.
- Analizar la eficiencia de algoritmos, tanto iterativos como recursivos.
- Determinar las estructuras de datos adecuadas, según el problema, para almacenamiento de información en memoria principal.
- Comparar diferentes alternativas de implementación de estructuras de datos, razonando sobre la eficiencia espacial y temporal de tales estructuras.
- Diferenciar diversas técnicas de diseño de algoritmos y razonar sobre su aplicabilidad y adecuación a la hora de resolver un problema.
- Comparar la eficiencia temporal de distintas soluciones alternativas que permiten resolver un mismo problema.
- Poder utilizar de manera eficaz un entorno de programación que incluya herramientas de edición, compilación, depuración y documentación de programas.
- Utilizar diferentes técnicas para modelar problemas complejos, planteando soluciones eficientes y efectivas.

Sistemas de evaluación

Los criterios de evaluación son:

- Adquisición de conocimientos relacionados con esquemas de programación y con análisis de algoritmos y de estructuras de datos.

- Resolución de problemas relacionándolos con los diferentes esquemas de programación.
- Capacidad para analizar la eficiencia y la eficacia de los algoritmos.
- Resolución clara, concisa y estructurada de los ejercicios y trabajos a presentar.
- Participación activa en clase resolviendo problemas planteados.

El sistema de evaluación se rige por la *Normativa de Evaluación de las Titulaciones Oficiales de Grado y Máster de la Universidad de Extremadura (RESOLUCIÓN de 26 de octubre de 2020, del Rector, por la que se ejecuta el acuerdo adoptado por el Consejo de Gobierno por el que se aprueba la Normativa de Evaluación de las Titulaciones oficiales de Grado y Máster de la Universidad de Extremadura, DOE N°212, 3 de noviembre de 2020)*. De acuerdo con la normativa vigente, los estudiantes pueden elegir entre las siguientes modalidades:

- **Modalidad de evaluación continua (EC):** sistema de evaluación constituido por diversas actividades distribuidas a lo largo del semestre de docencia de una asignatura. Esta modalidad puede incluir, además, oportunidades de recuperación de bloques al finalizar el semestre.
- **Modalidad de evaluación global (EG):** sistema de evaluación constituido exclusivamente por una prueba final o conjunto de pruebas finales que engloban todos los bloques, y que se realizará en la fecha oficial de cada convocatoria.

Durante las primeras cuatro semanas del curso, el estudiante podrá elegir, a través del Campus Virtual, la modalidad con la que quiere ser evaluado para cada convocatoria. En caso de ausencia de solicitud expresa por parte del estudiante, la modalidad asignada será la de evaluación continua (EC).

En ambas modalidades, el estudiante puede obtener la calificación máxima. La diferencia fundamental entre ellas radica en que, en la Evaluación Continua, se incluyen actividades de seguimiento durante el semestre que no pueden recuperarse posteriormente.

A continuación, se describen ambas modalidades y el modo de cálculo de la calificación final.

EVALUACIÓN CONTINUA (EC)

A lo largo del semestre, el estudiante participa en actividades evaluables que permiten mejorar la calificación obtenida en los bloques de teoría y práctica.

En el bloque teórico (T), se plantearán varios cuestionarios teóricos, generalmente al finalizar determinados temas. Estas pruebas permiten consolidar progresivamente los contenidos mediante evaluaciones periódicas. En el bloque práctico (P), se propondrán problemas de programación cuya resolución se llevará a cabo a través de un juez *online*. Cada problema tendrá una fecha límite específica, y la entrega fuera de plazo no será posible. Estas actividades no son recuperables: la no realización dentro del plazo establecido implicará una calificación de cero para la actividad correspondiente.

Al finalizar el curso, el estudiante deberá presentarse obligatoriamente a dos pruebas: un **test final de teoría** y una **prueba práctica final**. Esta última será individual, se realizará de forma presencial en un equipo de la Universidad de Extremadura, y su resolución será evaluada en tiempo real.

