

PLAN DOCENTE DE LA ASIGNATURA

Curso académico: 2024/2025

Identificación y características de la asignatura			
Código	501273	Créditos ECTS	6
Denominación (español)	Análisis y Diseño de Algoritmos (ADA)		
Denominación (inglés)	Algorithm Design and Analysis		
Titulaciones	Grado en Ingeniería Informática en Ingeniería de Computadores Grado en Ingeniería Informática en Ingeniería del Software		
Centro	Escuela Politécnica		
Semestre	3	Carácter	Obligatoria
Módulo	Común a la rama de Informática		
Materia	Programación		
Profesor/es			
Nombre	Despacho	Correo-e	Página web
M ^a Luisa Durán Martín-Meras	I.2 (17)	mlduran@unex.es	Web EPCC (mlduran)
Álvaro Rubio Largo	I.2 (15)	arl@unex.es	Web EPCC (arl)
Manuel Jesús Azabal	I.2 (19)	manuel@unex.es	Web EPCC (manuel)
José Ángel Barriga Corchero	Lab. Quercus (Ed. Invest.)	jose@unex.es	Web EPCC (jose)
Área de conocimiento	Lenguajes y Sistemas Informáticos		
Departamento	Ingeniería de Sistemas Informáticos y Telemáticos		
Profesor coordinador (si hay más de uno)	Álvaro Rubio Largo		
Competencias			
Competencias Básicas			
<i>Establecidas para el Grado en el ANEXO I 3.2 del RD 861/2010:</i>			
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.		
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.		
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.		

CB4.	Que los estudiantes puedan transmitir información, ideas, problemas y 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	
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	
<i>Según los planes de estudio aprobados, esta asignatura debe cubrir, total o parcialmente, las siguientes competencias específicas y sus resultados de aprendizaje</i>	
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	
<i>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</i>	
CT03.	Capacidad para resolver problemas.
CT07.	Capacidad de análisis y síntesis.
Contenidos	
Breve descripción del contenido	
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:

- Se estudian los diferentes aspectos que comprende el análisis de algoritmos en cuanto a eficiencia temporal, espacial y eficacia.
- Se estudian diferentes estructuras de datos continuando las del curso anterior.
- Se estudian diferentes esquemas de diseño de algoritmos.

Temario de la asignatura

Tema 1: Complejidad algorítmica

Contenidos:

- Eficiencia de un algoritmo: eficiencia espacial y eficiencia temporal
- Noción de complejidad
- Determinación del tiempo de ejecución de un algoritmo: casos peor, mejor, medio
- Medidas significativas de problemas comunes
- Reglas para el cálculo del tiempo de ejecución de un algoritmo
- Notación asintótica, utilidades y significado de las cotas:
 - cota superior (O). Propiedades
 - cota inferior (Ω). Propiedades
 - orden exacto (θ). Propiedades
- Medidas frecuentes. Ejemplos
- Análisis de algoritmos de búsqueda
- Análisis de algoritmos de ordenación
- Complejidad de algoritmos recursivos. Casos de estudio

Descripción de las actividades prácticas del Tema 1:

- Resolución de cuestiones multirrespuesta de los contenidos vistos.
- Análisis de algoritmos iterativos. Caso peor, mejor y medio.
- Análisis de algoritmos recursivos. Método de sustitución de las recurrencias. Teoremas Maestro (sustracción y división).

Tema 2: Estructuras de almacenamiento de datos no lineales y árboles

Contenidos:

- Tablas: descripción, implementación y métodos
- Colas de prioridad: descripción, implementación y métodos
- Árboles y montículos: descripción, implementación y métodos

Descripción de las actividades prácticas del Tema 2:

- Resolución de cuestiones multirrespuesta de los contenidos vistos.
- Diseño, diferentes formas de implementación y discusión de problemas con estructuras no lineales y árboles.

Tema 3: Grafos como estructura de datos

Contenidos:

- Grafos como estructura de almacenamiento y gestión de información: descripción, diferentes formas de implementación y sus métodos
- Estudio de la complejidad de los principales algoritmos sobre grafos

Descripción de las actividades prácticas del Tema 3:

- Resolución de cuestiones multirrespuesta de los contenidos vistos.

