

PROGRAMA DE LA ASIGNATURA

Curso académico: 2013/2014

Identificación y características de la asignatura				
Código	501273			Créditos ECTS 6
Denominación (Español)	Análisis y Diseño de Algoritmos (AyDA).			
Denominación (Inglés)	Analysis and Design of Algorithms			
Titulaciones	Grado en Ingeniería Informática en Ingeniería del Software Grado en Ingeniería Informática en Ingeniería de Computadores			
Centro	Escuela Politécnica, Cáceres			
Semestre	3º	Carácter	Obligatoria	
Módulo	Programación			
Materia	Informática			
Profesores	Despacho	Correo-e	Página web	
M ^a Luisa Durán Martín-Merás	I.2 (17)	mlduran@unex.es	http://campusvirtual.unex.es/	
Pablo García Rodríguez	Sala 3	pablogr@unex.es	gim.unex.es/pablogr	
Área de conocimiento	Lenguajes y Sistemas Informáticos			
Departamento	Ingeniería de Sistemas Informáticos y Telemáticos			
Profesora coordinadora	M ^a Luisa Durán Martín-Merás			
Competencias				
Competencias básicas del título:				
<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 específicas asignadas:				
<p>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.</p> <p>CI07: Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.</p>				
Resultados de aprendizaje de estas competencias:				

- Aprender y comprender el concepto de eficiencia temporal y espacial de un algoritmo.
- Capacitar al estudiante en el análisis de la eficiencia de algoritmos, tanto iterativos como recursivos.
- Capacitar al estudiante en la especificación formal de tipos abstractos de datos utilizando notaciones formales.
- Capacitar al estudiante en la verificación formal de algoritmos.
- Determinar las estructuras de datos adecuadas para almacenamiento de información en memoria principal de diferentes problemas.
- Conocer diferentes alternativas de implementación de estructuras de datos, razonando sobre la eficiencia espacial y temporal de tales estructuras.
- Familiarizar al estudiante con un conjunto de tipos abstractos de datos de frecuente aplicación, capacitando al alumno para identificar y aplicar ese conjunto de TADs de manera adecuada en una aplicación concreta.
- Conocer diversas técnicas de diseño de algoritmos y razonar sobre su aplicabilidad y adecuación a la hora de resolver un problema.
- Dotar al estudiante de capacidad analítica para comparar distintas soluciones alternativas que permiten resolver un mismo problema.
- Buscar, analizar, sintetizar y debatir 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.
- Analizar, planificar, diseñar y desarrollar 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.
- Poder utilizar de manera eficaz un entorno de programación que incluya herramientas de edición, compilación, depuración y documentación de programas.
- Aportar criterios para justificar la utilización de distintos paradigmas de programación y plataformas de desarrollo de software en un determinado contexto.
- Conocer profundamente las técnicas para modelar problemas complejos, planteando soluciones eficientes y efectivas.

Competencias transversales asignadas:

CT03: Capacidad para resolver problemas.

CT07: Capacidad de análisis y síntesis.

Resultados de aprendizaje de estas competencias:

- Reconocer la estructura de un problema, datos de entrada, incógnitas, magnitudes, condiciones iniciales, así como los pasos para su resolución.
- Desarrollar la capacidad de abstracción del alumno.
- Dotar al alumno de una sistemática de actuación ante el planteamiento de un determinado problema que se desee resolver.
- Extraer del problema las soluciones triviales, reconocer la multiplicidad de soluciones, etc...
- Saber elegir con fundamento los métodos y medios más adecuados para resolver un problema.
- Desarrollar la capacidad de observación, generalización, abstracción, razonamiento lógico, deductivo e inductivo, y síntesis.
- Identificar relaciones básicas, desagregando los fenómenos en sus partes componentes.
- Establecer relaciones causales sencillas, o identificar las ventajas y desventajas de las decisiones.
- Establecer prioridades en las tareas según su orden de importancia.
- Identificar relaciones múltiples, desglosar un problema complejo en sus partes.
- Capacitar al estudiante para establecer vínculos causales complejos.
- Analizar las relaciones entre las distintas partes de un problema o situación.
- Realizar trabajos que requieren estas destrezas: extrapolar resultados, sacar

conclusiones, sintetizar textos, ya sea de forma oral o escrita, de cara a entender especificaciones o documentar sus trabajos, etc.