La calificación de cada bloque se calculará teniendo en cuenta tanto la prueba final como el rendimiento durante el semestre, mediante las siguientes expresiones:

$$T = CTf + \max(0, (CCT - CTf) \times 0,3)$$

$$P = CPf + \max(0, (CPP - CPf) \times 0,3)$$

Donde:

- **CTf** es la calificación del test final de teoría.
- **CPf** es la calificación de la prueba final práctica.
- **CCT** es la media ponderada de los cuestionarios teóricos realizados durante el curso.
- **CPP** es la media ponderada de la resolución de los problemas de programación realizados durante el curso.

Para que las calificaciones de T y P puedan incorporarse al cálculo de la nota final, el estudiante debe obtener al menos 4 puntos sobre 10 tanto en CTf como en CPf. Cualquier calificación igual o superior a 4 puntos en dichas pruebas se conservará para el resto de las convocatorias del curso actual.

EVALUACIÓN GLOBAL (EG)

En esta modalidad, el estudiante será evaluado exclusivamente mediante pruebas finales. En cada convocatoria oficial, deberá realizar un test final de teoría y una prueba final práctica.

Las calificaciones se calculan directamente a partir de estas pruebas, por tanto:

$$T = CTf$$

$$P = CPf$$

Donde:

- **CTf** es la calificación del test final de teoría.
- **CPf** es la calificación de la prueba final práctica.

También en este caso es necesario alcanzar al menos 4 puntos sobre 10 en ambas pruebas para que sus notas puedan utilizarse en el cálculo de la nota final.

CÁLCULO DE LA NOTA FINAL

La nota final de la asignatura se obtiene como la media ponderada de las calificaciones de teoría y práctica, con igual peso en ambos bloques:

$$\text{Nota Final} = 0,5 \times T + 0,5 \times P$$

Para superar la asignatura, es necesario obtener una nota final igual o superior a 5 puntos sobre 10. Si una de las dos calificaciones parciales (teoría o práctica) es inferior a 4, se consignará automáticamente la calificación de SUSPENSO (3).

AUTORÍA DE LAS ACTIVIDADES

En todas las actividades de evaluación se presume la autoría íntegra por parte del estudiante. Si existen indicios razonables de que el contenido ha sido copiado de otro

estudiante, o generado total o parcialmente mediante herramientas de inteligencia artificial, el profesorado podrá exigir al estudiante una verificación adicional de la autoría. Asimismo, se reserva el derecho a aplicar las medidas disciplinarias contempladas en la normativa académica vigente.

Bibliografía (básica y complementaria)

La bibliografía o documentación de lectura obligatoria:

- **[Brassard97]** G. Brassard; P. Bratley. *Fundamentos de Algoritmia*. Prentice Hall, 2000.
- **[Martí04]** Narciso Martí; Yolanda Ortega; José A. Verdejo. *Estructuras de datos y métodos algorítmicos. Ejercicios resueltos*. Editorial Pearson – Prentice Hall (Prentice Práctica), 2004.
- **[Guerequeta00]** R. Guerequeta; A. Vallecillo. *Técnicas de Diseño de Algoritmos*, 2ª edición. Servicio de Publicaciones de la Universidad de Málaga, 2000. (<http://polaris.lcc.uma.es/~av/Libro/>).

Otra bibliografía de apoyo:

- **[Cormen01]** T.H. Cormen, C.E. Leiserson, et als. *Introduction to Algorithms*. Ed. M.I.T. Press McGraw Hill, 2001.
- **[Aho88]** A. Aho; J. Hopcroft; J. Ullman. *Estructuras de datos y algoritmos*. Editorial AddisonWesley Americana, 1988.
- **[Horowitz95]** E. Horowitz; S. Sahní. *Fundamentals of data structures in C++*. Editorial Computer Science Press, 1995.
- **[Peña98]** R. Peña Martí. *Diseño de programas. Formalismo y abstracción*. 2ª ed. Prentice-Hall, 1998.

Otros recursos y materiales docentes complementarios

Se utiliza con frecuencia los recursos de la plataforma virtual Moodle; cuestionarios, resolución de problemas en grupo dentro del aula, debates y discusión sobre la eficiencia de diferentes soluciones.