



Grado en Ingeniería Informática

DATOS DE LA ASIGNATURA

Nombre:

Estructura de Datos II

Denominación en inglés:

Data Structures II

Código:

606010206

Carácter:

Obligatorio

Horas:**Totales****Presenciales****No presenciales****Trabajo estimado:**

150

60

90

Créditos:**Grupos reducidos****Grupos grandes****Aula estándar****Laboratorio****Prácticas de campo****Aula de informática**

3

0

0

0

3

Departamentos:**Áreas de Conocimiento:**

Tecnologías de la Información

Lenguaje y Sistemas Informáticos

Curso:**Cuatrimestre:**

2º - Segundo

Segundo cuatrimestre

DATOS DE LOS PROFESORES

Nombre:**E-Mail:****Teléfono:****Despacho:**

Aroba Paez, Javier

aroba@uhu.es

959217670

16 - Edificio Torreumbria

*Maña López, Manuel J.

manuel.mana@dti.uhu.es

959217389

Edif. Torreumbria, Dcho. 47

Domínguez Olmedo, Juan
Luis

juan.dominguez@dti.uhu.es

959217371

TUPB-61

*Profesor coordinador de la asignatura

Consultar los horarios de la asignatura

1. Descripción de contenidos**1.1. Breve descripción (en castellano):**

Abstracción de datos y genericidad. Estructura de datos y algoritmos de manipulación: árboles binarios, árboles generales, árboles equilibrados, colas de prioridad, árboles B, grafos y diccionarios.

1.2. Breve descripción (en inglés):

Generic data abstraction. Data structures and algorithms: trees, binary trees, balanced trees, priority queues, B trees, graphs and dictionaries.

2. Situación de la asignatura**2.1. Contexto dentro de la titulación:**

La asignatura está muy relacionada con tres asignaturas de primer curso: Fundamentos de Programación, Estructuras de Datos I y Fundamentos de Análisis de Algoritmos. En la primera, se estudian los tipos de datos básicos y se hace una introducción a la POO. En la segunda, se estudian el concepto de TAD, las estructuras de datos lineales, así como la gestión dinámica de memoria, necesaria para la implementación de la mayoría de las estructuras de datos avanzadas. En la tercera, se presentan los métodos para el análisis de la eficiencia de algoritmos, tan necesarios para poder elegir la representación más adecuada de un TAD, la recursividad y los esquemas algorítmicos, que se usarán en la resolución de problemas con estructuras como árboles y grafos.

En la asignatura Matemáticas II, de primer curso, se realizan estudios formales sobre teoría de grafos y análisis combinatorio, necesarios para analizar las estructuras de datos y algoritmos de manipulación de los grafos.

También resulta de interés para esta asignatura, los contenidos de Metodología de la Programación, de primer cuatrimestre del segundo curso. En ella se estudian conceptos más avanzados de POO que los alumnos tendrán la oportunidad de poner en práctica en ED II.

Por otro lado, prácticamente todas las asignaturas que se encuentran en el plan de estudios después de Estructuras de Datos II requieren que el alumno tenga una buena base en los conceptos estudiados en la asignatura. Entre ellas podemos destacar: Diseño y Desarrollo de Sistemas de Información, Programación Concurrente y Distribuida, Administración de Bases de Datos, Gestión y Recuperación de Tipos Especiales de Datos.

2.2. Recomendaciones:

Se exponen, a continuación, algunas de las competencias que deberían poseer los alumnos antes de comenzar a cursar la asignatura:

- Ser capaz de desarrollar algoritmos en forma de pseudocódigo independiente de un lenguaje final concreto
- Ser capaz de codificar de forma clara utilizando correctamente las estructuras de control
- Saber agrupar conjuntos de instrucciones algorítmicas de forma coherente mediante funciones o procedimientos
- Conocer los mecanismos de paso de parámetros y utilizarlos correctamente
- Conocer el funcionamiento de la recursividad y ser capaz de construir algoritmos recursivos
- Conocer los tipos de datos simples y ser capaz de construir estructuras de datos basadas en estos
- Conocer el concepto de TAD y saber realizar una especificación, al menos, semiformal
- Conocer el funcionamiento y utilidad de la gestión dinámica de memoria
- Conocer los tipos de datos lineales (listas, pilas y colas) y sus algoritmos de manipulación
- Conocer los principios fundamentales de la POO
- Ser capaz de implementar programas sencillos en C++

3. Objetivos (Expresados como resultados del aprendizaje):

El objetivo general que se pretende alcanzar es proporcionar al alumno la capacidad de especificar, diseñar, implementar, evaluar y utilizar estructuras de datos adecuadamente, así como, conocer e identificar los algoritmos más característicos que operan sobre dichas estructuras.

