



Guía docente				
Datos Identificativos			2014/15	
Asignatura (*)	Algoritmos	Código	614G01011	
Titulación	Grao en Enxeñaría Informática			
Descritores				
Ciclo	Periodo	Curso	Tipo	Créditos
Grado	1º cuatrimestre	Segundo	Obligatoria	6
Idioma	Castellano			
Prerrequisitos				
Departamento	Computación			
Coordinador/a	Valderruten Vidal, Alberto	Correo electrónico	alberto.valderruten@udc.es	
Profesorado	Blanco Ferro, Antonio angel Fontenla Romero, Oscar Gómez Rodríguez, Carlos Hernandez Pereira, Elena Maria Jorge Castro, Jose Santiago Valderruten Vidal, Alberto	Correo electrónico	antonio.blanco.ferro@udc.es oscar.fontenla@udc.es carlos.gomez@udc.es elena.hernandez@udc.es santiago.jorge@udc.es alberto.valderruten@udc.es	
Web	www.madsgroup.org/docencia/alg			
Descripción general	<p>A materia de Algoritmos permite ao estudante de Enxeñaría informática profundizar nas técnicas de deseño dos algoritmos tendo en conta factores cualitativos e cuantitativos na avaliación dos mesmos. Dunha parte completa a formación na elaboración de programas eficientes e correctamente estruturados, e doutra parte permite abordar as técnicas de deseño máis utilizadas na resolución dos problemas que pode encontrar o enxeñeiro.</p> <p>É de destacar que a realización de experimentos de medición de tempos de execución sobre os distintos programas analizados achega un enfoque empírico que adoita ser moi valorado polo estudante, que pode así constatar a interpretación concreta das complexidades atopadas. As dificultades plantexadas por algúns casos estudados permiten unha refl exión complementaria sobre aspectos como a xestión de recursos informáticos, detalles de execución de procesos, arquitecturas e sistemas operativos utilizados, etc.</p> <p>Támén é destacable o estudo e análise dun conxunto importante de algoritmos fundamentais, cubrindo un amplo espectro de técnicas algorítmicas e de as súas aplicacións. A posibilidade de usar distintas técnicas na resolución dalgúns problemas leva naturalmente a pensar nas vantaxes e inconvenientes das distintas estratexias, e na necesidade de saber elixir a que mellor se adapta a cada situación.</p> <p>Por último é importante profundizar no rigor necesario para desenvolver non só solucións que se adapten a unhas especificacións dadas, sino ademáis que o fagan de modo e ficiente dende o punto de vista dos recursos informáticos necesarios. Resulta fundamental a ilustración mediante varios casos prácticos nos que a existencia de algoritmos eficientes coñecidos leva a desechar os deseños alternativos por moi naturais que podan resultar a primeira vista.</p>			

Competencias de la titulación	
Código	Competencias de la titulación
A12	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.
A13	Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema.
B3	Capacidad de análisis y síntesis
C3	Utilizar las herramientas básicas de las tecnologías de la información y las comunicaciones (TIC) necesarias para el ejercicio de su profesión y para el aprendizaje a lo largo de su vida.

Resultados de aprendizaje			
Competencias de materia (Resultados de aprendizaje)			Competencias de la titulación
Reconocer la importancia del estudio de la complejidad de los algoritmos y saber realizar estudios empíricos para determinarla.	A12	B3	C3
	A13		



Saber aplicar las técnicas de análisis de la complejidad de los algoritmos.	A12 A13	B3	
Identificar estructuras de datos adaptadas a los algoritmos estudiados para obtener implementaciones más eficientes y robustas.	A13	B3	C3
Conocer las técnicas más utilizadas en el diseño de los algoritmos.	A12	B3	
Utilizar diferentes modelos de computación y niveles de abstracción necesarios para el diseño de algoritmos.	A12	B3	
Comprender elementos de estudio sobre la complejidad computacional.	A12 A13	B3	

