

ASIGNATURA ESTRUCTURAS DE DATOS NO LINEALES

Código	21714016
Titulación	GRADO EN INGENIERÍA INFORMÁTICA
Módulo	MÓDULO II - FORMACIÓN COMÚN A LA RAMA DE INF ...
Materia	MATERIA II.2 PROGRAMACIÓN DE COMPUTADORES
Curso	2
Duración	SEGUNDO SEMESTRE
Tipo	OBLIGATORIA
Idioma	CASTELLANO
Ofertable en Lengua Extranjera	NO
Movilidad Nacional	SÍ
Movilidad Internacional	SÍ
Estudiante Visitante Nacional	SÍ
ECTS	6,00
Departamento	C137 - INGENIERIA INFORMATICA

REQUISITOS Y RECOMENDACIONES

Requisitos

Para poder cursar con razonables posibilidades de éxito esta asignatura es requisito haber superado las dos asignaturas de programación de primer curso (Introducción

a la Programación y Metodología de la Programación).

También es necesario haber cursado la asignatura Análisis de Algoritmos y Estructuras de Datos (segundo curso, primer cuatrimestre), así como muy recomendable cursar la asignatura del mismo cuatrimestre Programación Orientada a Objetos.

Cualquier matrícula efectuada que incumpla estos requisitos obviamente es responsabilidad exclusiva del estudiante.

Recomendaciones

Para poder seguir la asignatura razonablemente es aconsejable:

- Conocer los aspectos relacionados con la descomposición de problemas, el diseño modular y la abstracción operacional.
- Saber especificar de manera informal los algoritmos mediante precondiciones y postcondiciones.
- Ser capaz de definir algoritmos de una manera correcta mediante el uso de estructuras de control, bucles, sentencias condicionales, etc. según convenga a la finalidad, eficiencia y claridad del código.
- Conocer los mecanismos de transferencia de parámetros y utilizarlos correctamente.
- Conocer y usar correctamente los tipos de datos básicos que ofrecen los lenguajes de programación y especialmente, los tipos estructurados: cadenas de caracteres, vectores, matrices, registros y ficheros.
- Dominar el uso de punteros y la gestión dinámica de memoria.
- Ser capaz de implementar, en un lenguaje de programación de alto nivel, programas de pequeño y mediano tamaño haciendo uso de la descomposición modular del software.
- Saber elegir y diseñar adecuadamente casos de prueba para los programas y funciones implementados.
- Distinguir y saber resolver sumas aritméticas y geométricas. Reconocer otras sumas notables: la armónica y la expresión del número e mediante serie de potencias.
- Conocer las nociones básicas de combinatoria (combinaciones y permutaciones) y los rudimentos de la probabilidad discreta: variable aleatoria discreta, noción de

probabilidad, hipótesis de equiprobabilidad, media y media ponderada.

- Conocer los mecanismos de abstracción en programación y su importancia para la resolución de problemas.
- Comprender la necesidad de separación entre los niveles de especificación, implementación y aplicación en el desarrollo de módulos software.
- Conocer los conceptos de programación basada en tipos abstractos y de reutilización de los módulos de software.
- Conocer el concepto de recursividad y sus tipos y saber implementar algoritmos recursivos para resolver problemas.
- Saber separar perfectamente los aspectos computacionales puros de los efectos colaterales como la E/S y reconocer la importancia de que la E/S, los interfaces gráficos, etc., no deben mezclarse ni ensuciar las funciones que implementan los algoritmos y las estructuras de datos.
- Saber analizar algoritmos, así como expresar, interpretar y comparar los resultados de los análisis en notación asintótica.
- Saber realizar análisis empíricos de los algoritmos, utilizando técnicas de medida del tiempo en un computador.
- Saber resolver ecuaciones de recurrencia sencillas y algunos casos notables.

OFERTA EN LENGUA EXTRANJERA

No se oferta para Lengua Extranjera.

MOVILIDAD

- Movilidad Nacional (SICUE): Sí. Tipo de enseñanza: Presencial
- Movilidad Internacional: Sí. Tipo de enseñanza: Presencial
- Estudiante Visitante Nacional: Sí. Nº Plazas: 10. Tipo de enseñanza: Presencial

RESULTADO DEL APRENDIZAJE

Id.	Resultados
1	Desarrollar programas, basándose en tipos abstractos de datos, de forma independiente de la implementación de éstos.
2	Organizar un determinado volumen de datos de la forma más racional posible en función de los requisitos del problema a resolver.
3	Implementar de diferentes formas una especificación de software dada. El alumno aprenderá a escoger entre diferentes implementaciones alternativas de una abstracción de datos, y razonar sobre la solución escogida en función de los recursos necesarios (tiempo de ejecución, espacio requerido, etc.).
4	Resolver problemas utilizando los TAD mas apropiados.

COMPETENCIAS

Tipo	Competencia
GENERAL	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
GENERAL	Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado
GENERAL	Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía
GENERAL	Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.

Tipo	Competencia
GENERAL	Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.
ESPECÍFICA	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
ESPECÍFICA	Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.