Este objetivo puede desglosarse en los siguientes objetivos específicos:

- Valorar la importancia de la programación con Tipos Abstractos de Datos (TAD) como base del diseño modular, diferenciando los conceptos de especificación e implementación de un TAD
- Utilizar la especificación algebraica como técnica de especificación formal
- Estudiar y realizar las implementaciones de los TAD a partir de su especificación, utilizando el paradigma de la POO y, en concreto, el lenguaje C++
- Conocer las estructuras de datos no lineales y los algoritmos principales que se utilizan para su manipulación
- Tener la capacidad de elección de la estructura adecuada para cada tipo de problema
- Saber decidir entre dos o más soluciones algorítmicas similares sobre la base de la eficiencia de cada una de ellas
- Aprender a combinar diferentes estructuras de datos para resolver problemas complejos

4. Competencias a adquirir por los estudiantes

4.1. Competencias específicas:

- **CC07:** Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema

4.2. Competencias básicas, generales o transversales:

- **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
- **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
- **CG0:** Capacidad de análisis y síntesis: Encontrar, analizar, criticar (razonamiento crítico), relacionar, estructurar y sintetizar información proveniente de diversas fuentes, así como integrar ideas y conocimientos.
- **G02:** Capacidad de comunicación oral y escrita en el ámbito académico y profesional con especial énfasis, en la redacción de documentación técnica
- **G03:** Capacidad para la resolución de problemas
- **G04:** Capacidad para tomar decisiones basadas en criterios objetivos (datos experimentales, científicos o de simulación disponibles) así como capacidad de argumentar y justificar lógicamente dichas decisiones, sabiendo aceptar otros puntos de vista
- **G05:** Capacidad de trabajo en equipo.
- **T02:** Conocimiento y perfeccionamiento en el ámbito de las TIC's

5. Actividades Formativas y Metodologías Docentes

5.1. Actividades formativas:

- Sesiones de Teoría sobre los contenidos del Programa.
- Sesiones de Resolución de Problemas.
- Sesiones Prácticas en Laboratorios Especializados o en Aulas de Informática.
- Actividades Académicamente Dirigidas por el Profesorado: seminarios, conferencias, desarrollo de trabajos, debates, tutorías colectivas, actividades de evaluación y autoevaluación.

5.2. Metodologías docentes:

- Clase Magistral Participativa.
- Desarrollo de Prácticas en Laboratorios Especializados o Aulas de Informática en grupos reducidos.
- Resolución de Problemas y Ejercicios Prácticos.
- Tutorías Individuales o Colectivas. Interacción directa profesorado-estudiantes.
- Evaluaciones y Exámenes.

5.3. Desarrollo y justificación:

En las clases teóricas se presentarán los conceptos de manera clara y concisa utilizando para ello las herramientas docentes más adecuadas al alcance del profesor. Para cada tema, el alumno deberá trabajar ciertos contenidos de forma personal, con ayuda del material proporcionado, estimulando, de esta forma, el aprendizaje autónomo.

Se llevarán a cabo actividades de evaluación y autoevaluación tanto en clases de teoría como de laboratorio.

En las prácticas de laboratorio se realizarán ejercicios relacionados con el temario teórico, que serán de mayor complejidad que los ejemplos vistos en clases de teoría. Se utilizará el lenguaje de programación C++. Se fomentará el aprendizaje cooperativo, inculcando además en los alumnos el sentido ético que debe primar en los estudios universitarios, de forma que eviten comportamientos fraudulentos como la copia de prácticas. Se exige al alumno de entregar memorias al finalizar cada práctica pero se exigirá la entrega de algunos ejercicios.

En este curso académico 2016/17, la asignatura cuenta con un grupo de docencia en inglés (con su propia guía docente en inglés), impartido por el profesor Juan Luis Domínguez Olmedo.

6. Temario desarrollado:

- TEMA 1 Genericidad y Abstracción de Datos
 - 1.1 Abstracción de datos y Tipos Abstractos de Datos (TAD)
 - 1.2 Tipos Genéricos. Implementación en C++
 - 1.3 Especificación algebraica de un TAD
 - 1.4 Ejemplo de contenedor simple: TAD Conjunto
- TEMA 2 Árboles
 - 2.1 Conceptos, definiciones y terminología básica
 - 2.2 Especificación e implementación del árbol binario
 - 2.3 Recorridos sobre árboles binarios
 - 2.4 Especificación e implementación del árbol general
 - 2.5 Recorridos sobre árboles generales
- TEMA 3 Árboles Binarios de Búsqueda
 - 3.1 Conceptos, implementación y recorridos
 - 3.2 Árboles AVL
 - 3.3 Árboles Rojo-Negro
- TEMA 4 Diccionarios
 - 4.1 Conceptos
 - 4.2 Especificación algebraica
 - 4.3 Implementación
 - 4.4 Implementación de diccionarios usando un ARN
 - 4.5 Tablas dispersas
 - 4.6 Implementación de diccionarios usando tablas dispersas
- TEMA 5 Grafos
 - 5.1 Conceptos
 - 5.2 Especificación algebraica
 - 5.3 Implementación
 - 5.4 Recorridos sobre grafos
 - 5.5 Ordenación topológica
 - 5.6 Caminos mínimos sobre grafos: Algoritmos de Dijkstra y Floyd
 - 5.7 Árbol de expansión de coste mínimo: Algoritmos de Prim y Kruskal
- TEMA 6 Colas de Prioridad
 - 6.1 Concepto y especificación
 - 6.2 Implementación con Secuencias
 - 6.3 Implementación con Montículos
 - 6.4 Ordenación mediante montículos: Heapsort
- TEMA 7 Árboles B
 - 7.1 Conceptos y definiciones
 - 7.2 Árboles B+