- Aplicar estos métodos de forma automática en múltiples situaciones de su vida académica y profesional.

Temas y contenidos

Breve descripción del contenido

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 esquemas de diseño de algoritmos.
- Se introducen aspectos de formalización sobre la especificación de algoritmos y tipos abstractos de datos.

Temario de la asignatura

Denominación del tema 1: Verificación formal

Contenidos del tema 1:

- Introducción
- Especificación formal con pre- y post-condiciones
- Verificación a posteriori
- Verificación formal de algoritmos iterativos
- Verificación formal de algoritmos recursivos

Denominación del tema 2: Complejidad algorítmica

Contenidos del tema 2:

- Objetivo del análisis de algoritmos
- Análisis a priori y análisis a posteriori
- 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
 - Cota superior (O). Propiedades
 - Cota inferior (Ω). Propiedades
 - Orden exacto (Θ). Propiedades
- Utilidades y significado de las cotas superior e inferior
- Ordenes de Complejidad
- Medidas Frecuentes. Ejemplos
- Análisis de algoritmos de ordenación
- Análisis de algoritmos de búsqueda
- Repaso del concepto de recursividad y terminación recursiva
- Diseño de problemas utilizando recursividad
- Complejidad de algoritmos recursivos
- Ejercicios

Denominación del tema 3: Divide y vencerás

Contenidos del tema 3:

- Multiplicación de enteros muy grandes
- Búsqueda binaria
- Ordenación

- Mezcla
- Quicksort
- Búsqueda de la mediana
- Multiplicación de matrices
- Cálculo de potencias
- Ejercicios

Denominación del tema 4: Algoritmos voraces

Contenidos del tema 4:

- Características generales
- Algoritmos sobre Grafos
- Árbol de recubrimiento mínimo
- Caminos mínimos
- El problema del cambio (v.1)
- El problema de la mochila (v.1)
- Planificación de tareas
- Ejercicios

Denominación del tema 5: Esquemas de Vuelta Atrás y de Ramificación y Poda

Contenidos del tema 5:

- Características generales
- Algoritmos de exploración de grafos
- Recorrido en profundidad
- Recorrido en anchura
- Recorrido de Hamilton
- Recorrido de Euler
- El problema de la mochila (v.2)
- El problema de las n reinas
- Ejercicios
- Características generales del esquema de ramificación y poda
- El problema de la asignación de tareas
- El problema de la mochila (v.3)
- Ejercicios

Denominación del tema 6: Esquema de programación dinámica

Contenidos del tema 6:

- Características generales
- Cálculo de los coeficientes binomiales
- El problema del campeonato mundial
- El problema del cambio (v.2)
- El problema de la mochila (v.4)
- Caminos mínimos en grafos
- Multiplicación de matrices
- Warshall
- Enfoques que aplican recursión
- Funciones con memoria
- Ejercicios

Denominación del tema 7: Especificación formal

Contenidos del tema 7:

- Especificación formal de algoritmos
- Pre y Post condiciones, asertos lógicos
- Notación de Guttag
- Especificación algebraica de TADs
- Ejercicios prácticos con MAUDE
- Ejemplos

Actividades formativas

Horas de trabajo del alumno por tema		Presencial		Actividad de seguimiento	No presencial
Tema	Total	GG	SL	TP	EP
1. Verificación formal.	12,5	5	1,5	0	6
2. Complejidad algorítmica	25,5	8	1,5	0	16
3. Divide y vencerás	9	3	0	0	6
4. Algoritmos voraces	16	4	2	0	10
5. Esquema de Vuelta atrás +R&P	32	8	4	2	18
6. Programación dinámica	12,5	3	1,5	0	8
7. Especificación formal.	24,5	6	4,5	0	14
Examen	18	3	0	0	15
Evaluación del conjunto	150	40	15	2	93

GG: Grupo Grande (100 estudiantes).