Contenidos	
Tema	Subtema
<p>Tema 1</p> <p>Título del tema: Análisis de Algoritmos.</p> <p>Código: T1</p> <p>Presentación: En este primer tema se plantea el análisis de la complejidad de los algoritmos como uno de los principales objetivos del curso.</p> <p>En síntesis, se trata de añadir a los criterios que ya deben resultar familiares, los de estructuración y de corrección de los programas, el de la eficiencia de los algoritmos.</p>	<p>Unidades de contenido:</p> <ol style="list-style-type: none"> 1. Análisis de la eficiencia de los algoritmos: Notaciones asintóticas, Modelo de computación, Verificación empírica del análisis. 2. Cálculo de los tiempos de ejecución: Análisis de los casos peor y medio, Cálculo de la O, Resolución de recurrencias.
<p>Tema 2</p> <p>Título del tema: Estructuras de datos.</p> <p>Código: T2</p> <p>Presentación: En este segundo tema se propone una revisión de las estructuras de datos básicas (pilas, listas, colas, árboles, conjuntos y grafos) con el objetivo de estudiar todas las implicaciones que conlleva su uso en cuanto a las complejidades espacial y temporal.</p> <p>Igualmente se profundiza en el estudio de estructuras interesantes desde el punto de vista del tiempo de ejecución: las tablas de dispersión y los montículos, estructura ésta última a la que recurriremos más adelante cuando se trate de implementar mejoras en algoritmos de grafos y algún caso de programación dinámica. La complejidad de la operación de búsqueda puede servir como hilo conductor en buena parte de este tema.</p> <p>Conviene en una introducción de esta parte del curso el insistir en los criterios de estructuración que debemos mantener en el diseño de cualquier aplicación, motivando el uso de tipos de datos abstractos y su consiguiente implementación mediante módulos. El objetivo es dar así las líneas generales de lo que se considera la disciplina de programación que debe exigirse al estudiante para la realización de las prácticas.</p>	<p>Unidades de contenido:</p> <ol style="list-style-type: none"> 1. Pilas, colas, listas. 2. Árboles, montículos. 3. Dispersión (hashing). 4. Conjuntos disjuntos. 5. Grafos (representación).



<p>Tema 3</p> <p>Título del tema: Algoritmos sobre secuencias y conjuntos de datos</p> <p>Código: T3</p> <p>Presentación: El problema de la ordenación de una secuencia de elementos se convierte en esta parte del curso en una excusa ideal tanto para estudiar la complejidad de varios tipos de algoritmos como para presentar diferentes estrategias de diseño de algoritmos que se pueden extrapolar para la resolución de otros problemas.</p> <p>Uno de los algoritmos a los que se le dedicará especial atención es a la ordenación rápida, ya que permite introducir la característica fundamental de los algoritmos aleatorios, que se comportan de forma distinta al ejecutarse con una misma entrada. Una consecuencia directa es que el calificativo de peor caso o mejor caso para una entrada deja de tener sentido, aspecto que es importante debatir en clase.</p>	<p>Unidades de contenido:</p> <ol style="list-style-type: none">1. Algoritmos de búsqueda.2. Algoritmos de ordenación: Inserción, Shell, Montículos (heapsort), Fusión (mergesort), Ordenación Rápida (quicksort).3. Algoritmos aleatorios.
<p>Tema 4</p> <p>Título del tema: Algoritmos voraces</p> <p>Código: T4</p> <p>Presentación: En este tema se estudian algoritmos ávidos o voraces. Una vez explicada la técnica a través de sus características generales que presentaremos con la ayuda de algún ejemplo, se estudiarán los algoritmos más representativos de esta categoría: los algoritmos de grafos, una solución al problema de la mochila y algún problema de planificación de tareas.</p>	<p>Unidades de contenido:</p> <ol style="list-style-type: none">1. Problema de la mochila.2. Algoritmos de grafos: Ordenación Topológica, Árbol de recubrimiento mínimo, Caminos mínimos.3. Problemas de planificación de sistemas informáticos.
<p>Tema 5</p> <p>Título del tema: Diseño de algoritmos por inducción</p> <p>Código: T5</p> <p>Presentación: En este punto, ya se habrá visto a lo largo del curso varios algoritmos que siguen la estrategia divide y vencerás: ordenación por fusión y ordenación rápida, búsqueda dicotómica, suma de la subsecuencia máxima. . . El trabajo propuesto en la primera unidad de este tema consiste básicamente en generalizar los planteamientos de dicha estrategia identificando sus distintas características en cada uno de los algoritmos propuestos.</p> <p>En la segunda unidad del tema se plantea usar una estrategia ascendente mediante la búsqueda de una solución general a partir de las soluciones de subproblemas elementales. Desde el punto de vista de la eficiencia se cuestionará el uso de técnicas descendentes como divide y vencerás en determinadas situaciones. Mientras que con la opción de la programación dinámica se buscará un compromiso que permita, cuando sea posible, una optimización de la cantidad de memoria requerida por el algoritmo.</p>	<p>Unidades de contenido:</p> <ol style="list-style-type: none">1. Divide y Vencerás.2. Programación dinámica: Principio de optimalidad, Problema de la mochila.



