



Estructuras de datos y algoritmos

Guía de Aprendizaje – Información al estudiante

Datos Descriptivos

Asignatura	Estructuras de Datos y Algoritmos
Materia	Programación
Departamento responsable	LSIIS
Créditos ECTS	6
Carácter	Obligatoria
Titulación	Grado de Ingeniería Informática por la Universidad Politécnica de Madrid
Curso	Segundo
Especialidad	No aplica

Curso académico	2009-2010
Semestre en que se imparte	Ambos (septiembre a enero y febrero a junio)
Semestre principal	Primero
Idioma en que se imparte	Español
Página Web	http://lml.ls.fi.upm.es/aed



Profesorado

NOMBRE Y APELLIDO	DESPACHO	Correo electrónico
Manuel Carro Liñares (Coord.)	D-2303	mcarro@fi.upm.es
Marta Patiño	D-2311	mpatino@fi.upm.es
Ricardo Jiménez Peris	D-2311	mpatino@fi.upm.es
Adriana Toni Delgado	D-2304	atoni@fi.upm.es
José Crespo del Arco	D-2310	jcrespo@fi.upm.es
Lars-Ake Fredlund	D-2309	lfredlund@fi.upm.es
Germán Puebla Sánchez	D-2305	mcollado@fi.upm.es
Tonghong Li	D-2312	tonghong@fi.upm.es
Pablo Chico de Guzmán Huerta	L-3301	pchico@clip.dia.fi.upm.es

Nota: se espera la incorporación de un profesor más, presumiblemente al principio del semestre, para suplir la vacante que dejará la profesora Adriana Toni.

Conocimientos previos requeridos para poder seguir con normalidad la asignatura

Asignaturas superadas	<ul style="list-style-type: none">• Programación I• Programación II
Otros resultados de aprendizaje necesarios	<ul style="list-style-type: none">• Modelizar matemáticamente problemas reales y aplicar técnicas de matemática discreta para resolverlos.



Objetivos de Aprendizaje

COMPETENCIAS ASIGNADAS A LA ASIGNATURA Y SU NIVEL DE ADQUISICIÓN		
Código	Competencia	Nivel
CE-6	Comprender intelectualmente el papel central que tienen los algoritmos y las estructuras de datos, así como una apreciación del mismo.	4
CE-8	Poseer destrezas fundamentales de la programación que permitan la implementación de los algoritmos y las estructuras de datos en el software.	4
CE-9	Poseer las destrezas que se requieren para diseñar e implementar unidades estructurales mayores que utilizan los algoritmos y las estructuras de datos, así como las interfaces por las que se comunican estas unidades.	4
CE-19	Comprensión de la complejidad de los problemas informáticos y la viabilidad de su solución	2

LEYENDA: Nivel de adquisición 1: Conocimiento
Nivel de adquisición 2: Comprensión
Nivel de adquisición 3: Aplicación
Nivel de adquisición 4: Análisis y síntesis



RESULTADOS DE APRENDIZAJE DE LA ASIGNATURA			
Código	Resultado de aprendizaje	Competencias asociadas	Nivel de adquisición
RA1	Usar y definir estructuras de datos eficientes y adecuadas a cada problema.	CE-6, CE-8, CE-9	4
RA2	Resolver problemas algorítmicos no triviales.	CE-6, CE-8, CE-9	4
RA3	Razonar sobre la complejidad algorítmica.	CE-19	2
RA4	Razonar sobre la terminación.	<u>CE-19</u>	2



