



## Guía Docente

Datos Identificativos					2012/13
Asignatura (*)	Estrutura de Datos e da Información			Código	614111102
Titulación	Enxeñeiro en Informática				
Descritores					
Ciclo	Período	Curso	Tipo	Créditos	
1º e 2º Ciclo	2º cuatrimestre	Primeiro	Troncal	4	
Idioma	Castelán				
Prerrequisitos					
Departamento	Computación				
Coordinación	Alonso Pardo, Miguel angel	Correo electrónico	miguel.alonso@udc.es		
Profesorado	Alonso Pardo, Miguel angel	Correo electrónico	miguel.alonso@udc.es		
Web	campusvirtual.udc.es				
Descrición xeral	<p>La asignatura se podría englobar en el bloque temático de asignaturas relacionadas con la Ingeniería del Software, dentro del cual esta asignatura constituye una disciplina totalmente básica. En este grupo, las relaciones más estrechas se establecen con Bases de Datos I, Algoritmos, Metodología de la Programación y Programación Orientada a Objetos.</p> <p>Un segundo bloque temático de asignaturas relacionadas es el que podríamos denominar Fundamentos Matemáticos, y dentro de este grupo, especialmente con la asignatura Matemática Discreta I.</p> <p>Respecto al perfil profesional, muchas áreas de la computación requieren la habilidad de trabajar con las estructuras de datos que se estudian en esta asignatura.</p>				

## Competencias da titulación

Código	Competencias da titulación
A1	Aprender de maneira autónoma novos coñecementos e técnicas avanzadas axeitadas para a investigación, o deseño e o desenvolvemento de sistemas e servizos informáticos.
A2	Concibir e desenvolver novas arquitecturas de computación, en especial para sistemas multiprocesadores, analizando e adaptando diversas alternativas tecnolóxicas a cada problema concreto.
A3	Concibir e planificar o desenvolvemento de aplicacións informáticas complexas ou con requisitos especiais.
A4	Coñecer e aplicar diferentes protocolos de comunicación e sistemas de xestión de rede.
A5	Saber especificar, deseñar e implementar sistemas intelixentes cando as solucións convencionais non resultaren satisfactorias.
A6	Avaliar, definir, seleccionar e auditar plataformas hardware e software para a execución e desenvolvemento de aplicacións e servizos informáticos.
A7	Saber especificar, deseñar e implementar un sistema de información, empregando bases de datos.
A8	Concibir, despregar, organizar e xestionar un servizo informático complexo.
A9	Dirixir equipos de traballo ligados ao deseño de produtos, procesos, servizos informáticos e outras actividades profesionais.
A10	Saber especificar, deseñar e implementar unha política de seguridade no sistema.
B1	Aprender a aprender.
B2	Resolver problemas de forma efectiva.
B3	Aplicar un pensamento crítico, lóxico e creativo.
B4	Aprendizaxe autónoma.
B5	Traballar de forma colaborativa.
B6	Comportarse con ética e responsabilidade social como cidadán e como profesional.
B7	Comunicarse de maneira efectiva en calquera contorno de traballo.
B8	Traballar en equipos de carácter interdisciplinar.
B9	Capacidade para tomar decisións.
B10	Capacidade de xestión da informática (captación e análises da información).



B11	Razoamento crítico.
B12	Capacidade para a análise e a síntese.
B15	Motivación pola calidade.
C1	Expresarse correctamente, tanto de forma oral coma escrita, nas linguas oficiais da comunidade autónoma.
C2	Dominar a expresión e a comprensión de forma oral e escrita dun idioma estranxeiro.
C3	Utilizar as ferramentas básicas das tecnoloxías da información e as comunicacións (TIC) necesarias para o exercicio da súa profesión e para a aprendizaxe ao longo da súa vida.
C4	Desenvolverse para o exercicio dunha cidadanía aberta, culta, crítica, comprometida, democrática e solidaria, capaz de analizar a realidade, diagnosticar problemas, formular e implantar solucións baseadas no coñecemento e orientadas ao ben común.
C6	Valorar criticamente o coñecemento, a tecnoloxía e a información dispoñible para resolver os problemas cos que deben afrontarse.
C7	Asumir como profesional e cidadán a importancia da aprendizaxe ao longo da vida.
C8	Valorar a importancia que ten a investigación, a innovación e o desenvolvemento tecnolóxico no avance socioeconómico e cultural da sociedade.