Q CONTENIDOS

Contenido	Descripción
1. Árboles. 1.1. Concepto de árbol. Definiciones básicas. 1.2. Árboles binarios. 1.2.1. Especificación de operaciones. 1.2.2. Implementación vectorial de árboles binarios. 1.2.3. Implementación mediante un vector de posiciones relativas. 1.2.4. Implementación usando celdas enlazadas. 1.3. Árboles generales. 1.3.1. Especificación de operaciones. 1.3.2. Implementación mediante listas de hijos. 1.3.3. Implementación usando celdas enlazadas. 1.4. Recorridos de árboles en profundidad y anchura. 1.5. Búsqueda. 1.5.1. Árboles binarios de búsqueda. 1.5.2. Árboles de búsqueda equilibrados. 1.5.3. Árboles B. 1.5.4. Tablas hash. 1.6. Árboles parcialmente ordenados (montículos). Colas con prioridad.	

Contenido	Descripción
<p>2. Grafos.</p> <p>2.1. Concepto de grafo. Definiciones básicas.</p> <p>2.2. Diferentes representaciones de grafos.</p> <p>2.2.1. Matriz de adyacencia y matriz de costes.</p> <p>2.2.2. Listas de adyacencia.</p> <p>2.3. Recorridos de grafos. Búsqueda.</p> <p>2.3.1. En profundidad.</p> <p>2.3.2. En anchura.</p> <p>2.4. Algoritmos de caminos de coste mínimo.</p> <p>2.4.1. Algoritmo de Dijkstra.</p> <p>2.4.2. Algoritmo de Floyd.</p> <p>2.4.3. Algoritmo de Warshall.</p> <p>2.5. Algoritmos de árboles de extensión de coste mínimo.</p> <p>2.5.1. Algoritmo de Prim.</p> <p>2.5.2. Algoritmo de Kruskal.</p>	
<p>PRÁCTICAS: Resolución de problemas de programación utilizando tipos abstractos de datos no lineales.</p> <p>Práctica 1. Problemas de árboles binarios I</p> <p>Práctica 2. Problemas de árboles binarios II</p> <p>Práctica 3. Problemas de árboles generales</p> <p>Práctica 4. Problemas de árboles binarios de búsqueda</p> <p>Práctica 5. Problemas de árboles parcialmente ordenados y otros árboles</p> <p>Práctica 6. Problemas de grafos I</p> <p>Práctica 7. Problemas de grafos II</p> <p>Práctica 8. Problemas de grafos III</p>	

SISTEMA DE EVALUACIÓN

Procedimientos de evaluación

Tarea/Actividades	Medios, técnicas e instrumentos	Ponderación
Pruebas de evaluación de resultados de actividades de aprendizaje	Tests de respuesta múltiple. Rúbricas	10 %
Pruebas escritas	Exámenes escritos teórico-prácticos. Verificación de la corrección de las soluciones aportadas por el alumno.	90 %

Criterios de evaluación

MODALIDADES DE EVALUACIÓN

=====

Para la primera convocatoria (junio) se seguirá un sistema de evaluación continua. En el resto de convocatorias, ya que no habrá docencia, se aplicará el sistema de evaluación final, incluyendo aquellas en las que el alumno solicite evaluación global.

Evaluación continua:

=====

A lo largo del curso se realizarán varios cuestionarios online a través del campus virtual, para la evaluación de los resultados de las actividades de aprendizaje (EAA).

Así mismo, durante el desarrollo de la asignatura se realizarán una o más pruebas escritas relacionadas directamente con las prácticas de la primera parte del temario (PP1).

Al finalizar el curso, los alumnos realizarán una prueba escrita adicional, de

carácter teórico-práctico, con un componente de teoría de toda la asignatura (TEO) y otro de problemas de la segunda parte (PP2).

La nota de las pruebas de problemas (PRO) será la media aritmética de PP1 y PP2, es decir,

$$\text{PRO} = (\text{PP1} + \text{PP2}) / 2.$$

La nota correspondiente a la evaluación continua (NEC) del alumno será obtenida de la siguiente forma:

SI $\text{PRO} < 3,5$

$$\text{NEC} = 0,7 * \text{PRO}$$

SI NO

$$\text{NEC} = 0,7 * \text{PRO} + 0,1 * \text{EAA} + 0,2 * \text{TEO}$$

Evaluación final:

=====

Exclusivamente en casos de fuerza mayor por causas sobrevenidas, justificadas documentalmente, que impidan al alumno presentarse a las pruebas de evaluación continua, éste podrá solicitar al profesor coordinador de la asignatura el cambio al sistema de evaluación final, en un plazo máximo de 72 horas desde que se produzca la causa del impedimento

Se realizará un examen final escrito con contenidos teóricos (TEO) y prácticos (PRO) de la asignatura, en las fechas establecidas en las convocatorias oficiales del curso académico.

