



DESCRIPCIÓN DE LA ASIGNATURA

Grado/Máster en:	Graduado/a en Ingeniería del Software por la Universidad de Málaga
Centro:	Escuela Técnica Superior de Ingeniería Informática
Asignatura:	Análisis y Diseño de Algoritmos
Código:	201
Tipo:	Obligatoria
Materia:	Programación de Computadores
Módulo:	Formación común
Experimentalidad:	69 % teórica y 31 % práctica
Idioma en el que se imparte:	Inglés, Español
Curso:	2
Semestre:	1
Nº Créditos	6
Nº Horas de dedicación del estudiante:	150
Nº Horas presenciales:	60
Tamaño del Grupo Grande:	72
Tamaño del Grupo Reducido:	30
Página web de la asignatura:	http://informatica.cv.uma.es/

EQUIPO DOCENTE

Departamento: LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN

Área: LENGUAJES Y SISTEMAS INFORMÁTICOS

Nombre y Apellidos	Mail	Teléfono Laboral	Despacho	Horario Tutorías
Coordinador/a: CARLOS COTTA PORRAS	ccotta@uma.es	952137158	3.2.49 - E.T.S.I. INFORMÁTICA	Todo el curso: Lunes 09:30 - 10:30, Jueves 11:00 - 13:00, Martes 09:30 - 12:30
JOSE ANTONIO ONIEVA GONZALEZ	onieva@uma.es	952132898	3.2.40 - E.T.S.I. INFORMÁTICA	Primer cuatrimestre: Miércoles 15:15 - 17:15, Miércoles 10:00 - 12:00, Jueves 17:15 - 19:15
RAFAEL MORALES BUENO	rmorales@uma.es	952131395	3.2.22 - E.T.S.I. INFORMÁTICA	Primer cuatrimestre: Miércoles 12:30 - 18:30
GUILLERMO LOPEZ GARCIA	guilopgar@uma.es		3.3.12 - E.T.S.I. INFORMÁTICA	Primer cuatrimestre: Miércoles 17:00 - 19:00
JUAN MIGUEL ORTIZ DE LAZCANO LOBATO	lazcano@uma.es	952132805	3.2.21 - E.T.S.I. INFORMÁTICA	Todo el curso: Martes 09:30 - 13:30, Lunes 09:30 - 11:30

RECOMENDACIONES Y ORIENTACIONES

Esta asignatura requiere conocimientos de lógica y matemática finita para la comprensión de los temas dedicados al análisis de la Complejidad Algorítmica y a la Especificación de Programas. Por otro lado, para los bloques dedicados a las técnicas de diseño de algoritmos es necesario que los estudiantes tengan conocimientos sobre programación estructurada y orientada a objetos, así como sobre técnicas recursivas e iterativas para la resolución de problemas. Esta asignatura se imparte paralelamente a Estructura de Datos, lo que permite el uso de estructuras de datos complejas en la implementación de los algoritmos.

CONTEXTO

Esta asignatura se imparte en el primer cuatrimestre del segundo curso de los tres grados de Informática: Ingeniería Informática, Ingeniería del Software e Ingeniería de Computadores. En este mismo cuatrimestre se imparte también la asignatura Estructuras de Datos. Ambas asignaturas completan el estudio de las técnicas de programación básicas que comienza con las asignaturas de programación de primer curso. Análisis y Diseño de Algoritmos es una asignatura formal, en el sentido de que se hace hincapié en el diseño de programas correctos y eficientes, proporcionando herramientas matemáticas para que el estudiante pueda analizar sus algoritmos. Asimismo, la asignatura hace un recorrido por los métodos de diseño de algoritmos más importantes, que permiten conocer algoritmos clásicos y soluciones a los mismos utilizando distintas técnicas. Por otro lado, la asignatura también tiene un fuerte carácter aplicado, pues se realizan implementaciones de algunos algoritmos en el laboratorio.

Los grupos en los que se imparte la asignatura están muy coordinados, de manera que el temario, relaciones de problemas, prácticas de laboratorio, exámenes y demás recursos son compartidos por todos los grupos.

COMPETENCIAS

1 Competencias generales y básicas.

BÁSICAS

- CB02** 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.
- CB04** Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
- CB05** Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.