Sistema de evaluación de la asignatura

INDICADORES DE LOGRO		
Ref	Indicador	Relacionado con RA
I1	Sabe elegir, entre varias opciones, la estructura de datos más adecuada para resolver un problema.	RA1
I2	Comprende las implicaciones prácticas de los diferentes niveles de complejidad.	RA3
I3	Calcula la complejidad asintótica del código.	RA3
I4	Es capaz de proporcionar argumentos para demostrar la terminación de un programa.	RA4
I5	Conoce y utiliza las colecciones estándar de Java.	RA1
I6	Es capaz de implementar estructuras de datos relativamente avanzadas.	RA1
I7	Utiliza las implementaciones de las estructuras de datos vistas en la asignatura.	RA1
I8	Identifica situaciones en que es aplicable el divide y vencerás y genera código apropiado.	RA2
I9	Identifica situaciones en que son aplicables los algoritmos voraces y genera código apropiado.	RA2
I10	Identifica situaciones en que es aplicable la búsqueda con vuelta atrás y genera código apropiado.	RA2
I11	Identifica el tipo de fichero más apropiado para utilizar en diferentes situaciones.	RA1
I12	Es capaz de trabajar con ficheros binarios, tanto de datos simples como de objetos.	RA1



POLITÉCNICA



UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA
Campus de Montegancedo
Boadilla del Monte. 28660 Madrid

EVALUACION SUMATIVA			
Breve descripción de las actividades evaluables	Momento	Lugar	Peso en la calif.
Tests / pruebas escritas	Tres veces durante el cuatrimestre	Aula de clase o informática	40.00 %
Ejercicios a realizar en aula informática	Casi todas las semanas	Aula informática	40.00 %
Trabajos prácticos a realizar individualmente	Una durante cuatrimestre	En casa o en aulas de ordenadores	20.00 %
			Total: 100%

CRITERIOS DE CALIFICACIÓN



CRITERIOS DE CALIFICACIÓN

Criterios comunes a todos los regímenes

En los casos en los que se requiera a entrega de ejercicios de aula informática, trabajos prácticos o asistencia a pruebas escritas durante el período lectivo, dichas entregas y asistencia serán condición necesaria para aprobar la asignatura, salvo causa debidamente justificada y ajena a la voluntad de alumno. En caso de no entrega de un trabajo / ejercicio o no asistencia a una prueba escrita, la nota de la asignatura será **0**. Si se han entregado todos los ejercicios de aula informática, trabajos prácticos y se ha asistido a todas las pruebas evaluables, la nota de la asignatura se calculará como la media ponderada de las pruebas según los pesos establecidos en la tabla de evaluación sumativa.

Todas las pruebas y ejercicios son liberatorios de la parte evaluada y no se repetirán dentro de la misma convocatoria pruebas de recuperación, ya que no se requiere una nota mínima o de corte para superar la asignatura en ninguna de las pruebas.

Salvo que las normas vigentes de la UPM y de la Facultad de Informática establezcan lo contrario, las notas de teoría, trabajo práctico y aula informática no se guardarán entre convocatorias.

Alumnos en régimen de evaluación continua

Se realizarán tres tests (pruebas escritas cortas y/o de respuestas múltiples) a lo largo del curso, preferiblemente en los horarios destinados a pruebas evaluables, pero recurriendo si es necesario a los horarios destinados a aula informática y a la semana final de pruebas de evaluación. Se prevé realizar 11 sesiones de aula informática en las que se propondrán problemas para entregar individualmente durante la clase en el aula informática, si bien se podrán entregar tras la finalización de la misma con una penalización en la nota (ver más adelante). Se prevé asimismo realizar un trabajo práctico individual más largo que necesitará un esfuerzo repartido a lo largo de varias semanas.

Tanto el trabajo práctico como las sesiones de aula informática tendrán fechas de entrega en las que se puede alcanzar el 100% de la nota de las mismas, pero se admitirán entregas pasada la fecha señalada con una reducción del 20% de la nota global cada día. Pasados cinco días tras la fecha de entrega la nota máxima será un cero.

Como una excepción a la norma de entregar todos los trabajos y asistir a todas las pruebas escritas, se calificará como “no presentado” quien estando en régimen de evaluación continua no haya entregado ni el trabajo práctico, ni ningún ejercicio de aula informática, ni realizado ninguna prueba escrita.

Adicionalmente, para poder superar la asignatura mediante el régimen de evaluación continua es necesario la asistencia a al menos el 80% de las clases de teoría (salvo, de



CRITERIOS DE CALIFICACIÓN

nuevo, causas debidamente justificadas no achacables al alumno), para lo cual se establecerán métodos de control de asistencia. El no cumplimiento de este requisito conllevará una nota de **0** en la asignatura.