- Resolución de problemas sobre conceptos básicos de grafos.
- Resolución de problemas mediante el uso de árboles de expansión mínimo. Algoritmos de *Prim* y *Kruskal*.
- Resolución de problemas para determinar la existencia de caminos en un grafo valuado y no-valuado. Multiplicación de matrices. Algoritmo *Warshall*.
- Resolución de problemas para encontrar el camino mínimo entre todos los pares de vértices en un grafo. Algoritmos *Warshall* y *Floyd*.
- Resolución de problemas para encontrar el camino mínimo entre dos pares de vértices de un grafo. Algoritmo *Dijkstra*.
- Prácticas con otros algoritmos tipo sobre grafos.

Tema 4: Esquema algorítmico de Divide y vencerás

Contenidos:

- Multiplicación de enteros muy grandes
- Búsqueda binaria
- Ordenación
- Mezcla
- *Quicksort*
- Búsqueda de la mediana
- Multiplicación de matrices
- Cálculo de potencias

Descripción de las actividades prácticas del Tema 4:

- Resolución de cuestiones multirrespuesta de los contenidos vistos.
- Planteamiento y diseño de soluciones a problemas tipo utilizando el esquema algorítmico divide y vencerás.
- Estudio de la complejidad

Tema 5: Algoritmos voraces

Contenidos:

- Características generales
- Algoritmos sobre grafos
- Árbol de recubrimiento mínimo
- Caminos mínimos
- El problema del cambio
- El problema de la mochila
- Planificación de tareas

Descripción de las actividades prácticas del Tema 5:

- Resolución de cuestiones multirrespuesta de los contenidos vistos.
- Planteamiento y diseño de soluciones a problemas dados utilizando el esquema algorítmico voraz.
- Estudio de la complejidad

Tema 6: Esquemas de vuelta atrás y de ramificación y poda

Contenidos:

- Características generales
- Algoritmos de exploración de grafos
- Recorrido en profundidad
- Recorrido en anchura
- Recorrido de *Hamilton*
- Recorrido de *Euler*

- El problema de la mochila
- El problema de las n reinas
- Características generales del esquema de ramificación y poda
- El problema de la asignación de tareas
- El problema de la mochila

Descripción de las actividades prácticas del Tema 6:

- Resolución de cuestiones multirrespuesta de los contenidos vistos.
- Planteamiento y diseño de soluciones a problemas dados utilizando el esquema algorítmico vuelta atrás.
- Planteamiento y diseño de soluciones a problemas dados utilizando el esquema algorítmico ramificación y poda.
- Estudio de la complejidad

Tema 7: Esquema de programación dinámica

Contenidos:

- Características generales
- Multiplicación de matrices
- *Warshall y Floyd*
- El problema del cambio
- El problema de la mochila
- Caminos mínimos en grafos
- Enfoques que aplican recursión
- Cálculo de los coeficientes binomiales
- El problema del campeonato mundial

Descripción de las actividades prácticas del Tema 7:

- Resolución de cuestiones multirrespuesta de los contenidos vistos.
- Planteamiento y diseño de soluciones a problemas dados utilizando el esquema algorítmico programación dinámica.
- Estudio de la complejidad.

Actividades formativas

Horas de trabajo del alumno/a por tema		Horas Gran grupo	Actividades prácticas				Actividad de seguimiento	No presencial
Tema	Total	GG	PCH	LAB	ORD	SEM	TP	EP
1	27,5	9		3,5				15
2	14	4		2				8
3	28,5	8,5		3,5			1,5	15
4	4,5	1		1,5				2
5	12	3		3				6
6	30,5	7		6,5				17
7	4	1		1				2
Evaluación	29	4		1,5				24
TOTAL	150	37,5		22,5			1,5	88,5

GG: Grupo Grande (85 estudiantes).

PCH: prácticas clínicas hospitalarias (7 estudiantes)

LAB: prácticas laboratorio o campo (15 estudiantes)

ORD: prácticas sala ordenador o laboratorio de idiomas (20 estudiantes)

SEM: clases problemas o seminarios o casos prácticos (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:

1. Adquisición de conocimientos relacionados con esquemas de programación y con análisis de algoritmos y de estructuras de datos.
2. Resolución de problemas relacionándolos con los diferentes esquemas de programación.
3. Capacidad para analizar la eficiencia y la eficacia de los algoritmos.
4. Resolución clara, concisa y estructurada de los ejercicios y trabajos a presentar.
5. 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 Nº212, 3 de noviembre de 2020*). De acuerdo con la normativa vigente, los estudiantes pueden elegir entre las siguientes modalidades:

1. **Modalidad de evaluación continua (EvalCont)**: sistema de evaluación constituido por diversas actividades distribuidas a lo largo del semestre de docencia de una asignatura. Esta modalidad puede incluir además una prueba final, entendida esta como el conjunto de actividades de evaluación que tienen condicionada su celebración a la fecha oficial de examen para cada convocatoria.
2. **Modalidad de evaluación global (EvalGlob)**: sistema de evaluación constituido exclusivamente por una prueba final, que englobe todos los contenidos de la asignatura y que se realizará en la fecha oficial de cada convocatoria.