SL: Seminario/Laboratorio (prácticas clínicas hospitalarias = 7 estudiantes; prácticas laboratorio o campo = 15; prácticas sala ordenador o laboratorio de idiomas = 30, clases problemas o seminarios o casos prácticos = 40).

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

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

Sistema de evaluación		
Descripción de los criterios de evaluación		
<i>Aspectos a evaluar:</i>		
1. Adquisición de conocimientos relacionados con la titulación y la asignatura		
2. Resolución de problemas		
3. Estructuración clara, concisa y estructurada de los ejercicios y trabajos a presentar		
4. Participación activa en clase resolviendo problemas planteados		
<i>Actividades e instrumentos de evaluación</i>		
Actividades semipresenciales	<ul style="list-style-type: none"> ▪ Prueba para evaluación de habilidades en la verificación y análisis de complejidad algorítmica: 25% de las actividades no recuperables (en la semana 5ª) ▪ Exposición de problemas en clase delante de los compañeros: 25% de las tareas no recuperables (en la 9ª semana) ▪ Prueba de Mooshak (Concurso de programación): 25% de las tareas no recuperables (en la semana 10ª) ▪ Prueba de Maude (especificación algebraica): 25% de las tareas no recuperables (aproximadamente en la 14ª semana) 	25% No recuperable
Examen teórico	<ul style="list-style-type: none"> ▪ Una parte de teoría, que consistirá en un examen escrito tipo test. Para poder aprobar la asignatura habrá que obtener una puntuación mínima igual o superior al 40% de la puntuación máxima de ese examen. 	25% Recuperable
Examen práctico	<ul style="list-style-type: none"> ▪ Una parte de problemas. Habrá que sacar una nota mínima de 3 puntos sobre 10 en cada problema para que se pueda calcular la nota media. El examen consistirá en 3 problemas. 	50% Recuperable

Bibliografía y otros recursos

Bibliografía de apoyo seleccionada

[Aho 88] A. Aho; J. Hopcroft; J. Ullman. *Estructuras de datos y algoritmos*. Editorial Addison_Wesley Americana, 1988.

[Franch 01] X. Franch Gutiérrez. *Estructuras de datos. Especificación, diseño e implementación, 3ª edición*. Edicions UPC, 2001.

[Horowitz 95] E. Horowitz; S. Sahni. *Fundamentals of data structures in C++*. Editorial Computer Science Press, 1995.

[Peña 98] R. Peña Martí. *Diseño de programas. Formalismo y abstracción. 2ª ed. Prentice-Hall, 1998*.

*Bibliografía o documentación de lectura obligatoria**

[Martí 04] Narciso Martí; Yolanda Ortega; José A. Verdejo. *Estructuras de datos y métodos algorítmicos. Ejercicios resueltos*. Editorial Pearson – Prentice Hall (Prentice Práctica), 2004.

[Guerequeta 00] 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/>).

[Brassard 97] G. Brassard; P. Bratley. *Fundamentos de Algoritmia*. Prentice Hall, 2000.

Material elaborado por los profesores para el desarrollo de la asignatura: Visitar la página web de la asignatura en CVUEX.

Horario de tutorías

Tutorías programadas:

El estudiante recibirá 2 horas de tutorías programadas en grupo a lo largo de todo el cuatrimestre (en la 7ª semana de 15)

El horario se publicará con antelación suficiente en el calendario del curso.

Tutorías de libre acceso:

Se publicarán las tutorías de cada profesor en sus respectivos despachos, en el aula virtual de la asignatura y en la web del centro.

También se resolverán dudas en los espacios de comunicación del aula virtual de la asignatura.

Recomendaciones

- Tener superadas las asignaturas de programación de los cuatrimestres anteriores.
- Seguir el plan de trabajo marcado, prestando especial atención a la resolución de problemas.
- Asistir a clase, tanto de teoría como de laboratorio, y realizar las tareas necesarias para el seguimiento de las mismas, en tiempo y forma.
- El acceso regular y continuado al aula virtual de la asignatura, la participación activa en los foros y la realización de las actividades propuestas durante el curso.
- Acudir a las tutorías del profesorado para resolver las dudas.