Alumnos que optan por asistir a la prueba de junio

El Sistema de evaluación mediante sólo prueba final sólo se ofrecerá si así lo exige la Normativa Reguladora de los Sistemas de Evaluación en la UPM que esté vigente en el curso académico 2010-2011, y el procedimiento para optar por este sistema estará sujeto a lo que establezca en su caso Jefatura de Estudios de conformidad con lo que estipule dicha Normativa.

Los alumnos que elijan asistir a la prueba de junio están exentos de asistencia a clase y no deben realizar los tests que se impartirán a lo largo del curso. Sin embargo sí deben entregar los ejercicios correspondientes a las sesiones de aula informática (a los que podrán acceder mediante el Aula Virtual implementada sobre Moodle) y el trabajo práctico individual, todos ellos en las mismas fechas que sus compañeros que siguen el régimen de evaluación continua. El examen de junio consistirá en una prueba escrita que ratifique el mismo nivel de conocimientos que se han comprobado mediante las tres pruebas realizadas en el curso por los alumnos de evaluación continua.

Alumnos que asisten a la prueba de julio

Los alumnos que asistan únicamente a la prueba de julio deben realizar la entrega de un trabajo práctico similar al que realizaron sus compañeros durante el curso (evaluado con el mismo peso), cuya fecha y contenidos se harán públicos con suficiente antelación. El examen de julio consistirá en la realización de un ejercicio práctico en aula informática y una prueba escrita, ambos con el peso asignado al conjunto de las pruebas escritas en la evaluación sumativa.

Alumnos que hayan cambiado de plan de estudios

En cualquiera de los regímenes anteriores, y en concordancia con lo que se había venido haciendo en el plan de estudios Plan 1996, los alumnos que tuviesen aprobado el proyecto de Estructuras de Datos II y lo hubiesen realizado en Java no necesitan entregar los ejercicios de aula informática ni los trabajos prácticos, teniéndolas convalidadas con la nota que hubieran obtenido en el proyecto de Estructuras de Datos II. No obstante, recomendamos encarecidamente que quienes estén en ese caso realicen de forma íntegra la asignatura de Algoritmos y Estructura de Datos.



Contenidos y Actividades de Aprendizaje

CONTENIDOS ESPECÍFICOS		
Bloque / Tema / Capítulo	Apartado	Indicadores Relacionados
Tema 1: Colecciones e iteradores.	1.1 Listas implementadas con vectores.	I-1, I-5
	1.2 Complejidad.	I-2, I-3, I-4
	1.3 Cadenas doblemente enlazadas.	I-1, I-5
	1.4 Árboles generales.	I-1
	1.5 Árboles binarios. Implementación y recorridos.	I-1, I-6
	1.6 Colecciones de Java. Iteradores.	I-1, I-4, I-4
Tema 2: Tipos de datos con manejo de prioridades.	2.1 Colas con prioridad.	I-1, I-5
	2.2 Comparadores.	I-4
	2.3 Colas con prioridad y ordenación.	I-1, I-3, I-4
	2.4 Montículos e implementación mediante vectores.	I-1, I-6, I-5
	2.5 Complejidad	I-2, I-3
Tema 3: Implementación de funciones actualizables.	3.1 Tablas y diccionarios.	I-1, I-5
	3.2 Funciones "hash". Colisiones.	I-1, I-5
	3.3 Implementación de tablas "hash". Complejidad.	I-1, I-2, I-3, I-6
	3.4 Árboles binarios de búsqueda. Implementación y uso. Complejidad.	I-1, I-2, I-3, I-6
	3.5 Árboles AVL. Complejidad.	I-1, I-2, I-3, I-6
	3.6 Árboles multicamino. Árboles RB.	I-1, I-6
	3.7 Árboles B y memoria secundaria. Aplicación a bases de datos. Complejidad.	I-1, I-2, I-3, I-6, I-11, I-12
Tema 4: Algoritmos específicos y diseño.	4.1 Algoritmos de ordenación.	I-1, I-2, I-3, I-4
	4.2 Técnicas básicas de diseño de <i>software</i> .	I-1, I-4, I-8, I-9, I-10