La nota correspondiente a la evaluación final (NEF) será obtenida de la siguiente forma:

SI $\text{PRO} < 3,5$ ENTONCES

$$\text{NEF} = 0,7 * \text{PRO}$$

SINO

$$\text{NEF} = 0,7 * \text{PRO} + 0,3 * \text{TEO}$$

La copia total o parcial de las pruebas de evaluación, así como cualquier otro tipo de fraude detectado por los profesores, conllevará que se califique con 0 puntos a todos los implicados, sea cual fuere su papel, sin perjuicio de las acciones que la Inspección General de Servicios determine según la gravedad de la infracción.

CRITERIOS GENERALES DE EVALUACIÓN

=====

- Adecuación de la solución a la especificación del problema.
- Elección de los tipos abstractos de datos apropiados.
- Uso correcto de los TADs según su especificación.
- Elección adecuada de la implementación de cada TAD.
- Presentación, claridad y eficiencia de la solución.

PROFESORADO

Profesorado	Categoría	Coordinador
ALONSO DE LA HUERTA, JOSÉ ANTONIO	PROFESOR TITULAR ESCUELA UNIVERSITARIA	Sí
ARGUDO ARGUDO, JOSE FIDEL	PROFESOR TITULAR ESCUELA UNIV.	Sí
GARCÍA HORCAJADAS, M ^a TERESA	PROFESORA TITULAR DE ESCUELA UNIVERSITARIA	No
CABRERA SANCHEZ, JUAN FRANCISCO	PROFESOR/A SUSTITUTO/A INTERINO/A	No

ACTIVIDADES FORMATIVAS

Actividad	Horas	Detalle
01 Teoría	24	Las clases teóricas se basarán fundamentalmente en las explicaciones del profesor sobre el temario, así como en la realización de ejercicios prácticos (sobre pizarra) asociados al mismo.
02 Prácticas, seminarios y problemas	12	Se incentivará la participación activa del alumnado en las clases, realizando en grupos el desarrollo de especificaciones e implementaciones de TAD, y provocando el profesor un debate abierto sobre cada uno de los temas que se traten, motivando a los alumnos para que propongan soluciones alternativas a los problemas planteados y su posterior discusión.
03 Prácticas de informática	24	En las clases prácticas se proporcionará al alumno guiones de prácticas en los que se incluirán cuestiones teóricas y una serie de problemas de programación, que se resolverán empleando un lenguaje orientado a objetos. Los alumnos asistirán a clase con dichos guiones, que los tendrán disponibles en el campus virtual con suficiente antelación, y con los problemas planteados, de forma que en clase se discutirá en grupo la resolución de dichos problemas y el profesor explicará aquéllos que puedan plantear mayor dificultad; finalmente, cada alumno programará en el ordenador las soluciones de los problemas del guión.
10 Actividades formativas no presenciales	86,00	Estas actividades se corresponden con las horas de trabajo personal del alumno, incluyendo las horas de estudio de los contenidos teóricos y prácticos de la asignatura, así como la realización de problemas y trabajos propuestos.
12 Actividades de evaluación	4,00	Examen final

BIBLIOGRAFÍA

Bibliografía Básica

Aho, A.; Hopcroft, J.; Ullman, J.

Estructuras de datos y algoritmos. Addison-Wesley, 1988.

Alonso, J.A.; Argudo, J.F.; García, M.T. Estructuras de Datos en C. Depto. Lenguajes y Sistemas Informáticos, UCA, 2003.

Fernández-Valdivia, J.; Garrido, A.; García, M. Estructuras de datos. Un enfoque práctico usando C. 1998.

Garrido, A.; Fernández-Valdivia, J. Abstracción y Estructura de Datos en C++. Delta Publicaciones. 2013.

Heileman, G. L. Estructuras de Datos, Algoritmos y Programación Orientada a Objetos.

McGraw-Hill, 1996.

Langsam, Y; Augenstein, M. J.; Tenenbaum, A. M. Estructuras de Datos con C y C++. Prentice-Hall, 1997.

Bibliografía Específica

Hernández Figueroa, Z. J. y otros Fundamentos de Estructuras de Datos. Soluciones en Ada, Java y C++. Paraninfo, 2005.

Nyhoff, Larry R. TADs, estructuras de datos y resolución de problemas con C++. Prentice-Hall, 2006.

Standish, T.A. Data Structures, Algorithms and Software Principles in C.

Addison-Wesley, 1995.

Stroustrup, Bjarne. The C++ programming language. Special edition. Addison-Wesley. 2000.

Stroustrup, Bjarne. Programming: Principles and practice using C++. Addison-Wesley. 2008.

Weiss, M. Data Structures and Algorithm Analysis in C. Addison-Wesley, 1996.

El presente documento es propiedad de la Universidad de Cádiz y forma parte de su Sistema de Gestión de Calidad Docente.

En aplicación de la Ley 3/2007, de 22 de marzo, para la igualdad efectiva de mujeres y hombres, así como la Ley 12/2007, de 26 de noviembre, para la promoción de la igualdad de género en Andalucía, toda alusión a personas o colectivos incluida en este documento estará haciendo referencia al género gramatical neutro, incluyendo por lo tanto la posibilidad de referirse tanto a mujeres como a hombres.