<p>Tema 6</p> <p>Título del tema: Exploración de grafos.</p> <p>Código: T6</p> <p>Presentación: El objetivo de este tema es el de dar una visión más amplia de las aplicaciones de los grafos en el tratamiento de problemas de diversa índole, así como la de no dejar de lado las técnicas algorítmicas ligadas al desarrollo de importantes áreas de la computación como la inteligencia artificial.</p> <p>Los algoritmos de grafos vistos en el tema de algoritmos voraces (T4) coinciden en realizar un recorrido de todos los nodos del grafo. Se insistirá entonces en cómo mejorar los tiempos de ejecución de los algoritmos que se presenten evitando un análisis exhaustivo de todos los nodos.</p>	<p>Unidades de contenido:</p> <ol style="list-style-type: none"> 1. Juegos de estrategia. 2. Recorridos. 3. Algoritmos con retroceso.
<p>Tema 7</p> <p>Título del tema: Complejidad Computacional.</p> <p>Código: T7</p> <p>Presentación: En este último tema planteamos un razonamiento sobre el conjunto de los algoritmos capaces de resolver cada tipo de problema. Hablaremos de las complejidades de los problemas, de cotas inferiores para la complejidad de los problemas y de NP-compleción, en definitiva, de las principales técnicas y conceptos que se utilizan en el estudio de la complejidad computacional.</p>	<p>Unidades de contenido:</p> <ol style="list-style-type: none"> 1. NP-Compleitud, Problemas NP-completos

Planificación			
Metodologías / pruebas	Horas presenciales	Horas no presenciales / trabajo autónomo	Horas totales
Sesión magistral	28.75	28.75	57.5
Prueba de respuesta breve	1.25	6.25	7.5
Prácticas de laboratorio	19	19	38
Trabajos tutelados	4	2	6
Solución de problemas	5	10	15
Prueba objetiva	4	20	24
Atención personalizada	2	0	2

(*Los datos que aparecen en la tabla de planificación són de carácter orientativo, considerando la heterogeneidad de los alumnos)

Metodologías	
Metodologías	Descripción
Sesión magistral	Clases magistrales en la exposición de los conocimientos teóricos utilizando diferentes recursos: la pizarra, transparencias, proyecciones, demostraciones y la facultad virtual. Puede incluir conferencia invitada.
Prueba de respuesta breve	En general consiste en la realización de ejercicios sobre la ejecución de casos sobre los algoritmos estudiados o sobre su adaptación a otras situaciones. Estas pruebas son evaluadas.
Prácticas de laboratorio	Prácticas diseñadas por el profesor basadas en los conocimientos que el estudiante va adquiriendo en las clases magistrales y que por tanto las complementan. Los estudiantes desarrollarán estos trabajos en parejas a lo largo del curso, e individualmente en una práctica final que se incluye en la prueba objetiva. Se implementarán programas que ilustren los problemas relacionados con el tema. Se pedirá el informe de resultados para su evaluación. A lo largo de las horas asignadas a cada práctica se evaluarán los informes de la práctica anterior.



Trabajos tutelados	Trabajos tutelados propuestos por el profesor y desarrollados por los estudiantes o bien en grupo o bien individualmente.
Solución de problemas	Se desarrollarán ejemplos sobre los conocimientos teóricos relacionados con el tema, y se resolverán dudas. Se evaluará individualmente la resolución de algunos problemas.
Prueba objetiva	Se evaluará el dominio de los conocimientos teóricos y operativos de la materia. Se evaluará igualmente la práctica individual final.

Atención personalizada

Metodologías	Descripción
Trabajos tutelados	Clases de problemas en grupos reducidos: Se desarrollarán ejemplos sobre los conocimientos teóricos relacionados con el tema, y se resolverán dudas.
Prácticas de laboratorio	Trabajos tutelados bien individuales bien en grupo sobre algún aspecto del tema a estudiar. Son controlados por parte del profesor mediante tutorías en grupo y controles de evaluación.
Solución de problemas	Prácticas de aula de informática: Se implementarán programas que ilustren los problemas relacionados con el tema. Se pedirán el informe de resultados para su evaluación.