Breve descripción de las modalidades organizativas utilizadas y de los métodos de enseñanza empleados

Tabla 7. Modalidades organizativas de la enseñanza












MODALIDADES ORGANIZATIVAS DE LA ENSEÑANZA		
Escenario	Modalidad	Finalidad
	Clases Teóricas	<i>Hablar a los estudiantes</i>
	Seminarios-Talleres	<i>Construir conocimiento a través de la interacción y la actividad de los estudiantes</i>
	Clases Prácticas	<i>Mostrar a los estudiantes cómo deben actuar</i>
	Prácticas Externas	<i>Completar la formación de los alumnos en un contexto profesional</i>
	Tutorías	<i>Atención personalizada a los estudiantes</i>
	Trabajo en grupo	<i>Hacer que los estudiantes aprendan entre ellos</i>
	Trabajo autónomo	<i>Desarrollar la capacidad de autoaprendizaje</i>

Tabla 9. Métodos de enseñanza

MÉTODOS DE ENSEÑANZA		
	Método	Finalidad
	Método Expositivo/Lección Magistral	Transmitir conocimientos y activar procesos cognitivos en el estudiante
	Estudio de Casos	Adquisición de aprendizajes mediante el análisis de casos reales o simulados
	Resolución de Ejercicios y Problemas	Ejercitar, ensayar y poner en práctica los conocimientos previos
	Aprendizaje Basado en Problemas (ABP)	Desarrollar aprendizajes activos a través de la resolución de problemas
	Aprendizaje orientado a Proyectos	Realización de un proyecto para la resolución de un problema, aplicando habilidades y conocimientos adquiridos
	Aprendizaje Cooperativo	Desarrollar aprendizajes activos y significativos de forma cooperativa
	Contrato de Aprendizaje	Desarrollar el aprendizaje autónomo

Se conoce como método expositivo "la presentación de un tema lógicamente estructurado con la finalidad de facilitar información organizada siguiendo criterios adecuados a la finalidad pretendida". Esta metodología -también conocida como lección (*lecture*)- se centra fundamentalmente en la exposición verbal por parte del profesor de los contenidos sobre la materia objeto de estudio. El término "lección magistral" se suele utilizar para denominar un tipo específico de lección impartida por un profesor en ocasiones especiales.

Análisis intensivo y completo de un hecho, problema o suceso real con la finalidad de conocerlo, interpretarlo, resolverlo, generar hipótesis, contrastar datos, reflexionar, completar conocimientos, diagnosticarlo y, en ocasiones, entrenarse en los posibles procedimientos alternativos de solución.

Situaciones en las que se solicita a los estudiantes que desarrollen las soluciones adecuadas o correctas mediante la ejercitación de rutinas, la aplicación de fórmulas o algoritmos, la aplicación de procedimientos de transformación de la información disponible y la interpretación de los resultados. Se suele utilizar como complemento de la lección magistral.

Método de enseñanza-aprendizaje cuyo punto de partida es un problema que, diseñado por el profesor, el estudiante ha de resolver para desarrollar determinadas competencias previamente definidas.

Enfoque interactivo de organización del trabajo en el aula en el cual los alumnos son responsables de su aprendizaje y del de sus compañeros en una estrategia de corresponsabilidad para alcanzar metas e incentivos grupales.

Es tanto un método, a utilizar entre otros, como un enfoque global de la enseñanza, una filosofía.

Un acuerdo establecido entre el profesor y el estudiante para la consecución de unos aprendizajes a través de una propuesta de trabajo autónomo, con una supervisión por parte del profesor y durante un período determinado. En el contrato de aprendizaje es básico un acuerdo formalizado, una relación de contraprestación recíproca, una implicación personal y un marco temporal de ejecución.