Durante las primeras 4 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 (EvalCont).

En ambas modalidades de evaluación, el estudiante puede optar a la calificación máxima. La principal diferencia entre ambas modalidades es la ponderación de cada uno de los bloques evaluables: **Actividades Semipresenciales (AS), Prueba Escrita (PE) y Prueba de Laboratorio (PL)**.

A continuación, se detallan las características de cada uno de los bloques de evaluación:

- **AS (no recuperable)**: Se realizarán cuatro actividades de evaluación continua a lo largo del cuatrimestre (**AS1, AS2, AS3 y AS4**), con el objetivo de conseguir que el estudiante mantenga compaginadas sus horas de estudio de la asignatura (tiempo no-presencial) con el avance del temario durante el cuatrimestre (tiempo presencial). Una parte de estas actividades se realizará en el aula y otra parte es trabajo personal. Cada actividad se evalúa sobre 10 puntos (sin calificación mínima). La nota de este bloque se guarda para todas las convocatorias del curso y se calcula como:

$$\text{AS} = 25\% * \text{AS1} + 25\% * \text{AS2} + 25\% * \text{AS3} + 25\% * \text{AS4}$$

- **PE (recuperable):** Este bloque está compuesto por dos pruebas, una con los contenidos de los temas 1 y 2 (**Prueba Escrita 1, PE1**) y la otra con el resto de los temas (**Prueba Escrita 2, PE2**). La estructura de PE1 y PE2 es idéntica:
 - **Test:** cuestionario de preguntas de respuestas múltiples de las cuales solo una es correcta. Se requiere una calificación mínima de 4 puntos sobre 10 para calcular la media de la prueba. El tiempo máximo para la realización del test será de 30 minutos.
 - **Problemas:** ejercicios orientados a medir la capacidad del estudiante a la hora de analizar un problema, identificar información relevante, formular hipótesis, tomar decisiones y proporcionar soluciones. Se requiere una calificación mínima de 3 puntos sobre 10 en cada ejercicio para calcular la media de la prueba. El tiempo máximo para resolver los problemas planteados será de 90 minutos.

Siempre que se alcancen las calificaciones mínimas en cada una de las partes (Test y Problemas), la nota en **PE1** y **PE2** podrá ser calculada como:

$$\mathbf{PE1 = 30\% * Test_{PE1} + 70\% * Problemas_{PE1}}$$

$$\mathbf{PE2 = 30\% * Test_{PE2} + 70\% * Problemas_{PE2}}$$

La nota de una de las partes (**PE1** o **PE2**) superada con una nota igual a superior a 5 puntos sobre 10, se guardará para todas las convocatorias del mismo curso académico. No se guardará ninguna calificación para ninguna convocatoria de cursos posteriores.

La nota de este bloque se calcula como:

$$\mathbf{PE = 50\% * PE1 + 50\% * PE2}$$

- **PL (recuperable):** Esta prueba sólo la deben hacer aquellos estudiantes que hayan elegido "evaluación global". Se realizarán diversas pruebas de prácticas de programación en un laboratorio de informática. El tiempo máximo para la prueba de laboratorio será de 90 minutos.

Una vez definidos los bloques evaluables; a continuación, se muestra la ponderación asignada según el tipo de evaluación (EvalCont o EvalGlob):

$$\mathbf{EvalCont^1 = 25\% * AS + 75\% * PE + 0\% * PL}$$

$$\mathbf{EvalGlob = 0\% * AS + 75\% * PE + 25\% * PL}$$

¹: A mediados del semestre se realizará un examen parcial con los contenidos de **PE1** y con los mismos requisitos que en el examen final. Los estudiantes que obtengan una nota igual o superior a 5 en ese examen parcial estarán exentos de examinarse de esa primera parte en el examen final.

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 Addison_Wesley Americana, 1988.
- **[Horowitz95]** E. Horowitz; S. Sahni. 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.