Evaluación

Metodologías	Descripción	Calificación
Prueba de respuesta breve	3 pruebas escritas de evaluación continua, en las que se evaluará el dominio de los contenidos de los trabajos académicos a revisar. Se realizarán durante las horas de teoría y se anunciarán con al menos 7 días de antelación.	15
Prácticas de laboratorio	4 prácticas realizadas en parejas, en las que se evaluará: estructuración de los programas, calidad de la documentación, claridad, adecuación y explicación de los resultados. La entrega en tiempo y forma de las prácticas es condición necesaria para acceder al examen individual final de prácticas de la 1ª oportunidad. La evaluación se realiza mediante tutorías de seguimiento de las prácticas, durante las horas de práctica.	10
Solución de problemas	Se desarrollarán ejemplos sobre los conocimientos teóricos relacionados con el tema, y se resolverán dudas. Incluye evaluación de 3 trabajos. Se realizarán durante las horas de Trabajo en Grupo Reducido (Seminarios), pudiendo en algún caso completarse en horas no presenciales.	15
Prueba objetiva	Se evaluará el dominio de los conocimientos teóricos y operativos de la materia. Examen individual de teoría (2h): 40% Examen individual de prácticas (2h): 20% Para ser convocada/o al examen de prácticas de la 1ª oportunidad, es necesaria la entrega en plazo de todas las prácticas de laboratorio.	60
Otros		

Observaciones evaluación



En la 2ª oportunidad el estudiante podrá presentarse nuevamente tanto al examen de prácticas como al examen teórico (partes previstas en la prueba objetiva).

El examen individual de prácticas (prueba objetiva) tendrá lugar en el mismo día fijado para el examen de la asignatura y podrán establecerse distintos turnos dependiendo del número de estudiantes matriculados; es necesario disponer, previamente, del material de todas las prácticas realizadas a lo largo del curso en la cuenta de usuario de cada estudiante.

Presentarse al examen de prácticas o al examen de teoría supondrá una calificación distinta al "no presentado" en el acta correspondiente.

Según lo previsto por la Norma que regula el régimen de dedicación al estudio de los estudiantes de Grado de la UDC, los estudiantes podrán optar por una dedicación a tiempo parcial. Su implantación en el ámbito de esta asignatura supondrá que, de forma general, la calificación que figurará en el acta será la mejor entre la obtenida según el criterio especificado en esta sección de la guía docente, y la obtenida únicamente con la prueba objetiva ponderando en 2/3 partes el examen de teoría y 1/3 parte el examen de prácticas.

En la oportunidad adelantada de diciembre el 100% de la evaluación corresponderá a un examen específico de teoría que incluirá cuestiones relacionadas con las prácticas.

Fuentes de información

Básica	<ul style="list-style-type: none">- M. A. Weiss (1995). Estructuras de Datos y Algoritmos. Addison Wesley- G. Brassard y P. Bratley (1997). Fundamentos de Algoritmia. Prentice Hall- U. Manber (1989). Introduction to Algorithms - A Creative Approach. Addison Wesley
Complementaria	<ul style="list-style-type: none">- R. Sedgewick (1988). Algorithms. Addison Wesley- R. Peña Marí (2005). Diseño de Programas. Formalismo y Abstracción. Tercera edición.. Pearson Prentice Hall- B. W. Kernighan y D. M. Ritchie (1991). El lenguaje de programación C, 2ª edición. Prentice Hall- T. H. Cormen, C. E. Leiserson y R. L. Rivest (1990). Introduction to Algorithms. MIT Press

Recomendaciones

Asignaturas que se recomienda haber cursado previamente

Concurrencia y Paralelismo/614G01018

Sistemas Inteligentes/614G01020

Asignaturas que se recomienda cursar simultáneamente

Asignaturas que continúan el temario

Matemática Discreta/614G01004

Programación II/614G01006

Otros comentarios

(*) La Guía Docente es el documento donde se visualiza la propuesta académica de la UDC. Este documento es público y no se puede modificar, salvo cosas excepcionales bajo la revisión del órgano competente de acuerdo a la normativa vigente que establece el proceso de elaboración de guías