1 Competencias generales y básicas.

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

2 Competencias específicas.

Formacion Comun

- CC05** Conocimiento, administración y mantenimiento sistemas, servicios y aplicaciones informáticas.
- CC06** 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
- CC07** Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.
- CC08** Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

CONTENIDOS DE LA ASIGNATURA

Complejidad algorítmica

Tema 1.- Complejidad algorítmica:Notaciones O, Omega, Theta. Ecuaciones recurrentes. Clases de complejidad. Algoritmos ordenación básicos (burbuja, inserción, selección).

Especificación de programas imperativos.

Tema 2.- Especificación de programas imperativos: Especificaciones con pre- y post-condiciones.

Técnicas de diseño de algoritmos

Tema 3.- Divide y vencerás: Concepto. Ordenación y búsqueda (búsqueda binaria, mergesort, quicksort). Algoritmos recursivos.

Tema 4.- Programación dinámica: Optimalidad de Bellman. Resolución bottom-up y top-down. Ejemplos (mochila 0-1, ...)

Tema 5.- Algoritmos voraces: Concepto. Algoritmos voraces para resolución exacta (mochila continua, monedas). Heurísticas voraces (TSP, mochila 0-1). Árboles de recubrimiento mínimo: algoritmos de Kruskal y Prim. Caminos mínimos. Algoritmo de Dijkstra.

Tema 6.- Vuelta atrás: Conceptos y ejemplos (n-reinas). Vuelta atrás para enumeración y optimización.

Tema 7.- Ramificación y poda: Conceptos y ejemplos (TSP, mochila 0-1). Estrategias de relajación

ACTIVIDADES FORMATIVAS

Actividades presenciales

Actividades expositivas

Lección magistral

Actividades prácticas en instalaciones específicas

Prácticas en laboratorio

Actividades no presenciales

Actividades prácticas

Resolución de problemas

Estudio personal

Estudio personal

ACTIVIDADES DE EVALUACIÓN

Actividades de evaluación no presenciales

Actividades de evaluación de la asignatura con participación alumnos

Otras actividades no presenciales eval.asignatura: Entrega de prácticas de laboratorio

Actividades de evaluación presenciales



Actividades de evaluación presenciales

Actividades de evaluación del estudiante

- Examen parcial
- Examen final
- Participación en clase

RESULTADOS DE APRENDIZAJE / CRITERIOS DE EVALUACIÓN

Los alumnos que superen esta asignatura deben:

- [R1] conocer cómo calcular la complejidad de un algoritmo iterativo y recursivo (CC6),
- [R2] saber emplear formalismo lógico para especificar las condiciones de funcionamiento y salida de un algoritmo (CC8).

Además, el alumno tendrá conocimiento de técnicas de diseño de algoritmos (CG8, CC6) lo que les capacitará para:

- [R3] aplicarlos a problemas específicos a los que se enfrentarán durante el curso (CB2, CG9).
- [R4] razonar sobre la aplicabilidad e idoneidad de cada uno de ellos en cada situación concreta, (CB4, CB5, CG9, CC6).
- [R5] implementarlos de manera eficiente haciendo uso de las herramientas y estructuras de datos más adecuados (CC5, CC7, CC8).
- [R6] comparar e implementar los principales algoritmos de ordenación y búsqueda (CC6, CC8)

Cada uno de estos resultados de aprendizaje llevan implícitos en su propia definición el criterio de evaluación, cuyo desarrollo se detalla formalmente en el apartado de "Procedimiento de Evaluación".

PROCEDIMIENTO DE EVALUACIÓN

Para la evaluación, la asignatura se divide en cuatro bloques (tres teóricos y uno práctico) para un total de 10 puntos

- B1 - Complejidad, Especificación y Divide y Vencerás (Temas 1-3)
- B2 - Programación Dinámica y Algoritmos Voraces (Temas 4 y 5)
- B3 - Algoritmos de Exploración de Espacios de Estados (Temas 6 y 7)
- P - Prácticas de Laboratorio

El bloque B1 abarca fundamentalmente los resultados de aprendizaje R1, R6 y R2; Los resultados R3 y R4 se evaluarán a través de los bloques B1, B2 y B3, mientras que el bloque P permitirá valorar la consecución del resultado R5.