Resultados da aprendizaxe			
Competencias de materia (Resultados de aprendizaxe)	Competencias da titulación		
Conocer los mecanismos de abstracción en el diseño de estructuras de datos	A1 A6 A9	B1 B2 B3 B4	C3 C6
Comprender el concepto de Tipo Abstracto de Datos en programación operacional	A6 A8	B2 B3 B4 B5 B9 B11 B12 B15	C3 C6
Distinguir los niveles de especificación, implementación y uso para el diseño de tipos abstractos de datos.	A6 A8 A9	B2 B3 B4 B5 B9 B11 B12 B15	C3 C6
Reconocer las estructuras de datos elementales que se necesitan para resolver un problema.	A5 A6 A8 A9 A10	B2 B3 B4 B5 B9 B10 B11 B12 B15	C3 C6



Construir una especificación formal de un tipo abstracto de datos para resolver un problema.	A3 A4 A5 A8 A9	B2 B3 B4 B5 B9 B10 B11 B15	C3 C6
Diseñar tipos abstractos de datos a partir de una especificación formal.	A3 A4 A5 A8	B2 B3 B4 B5 B9 B11 B12	C3 C6
Programar los algoritmos de manipulación de las estructuras de datos diseñadas.	A2 A4 A6 A7 A10	B2 B3 B5 B6 B7 B8 B11 B15	C3 C6 C8
Usar las estructuras de datos para solucionar problemas reales.	A1 A4 A6 A7 A8 A9	B2 B3 B5 B6 B7 B8 B9 B15	C3 C6
Asumir la necesidad de una buena especificación y un buen diseño como pasos previos a la codificación.	A1	B6 B11 B15	C1 C2 C4 C6
Interiorizar las buenas prácticas de programación.	A1 A2 A4 A6 A7	B6 B11 B12 B15	C7 C8

Contidos	
Temas	Subtemas
Gestión dinámica de memoria	Organización de la memoria de un programa. Definición de variables de tipo puntero. Reserva y destrucción dinámica de memoria. Asignación y comparación.



Recursividad	<p>El concepto de recursión.</p> <p>Principios de diseño de subprogramas recursivos.</p> <p>Ejemplos de subprogramas recursivos.</p> <p>La recursión con respecto a la iteración.</p> <p>Algoritmos recursivos de ordenación.</p>
Introducción a los Tipos Abstractos de Datos	<p>La abstracción en programación: Concepto, Evolución a través de la historia de la programación, TAD y Programación orientada a objetos.</p> <p>La modularidad en programación.</p> <p>Tipos Abstractos de Datos: Definición y concepto, Diferencias entre tipo de dato, estructuras de datos y TAD, Construcción de un TAD, Ventajas de la abstracción de datos.</p>
Listas	<p>Especificación informal del TAD Lista.</p> <p>Implementación del TAD Lista.</p> <p>El TAD Lista ordenada: especificación e implementación.</p> <p>Multilistas y listas multiordenadas: concepto, representaciones y usos.</p>
Colas	<p>Especificación informal del TAD Cola.</p> <p>Implementación del TAD Cola.</p> <p>Variantes de las colas. Colas de prioridad.</p> <p>Aplicaciones en computación.</p>
Pilas	<p>Especificación informal del TAD Pila.</p> <p>Implementación del TAD Pila.</p> <p>Aplicaciones en computación.</p>
Árboles	<p>Definición de árbol y terminología</p> <p>Árboles binarios: Especificación informal, Implementación.</p> <p>Variantes de árboles binarios: Especificación e implementación de operaciones.</p>
Árboles de búsqueda	<p>Árboles binarios de búsqueda.</p> <p>Árboles AVL.</p>
Grafos	<p>Introducción</p> <p>Recorridos sobre grafos</p> <p>Operaciones comunes sobre grafos</p>

## Planificación

Metodoloxías / probas	Horas presenciais	Horas non presenciais / traballo autónomo	Horas totais
Sesión maxistral	15	15	30
Aprendizaxe colaborativa	0	5.4	5.4
Solución de problemas	19	19	38
Prácticas de laboratorio	11	11	22
Proba obxectiva	3.6	0	3.6
Atención personalizada	1	0	1

\*Os datos que aparecen na táboa de planificación son de carácter orientativo, considerando a heteroxeneidade do alumnado

## Metodoloxías

Metodoloxías	Descrición
Sesión maxistral	<p>En las clases presenciais de teoría, el profesor realizará una breve descripción de los contenidos temáticos y de los objetivos básicos perseguidos, con el fin de dotar al alumno de una visión global de la materia. Además tratará de establecer interrelaciones con otros conceptos previamente adquiridos, de forma que se pueda establecer una línea temporal, y expondrá la bibliografía recomendada. Seguidamente pasará a desarrollar los contenidos teóricos, utilizando como método la clase magistral.</p>