Método de enseñanza-aprendizaje en el que los estudiantes llevan a cabo la realización de un proyecto en un tiempo determinado para resolver un problema o abordar una tarea mediante la planificación, diseño y realización de una serie de actividades, y todo ello a partir del desarrollo y aplicación de aprendizajes adquiridos y del uso efectivo de recursos.



POLITÉCNICA



UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA
Campus de Montegancedo
Boadilla del Monte, 28660 Madrid

BREVE DESCRIPCIÓN DE LAS MODALIDADES ORGANIZATIVAS UTILIZADAS Y METODOS DE ENSEÑANZA EMPLEADOS

CLASES DE TEORIA	Método expositivo y estudio de casos.
CLASES DE PROBLEMAS	Aprendizaje basado en problemas. Estudio de casos.
PRÁCTICAS	Aprendizaje basado en problemas.
TRABAJOS AUTONOMOS	
TRABAJOS EN GRUPO	No hay en esta asignatura.
TUTORÍAS	Atención personalizada. Estudio de casos.



Recursos didácticos

RECURSOS DIDÁCTICOS	
BIBLIOGRAFÍA	Data Structures and Algorithms in Java (Goodrich & Tamassia), ediciones 4 y 5.
RECURSOS WEB	Página web de la asignatura (http://lml.ls.fi.upm.es/aed)
	Sitio Moodle de la asignatura (http://web3.fi.upm.es/AulaVirtual/course/view.php?id=146)
	http://ww0.java4.datastructures.net/
	http://net3.datastructures.net/
EQUIPAMIENTO	Aula informática equipada con terminales para los alumnos, proyector para el profesor y pizarra.
	Aulas de clase
	Compiladores y JRE de java versión 1.6
	Entorno de desarrollo integrado (IDE) Eclipse



Cronograma de trabajo de la asignatura

Semana	Actividades en Aula	Actividades en Aula informática	Trabajo Individual	Trabajo en Grupo	Actividades de Evaluación	Otros
Semana 1 (8 horas)	<ul style="list-style-type: none"> Listas implementadas con arrays. Repaso de complejidad. (2 horas) 		<ul style="list-style-type: none"> Estudio teoría y repaso conceptos básicos anteriores (6 horas). 			
Semana 2 (9 horas)	<ul style="list-style-type: none"> Listas como nodos enlazados (1 hora). Árboles generales. (1 hora). 	<ul style="list-style-type: none"> Completar métodos de lista indexada implementada mediante un vector (2 horas). 	<ul style="list-style-type: none"> Estudio teoría y repaso problemas aula informática (5 horas) 			
Semana 3 (9 horas)	<ul style="list-style-type: none"> Colecciones de Java e iteradores. (2 horas). 	<ul style="list-style-type: none"> Uso de listas (2 horas). 	<ul style="list-style-type: none"> Estudio teoría y repaso problemas aula informática (5 horas) 			
Semana 4 (9 horas)	<ul style="list-style-type: none"> Árboles binarios. Implementación, recorridos y usos. (2 horas) 	<ul style="list-style-type: none"> Implementación de un iterador para ArrayList (2 horas). 	<ul style="list-style-type: none"> Estudio teoría y repaso problemas aula informática (5 horas) 			
Semana 5 (9 horas)	<ul style="list-style-type: none"> Colas con prioridad. Comparadores. (2 hora) 		<ul style="list-style-type: none"> Preparación prueba escrita (6 horas). 		<ul style="list-style-type: none"> Primera prueba escrita (1 hora). 	
Semana 6 (10 horas)	<ul style="list-style-type: none"> Ordenación con colas de prioridad. Complejidad. Montículos. (2 horas) 	<ul style="list-style-type: none"> Evaluar árbol de expresiones (2 horas). 	<ul style="list-style-type: none"> Estudio teoría y repaso problemas aula informática (2 horas) . Realización de trabajo práctico individual (4 horas). 			