La evaluación de los alumnos va a realizarse de forma continua, teniendo en cuenta que el bloque B1 contribuye 3p, los bloques B2-B3 2.5p cada uno y el bloque P 2p. Para evaluar los bloques teóricos, tras la impartición de los temas correspondientes se realizará un control que tendrá una nota B_i ($1 \leq i \leq 3$). La nota del bloque práctico P se calculará en función de las prácticas de laboratorio realizadas por cada estudiante. De esta forma, tras la realización de los controles y de las prácticas cada alumno podrá obtener una nota $N = B_1 + B_2 + B_3 + P$ entre 0 y 10. Aquellos alumnos con una nota N mayor o igual a 5 y que en cada uno de los bloques B1 a B3 hayan sacado como mínimo un 30% de la puntuación en cada caso (esto es, 0.9p para B1, 0.75p para B2 y B3) aprobarán, obteniendo como nota final $\min(10, N + A)$ donde A corresponde a los posibles puntos adicionales por participación en clase (véase más abajo).

En caso de que el alumno no haya podido seguir la asignatura a través del proceso de evaluación continua (en particular en el caso de estudiantes a tiempo parcial o deportistas reconocidos de alto nivel que por su casuística personal no hayan podido completar las actividades del curso) se realizará un examen ordinario al final del curso que permitirá compensar los exámenes parciales de los bloques B1-B2 (el parcial del bloque B3 se realiza en cualquier caso como parte de este examen final). El examen de la convocatoria ordinaria de la asignatura comprenderá 3 partes, una por cada bloque teórico. Cualquier alumno podrá presentarse a cualquiera de los bloques teóricos en el examen ordinario. En caso de que alguno de los bloques B1-B2 se realice dos veces (durante el curso y en el examen final ordinario), la nota del mismo será la mayor de las dos notas obtenidas. En el examen de febrero, no hay parte práctica (la nota es la de las prácticas de laboratorio desarrolladas durante el curso, cuya realización no exige presencialidad completa).

Durante el curso se plantearán diversas prácticas, ejercicios y cuestiones de carácter voluntario. Mediante su realización se podrán obtener hasta 1.6 puntos adicionales, que se sumarán a la nota de cada estudiante, una vez que haya aprobado la asignatura.

Las calificaciones de cada uno de los bloques y de la parte práctica se guardan para el examen de septiembre, cuya estructura será exactamente la misma que la del examen de febrero. Tanto en esta segunda convocatoria como en las extraordinarias los alumnos podrán obtener un 100% de la calificación máxima realizando el examen de manera íntegra.

BIBLIOGRAFÍA Y OTROS RECURSOS

Básica

- Algorithms; Robert Sedgewick y Kevin Wayne; Addison-Wesley; 2011
- Algorithms; Sanjoy Dasgupta, Christos Papadimitriou y Umesh Vazirani; McGraw Hill; 2008
- Especificación, Derivación y Análisis de Algoritmos; Narciso Martí Oliet; Prentice Hall; 2006
- Fundamentos de Algoritmia; Gilles Brassard y Paul Bratley; Pearson Prentice-Hall; 1997
- Introduction to Algorithms; T.H. Cormen, C.E. Leiserson, R.L. Rivest y C. Stein; McGraw-Hill; 2010



Introduction to the Design and Analysis of Algorithms; Anany V. Levitin; Wiley; 2007

Complementaria

Algorithm Design; J. Kleinberg y E. Tardos; Pearson; 2006

Data Structures and Algorithm Analysis in Java; M. A. Weiss; Pearson; 2007

Estructura de datos y métodos algorítmicos; Narciso Martí Oliet, Yolanda Ortega Mallén y José Alberto Verdejo López; Prentice Hall; 2003

Fundamentals of computer algorithms; E. Horowitz y S. Sahni; Computer Science Press; 1978

Programación Dinámica. Introducción y Ejercicios Resueltos; C. Cotta; UMA Editorial; 2018

The Science of Programming; Gries; Springer; 1981

DISTRIBUCIÓN DEL TRABAJO DEL ESTUDIANTE

ACTIVIDAD FORMATIVA PRESENCIAL

Descripción	Horas	Grupo grande	Grupos reducidos
Lección magistral	41,4	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Prácticas en laboratorio	18,6	<input type="checkbox"/>	<input checked="" type="checkbox"/>

TOTAL HORAS ACTIVIDAD FORMATIVA PRESENCIAL 60

ACTIVIDAD FORMATIVA NO PRESENCIAL

Descripción	Horas
Resolución de problemas	45
Estudio personal	30

TOTAL HORAS ACTIVIDAD FORMATIVA NO PRESENCIAL 75

TOTAL HORAS ACTIVIDAD EVALUACIÓN 15

TOTAL HORAS DE TRABAJO DEL ESTUDIANTE 150

