

PROGRAMA DE LA ASIGNATURA

Curso académico: 2017/2018

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 de Computadores Grado en Ingeniería Informática en Ingeniería del Software		
Centro	Escuela Politécnica		
Semestre	3º	Carácter	Obligatoria
Módulo	Común a la rama de Informática		
Materia	Programación		
Profesor/es			
Nombre	Despacho	Correo-e	Página web
M ^a Luisa Durán Martín-Merás	I.2 (17)	mlduan@unex.es	https://goo.gl/ayVwsk
Pablo García Rodríguez	Dirección del centro	pablogr@unex.es	https://goo.gl/dP1NhX
Alberto Gómez Mancha	I.2 (17)	agomez@unex.es	https://goo.gl/tY513e
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</p>			

datos más adecuados a la resolución de un problema.
Competencias transversales asignadas:
CT03: Capacidad para resolver problemas. CT07: Capacidad de análisis y síntesis.
Contenidos
Breve descripción del contenido
<p>Fundamentos teóricos de programación y lenguajes de programación. 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. Técnicas de verificación y validación de programas.</p> <p>La materia comprende tres partes; cada una de ellas se desarrolla tanto en el aspecto teórico como en el práctico:</p> <ul style="list-style-type: none"> • 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
<p><u>Tema 1: Verificación formal</u></p> <ul style="list-style-type: none"> • Introducción • Especificación formal con pre- y post-condiciones • Verificación a posteriori • Verificación formal de algoritmos iterativos • Verificación formal de algoritmos recursivos
<p><u>Tema 2: Complejidad algorítmica</u></p> <ul style="list-style-type: none"> • 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 superior e inferior <ul style="list-style-type: none"> ▪ 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. • Ejercicios
<p><u>Tema 3: Estructuras de datos</u></p> <ul style="list-style-type: none"> ▪ Tablas ▪ Cola de prioridad ▪ Árboles y montículos ▪ Grafos <p>Para todas ellas:</p> <ul style="list-style-type: none"> ○ Descripción ○ Implementaciones y sus métodos ○ Ejercicios

Tema 4: Esquema de programación dinámica

- 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.
- Ejercicios

Tema 5: Esquemas de Vuelta Atrás y de Ramificación y Poda

- 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
- Ejercicios
- Características generales del esquema de ramificación y poda
- El problema de la asignación de tareas
- El problema de la mochila
- Ejercicios

Tema 6: Algoritmos voraces

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

Tema 7: Divide y vencerás

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

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	24,5	7	1,5	0	16
3. Estructuras de datos	26	9	3	0	14
4. Programación dinámica	19	4	3	0	12
5. Vuelta atrás +R&P	29	6	3	2	18
6. Algoritmos voraces	10,5	3	1,5	0	6
7. Divide y vencerás	10,5	3	1,5	0	6
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.

Metodologías docentes
<p>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.</p> <p>Cuando trabajamos en "Grupo Pequeño" se seguirá una metodología de <i>aprendizaje Basado en Problemas</i> basándose sobre todo en la discusión y debate por parte de los alumnos acerca de diferentes soluciones para un mismo problema.</p> <p>Además se dispone del <i>Aula Virtual</i> donde también es posible abrir diferentes foros de discusión para diferentes problemas donde se aplican de forma práctica los conceptos teóricos de la asignatura.</p> <p>Los alumnos que no alcanzan los requisitos pueden someter a debate las soluciones de problemas con el profesor utilizando las horas de tutorías tradicionales.</p>
Resultados de aprendizaje
<p>Resultados de aprendizaje que se corresponden con la memoria verificada del título:</p> <ul style="list-style-type: none"> • Justificar la utilización de distintos paradigmas de programación y plataformas de desarrollo de software en un determinado contexto. • Buscar, analizar, sintetizar y criticar 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.

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

- Verificar formalmente algoritmos. Demostrar que son matemáticamente correctos.
- 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.

Sistema de evaluación

Descripción de los criterios de evaluación

Aspectos a evaluar:

1. Adquisición de conocimientos relacionados con esquemas de programación y con análisis formal 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

Esta asignatura puede superarse siguiendo el sistema de evaluación continua o mediante una prueba final de carácter global. El estudiante debe indicar formalmente al profesor su opción durante las tres primeras semanas del semestre mediante el mecanismo que se indicará a principio de curso. Si un estudiante no comunica el tipo de evaluación elegido en el plazo indicado se supondrá que opta por la evaluación continua.

A continuación se detallan las características de ambos sistemas de evaluación.

Evaluación continua:

- La puntuación de cada bloque se calculará sobre 10.

Actividades semipresenciales	Se realizarán diversas pruebas a lo largo del cuatrimestre, 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). Esta nota se guardará para todas las convocatorias del curso.	25% No recuperable
Examen	Estará compuesto por: <ul style="list-style-type: none"> • Test: Para poder aprobar la asignatura habrá que obtener una puntuación mínima igual o superior a 4 sobre 10. • Problemas: Habrá que sacar una nota mínima de 3 puntos sobre 10 en cada problema para que se pueda calcular la nota media 	75% Recuperable
<p>A mediados del semestre se realizará un examen parcial del mismo tipo que el examen final sobre los contenidos de los tres primeros temas de la asignatura. Los estudiantes que obtengan una nota igual o superior a 5 en el examen parcial estarán exentos de hacer una parte del examen final (test y problemas). El peso del parcial será del 35% de la nota final. La nota del parcial se guardará para todas las convocatorias del mismo curso académico.</p>		

Prueba final global:

- Para superar esta asignatura con la prueba final global deben superarse los requisitos mínimos de las dos partes de las que constará el examen de la convocatoria: Laboratorio y Examen.
- La puntuación de cada parte se calculará sobre 10.
- La nota de una parte superada no se guardará para ninguna convocatoria posterior del curso.

Laboratorio	Se realizarán diversas pruebas en un examen en un laboratorio.	25%
Examen	Estará compuesto por: <ul style="list-style-type: none"> • Test: Para poder aprobar la asignatura habrá que obtener una puntuación mínima igual o superior a 4 sobre 10. • Problemas: Habrá que sacar una nota mínima de 3 puntos sobre 10 en cada problema para que se pueda calcular la nota media 	75%

La parte denominada Examen será la misma para todos los estudiantes, independientemente del sistema de evaluación que hayan seleccionado.

Bibliografía y otros recursos

*Bibliografía o documentación de lectura obligatoria**

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

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

Bibliografía de apoyo seleccionada

- **[Cormen]** T.H. Cormen, C.E. Leiserson, et als. *Introduction to Algorithms*. Ed. M.I.T. Press McGraw Hill, 2001.
- **[Aho 88]** A. Aho; J. Hopcroft; J. Ullman. *Estructuras de datos y algoritmos*. Editorial Addison_Wesley Americana, 1988.
- **[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.

Horario de tutorías

Tutorías programadas:

El estudiante recibirá 2 horas de tutorías programadas en grupo a lo largo de todo el cuatrimestre.

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.