Aprendizaxe colaborativa	Lectura de fuentes bibliográficas propuestas y elaboración de un documento-resumen de extensión limitada, que sirva como punto de partida y referencia de otras actividades.
Solución de problemas	En las clases presenciales de problemas, con el fin de afianzar los conceptos teóricos se presentarán supuestos prácticos, que en un principio serán resueltos por el profesor para orientar a los alumnos. Tanto en las clases de problemas como en los ejemplos mostrados durante las exposiciones teóricas, cuando éstos impliquen el desarrollo de código o pseudocódigo éste se realizará mostrando los sucesivos pasos del diseño ascendente. Con esto pretendemos: a) que el alumno se acostumbre al uso de este método, y b) evitar que se pierda en los detalles de sintaxis y las características particulares del lenguaje, en lugar de fijar su atención en la comprensión y diseño de la solución. Como actividades no presenciales, se plantearán ejercicios adicionales que el alumno deberá resolver y comentar/corregir con el profesor durante las horas de tutorías, colectivas y/o individuales. Se trata de fomentar la participación de los alumnos y promover, en la medida de lo posible, el diálogo abierto y la valoración de soluciones. Al finalizar cada tema, se proporcionará, utilizando los recursos virtuales de docencia, un test de autoevaluación para que al alumno pueda comprobar el progreso de su aprendizaje.
Prácticas de laboratorio	Las clases de prácticas obligan a la realización de prácticas de programación sobre estructuras de datos en un lenguaje de alto nivel. Se impondrá una periodicidad en su entrega para fomentar el estudio continuo. El enunciado de las prácticas, que se proporcionará con la suficiente antelación, detallará el problema y las especificaciones, que el alumno deberá respetar estrictamente. Este enunciado se analizará con los alumnos durante la primera hora de prácticas del bloque correspondiente. Posteriormente, la labor del profesor será la de supervisar las sesiones de prácticas, solucionando dudas y corrigiendo errores de interpretación, malos hábitos de programación y errores de sintaxis, etc.
Proba obxectiva	Evaluación sumativa del alumno mediante un examen final al término del cuatrimestre. Éste será eminentemente práctico para que el alumno pueda demostrar que ha adquirido los conocimientos necesarios de abstracción y diseño de TADs y se ha entrenado lo suficiente como para poseer las habilidades precisas para resolver supuestos prácticos que impliquen la aplicación de dichas estructuras. La puntuación asignada a cada una de las preguntas del examen irá consignada en la prueba.

### Atención personalizada

Metodoloxías	Descrición
Aprendizaxe colaborativa Solución de problemas Prácticas de laboratorio Proba obxectiva	El desarrollo, tanto de las clases magistrales como de las de resolución de problemas y los laboratorios de prácticas, se realizará atendiendo al progreso de los alumnos en las capacidades de comprensión y asimilación de los contenidos impartidos. El avance general de la clase se compaginará con una atención específica a aquellos alumnos que presenten mayores dificultades en la tarea del aprendizaje y con un apoyo adicional a aquellos otros que presenten mayor desenvoltura y deseen ampliar conocimientos.  En lo que respecta a las tutorías individuales, dado su carácter personalizado, no deben dedicarse a extender los contenidos con nuevos conceptos, sino a aclarar los conceptos ya expuestos. El profesor las utilizará como una interacción que le permita extraer conclusiones respecto al grado de asimilación de la materia por parte de los alumnos.

### Avaliación

Metodoloxías	Descrición	Cualificación
Proba obxectiva	Realización obligatoria. Necesario aprobar la prueba para superar la asignatura.	100
Outros		

### Observacións avaliación

--

### Fontes de información

--



<b>Bibliografía básica</b>	<ul style="list-style-type: none"><li>- Liskov, B. y Guttag, J. (1989). Abstraction and specification in program development. The MIT Press</li><li>- Standish, T.A. (1994). Data structures, algorithms, and software principles. Addison-Wesley</li><li>- Cairó O. y Guardati S. (1993). Estructuras de datos. McGraw-Hill Interamericana de México, S.A. de C.V.</li><li>- Weiss, M.A. (1995). Estructuras de datos y algoritmos. Wilmington, Delaware. Addison-Wesley Iberoamericana</li><li>- Joyanes Aguilar, L. y Zahonero Martínez, I. (1998). Estructuras de Datos: algoritmos, abstracción y objetos. Madrid. McGraw-Hill/Interamericana de España</li><li>- Collado Machuca, M., Morales Fernández, R. y Moreno Navarro, J.J. (1987). Estructuras de Datos: Realización en Pascal. Madrid. Diaz de Santos</li><li>- Carmona Poyato, A., Merina Carnicer, R., Madrid Cuevas, F.J., Romero del Castillo J.A., Fernández G. (1999). Pascal y Estructuras de datos. Servicio Publicaciones, Universidad de Córdoba</li><li>- Dale, N. y Lilly, S.C. (1989). Pascal y Estructuras de datos (segunda edición). Madrid. McGraw-Hill/Interamericana de España</li></ul>
<b>Bibliografía complementaria</b>	<ul style="list-style-type: none"><li>- Hernández, R., Lázaro, J.C., Dormido, R. y Ros, S. (2001). Estructuras de datos y algoritmos. Madrid. Prentice Hall</li><li>- Aho, A.V., Hopcroft, J.E. y Ullman, J.D. (1988). Estructuras de datos y algoritmos. Wilmington, Delaware. Addison-Wesley Iberoamericana</li><li>- Horowitz, E. y Sahni, S. (1990). Fundamentals of Data Structures in Pascal. Rockville, Maryland. Computer Science Press</li></ul>

## Recomendacións

### Materias que se recomenda ter cursado previamente

Programación/614111109

### Materias que se recomenda cursar simultaneamente

### Materias que continúan o temario

### Observacións

(\*A Guía docente é o documento onde se visualiza a proposta académica da UDC. Este documento é público e non se pode modificar, salvo casos excepcionais baixo a revisión do órgano competente dacordo coa normativa vixente que establece o proceso de elaboración de guías