7. Bibliografía

7.1. Bibliografía básica:

- Data Structures and Algorithms in C++
Autor: Michael T. Goodrich, Roberto Tamassia, David M. Mount
Editorial: John Wiley & Sons, Inc.
Año: 2004
- TADs, Estructuras de Datos y Resolución de Problemas con C++
Autor: Larry R. Nyhoff
Editorial: Prentice Hall
Año: 2006
- Fundamentos de Estructuras de Datos. Soluciones en Ada, Java y C++
Autor: Zenón José Hernández Figueroa y otros
Editorial: Thomson
Año: 2005

7.2. Bibliografía complementaria:

- Estructuras de datos y métodos algorítmicos. Ejercicios resueltos.
Autor: Narciso Martí Oliet, Yolanda Ortega Mallén, José A. Verdejo López
Editorial: Prentice-Hall
Año: 2004
- C++ estándar. Programación con el Estándar ISO y la Biblioteca de Plantillas (STL)
Autor: E. Hernández, J. Hernández, M. C. Juan
Editorial: Paraninfo
Año: 2002

8. Sistemas y criterios de evaluación.

8.1. Sistemas de evaluación:

- Examen de teoría/problemas
- Seguimiento Individual del Estudiante
- Examen de prácticas

8.2. Criterios de evaluación y calificación:

Bloque - Aspecto	Criterio	Instrumento	Peso	
I	Conocimientos teóricos	Nivel de consecución de objetivos teóricos	Examen de la asignatura	50%
II	Conocimientos prácticos	Nivel de consecución de objetivos prácticos	Pruebas prácticas individuales	30%
III	Asistencia y participación	Asistencia y participación activa en actividades	Lista asistencia y observaciones	5%
III	Asistencia y participación	Capacidad para resolver ejercicios propuestos	Problemas de los boletines	15%

Para aprobar la asignatura, el alumno deberá aprobar por separado los bloques I y II, y además, sumar 5 puntos entre las notas de los 3 bloques. Por otra parte, para el bloque II las notas de las pruebas deben ser compensables. Debe tenerse en cuenta que el bloque III sólo se evalúa durante el cuatrimestre en el que se desarrolla la asignatura.

Los alumnos que, de forma justificada, no puedan asistir regularmente a clase, deberán realizar las 2 pruebas prácticas y el examen de la asignatura. Estos alumnos deberán entregar el justificante correspondiente durante las primeras semanas del cuatrimestre. En este caso, el peso de las pruebas prácticas será del 30% y el del examen teórico el 70%, y para aprobar la asignatura se deberán aprobar por separado cada una de las 3 pruebas.

Las notas de cada uno de los bloques aprobados se guardarán hasta la convocatoria de septiembre. Los alumnos que no hayan aprobado en junio se deben presentar a la convocatoria oficial de septiembre y realizar la parte del examen correspondiente al bloque suspenso (bloques I o II).

En la convocatoria de septiembre, el examen práctico (bloque II) se realizará por escrito el mismo día que el examen teórico. En esta convocatoria se mantiene el peso de cada uno de los bloques.

9. Organización docente semanal orientativa:

	<i>Semanas</i>	<i>Grupos Grandes</i>	<i>Grupos Reducidos Aula Estándar</i>	<i>Grupos Reducidos Aula de Informática</i>	<i>Grupos Reducidos Laboratorio</i>	<i>Grupos Reducidos prácticas de campo</i>	Pruebas y/o actividades evaluables	Contenido desarrollado
#1	2	0	2	0	0			Presentacion + Tema 1
#2	2	0	2	0	0			Tema 1 + Tema 2
#3	2	0	2	0	0			Tema 2
#4	2	0	2	0	0			Tema 2 + Tema 3
#5	2	0	2	0	0			Tema 3
#6	2	0	2	0	0			Tema 3
#7	2	0	2	0	0			Tema 4
#8	2	0	2	0	0			Tema 4
#9	2	0	2	0	0	Prueba I de Laboratorio		Tema 5
#10	2	0	2	0	0			Tema 5
#11	2	0	2	0	0			Tema 5
#12	2	0	2	0	0			Tema 5
#13	2	0	2	0	0			Tema 6
#14	2	0	2	0	0			Tema 6 - Tema 7
#15	2	0	2	0	0	Prueba II de Laboratorio		Tema 7
	30	0	30	0	0			