Semana	Actividades en Aula	Actividades en Aula informática	Trabajo Individual	Trabajo en Grupo	Actividades de Evaluación	Otros
Semana 7 (10 horas)	<ul style="list-style-type: none"> Implementación de montículos con vectores. Complejidad. (2 horas) 	<ul style="list-style-type: none"> Implementar comparadores para colas con prioridad (2 horas). 	<ul style="list-style-type: none"> Estudio teoría y repaso problemas aula informática (2 horas). Realización de trabajo práctico individual (4 horas). 			
Semana 8 (10 horas)	<ul style="list-style-type: none"> Tablas y diccionarios. Funciones hash. Colisiones. (2 horas) 	<ul style="list-style-type: none"> Corregir implementación errónea de montículos (2 horas). 	<ul style="list-style-type: none"> Estudio teoría y repaso problemas aula informática (2 horas) . Realización de trabajo práctico individual (4 horas). 			
Semana 9 (10 horas)	<ul style="list-style-type: none"> Implementación de tablas hash. Complejidad. (2 horas) 	<ul style="list-style-type: none"> Implementación y comparación de funciones hash (2 horas). 	<ul style="list-style-type: none"> Estudio teoría y repaso problemas aula informática (2 horas) Realización de trabajo práctico individual (4 horas). 			
Semana 10 (10 horas)	<ul style="list-style-type: none"> Árboles binarios de búsqueda. Inserción y borrado. Recorridos. Complejidad. Implementación. (2 horas) 		<ul style="list-style-type: none"> Preparación prueba escrita (3 horas). Realización de trabajo práctico individual (4 horas). 		<ul style="list-style-type: none"> Segunda prueba escrita (1 hora). 	
Semana 11 (10 horas)	<ul style="list-style-type: none"> Árboles perfectamente equilibrados. Árboles AVL. Algoritmos. Complejidad. (2 horas) 	<ul style="list-style-type: none"> Uso de iteradores con tablas (2 horas). 	<ul style="list-style-type: none"> Estudio teoría y repaso problemas aula informática (2 horas) Realización de trabajo práctico individual (4 horas). 			
Semana 12 (10 horas)	<ul style="list-style-type: none"> Árboles multicamino. Árboles 2-3-4. Árboles RB (2 horas). 	<ul style="list-style-type: none"> Borrado en árboles mediante marcado (2 horas). 	<ul style="list-style-type: none"> Estudio teoría y repaso problemas aula informática (2 horas) Realización de trabajo práctico individual (4 horas). 			



Semana	Actividades en Aula	Actividades en Aula informática	Trabajo Individual	Trabajo en Grupo	Actividades de Evaluación	Otros
Semana 13 (10 horas)	<ul style="list-style-type: none"> Almacenamiento en memoria secundaria. Árboles B y B+ (2 horas). 	<ul style="list-style-type: none"> Dividir, mezclar y ordenar ficheros de caracteres (2 horas). 	<ul style="list-style-type: none"> Estudio teoría y repaso problemas aula informática (2 horas) Realización de trabajo práctico individual (4 horas). 			
Semana 14 (10 horas)	<ul style="list-style-type: none"> Algoritmos de ordenación. Complejidad. (2 horas) 	<ul style="list-style-type: none"> Usar expresiones regulares para <i>tokenizar</i> un fichero (2 horas). 	<ul style="list-style-type: none"> Estudio teoría y repaso problemas aula informática (2 horas). Realización de trabajo práctico individual (4 horas). 		<ul style="list-style-type: none"> Entrega trabajo práctico. 	
Semana 15 (10 horas)	<ul style="list-style-type: none"> Diseño de algoritmos y ejemplos (2 horas). 		<ul style="list-style-type: none"> Estudio teoría y repaso problemas aula informática (8 horas). 			
Semana 16 (10 horas)	<ul style="list-style-type: none"> Diseño de algoritmos y ejemplos (2 horas). 		<ul style="list-style-type: none"> Estudio teoría y repaso problemas aula informática (8 horas). 			
Semana examen (8 horas)			<ul style="list-style-type: none"> Estudio teoría (7 horas). 		<ul style="list-style-type: none"> Tercera prueba escrita (1 hora). Prueba escrita para junio (3 horas). 	

Nota: Para cada actividad se especifica la dedicación en horas que implica para el alumno.