



Guía docente				
Datos Identificativos				2019/20
Asignatura (*)	Programación II	Código	614G01006	
Titulación	Grao en Enxeñaría Informática			
Descriptorios				
Ciclo	Periodo	Curso	Tipo	Créditos
Grado	2º cuatrimestre	Primero	Formación básica	6
Idioma	CastellanoInglés			
Modalidad docente	Presencial			
Prerrequisitos				
Departamento	Ciencias da Computación e Tecnoloxías da InformaciónComputación			
Coordinador/a	Alonso Pardo, Miguel angel	Correo electrónico	miguel.alonso@udc.es	
Profesorado	Alonso Pardo, Miguel angel Barreira Rodriguez, Noelia Bolón Canedo, Verónica Cabreiro Canosa, Mariano Javier De Moura Ramos, Jose Joaquim Gómez Rodríguez, Carlos Guijarro Berdiñas, Berta M. Hernandez Pereira, Elena Maria Monroy Camafreita, Juan Paz López, Alejandro Pérez Sánchez, Beatriz Sanchez Maroño, Noelia Vilares Ferro, Jesus	Correo electrónico	miguel.alonso@udc.es noelia.barreira@udc.es veronica.bolon@udc.es mariano.cabreiro@udc.es joaquim.demoura@udc.es carlos.gomez@udc.es berta.guijarro@udc.es elena.hernandez@udc.es juan.monroy@udc.es alejandro.paz.lopez@udc.es beatriz.perezs@udc.es noelia.sanchez@udc.es jesus.vilares@udc.es	
Web	moodle.udc.es			
Descripción general	<p>La asignatura pertenece al bloque de materias de Lenguajes y Programación del Módulo de Formación Básica de la titulación, con una fuerte interrelación con las materias del Módulo Común a la Rama de Informática. Las relaciones más estrechas se establecen con Bases de Datos, Algoritmos y Diseño Software.</p> <p>Un segundo bloque temático de asignaturas relacionadas es el que forman aquellas de la Materia Matemáticas, y dentro de este grupo, especialmente la asignatura Matemática Discreta.</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>			
Plan de contingencia				

Competencias del título	
Código	Competencias del título
A3	Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la ingeniería.
A4	Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
B1	Capacidad de resolución de problemas
B3	Capacidad de análisis y síntesis
C3	Utilizar las herramientas básicas de las tecnologías de la información y las comunicaciones (TIC) necesarias para el ejercicio de su profesión y para el aprendizaje a lo largo de su vida.
C6	Valorar críticamente el conocimiento, la tecnología y la información disponible para resolver los problemas con los que deben enfrentarse.

Resultados de aprendizaje	
Resultados de aprendizaje	Competencias del título



Dominar los mecanismos de gestión dinámica de la memoria.	A4	B1	C6
Conocer los mecanismos de abstracción en el diseño de estructuras de datos.	A4	B1 B3	C3 C6
Construir especificaciones, diseñar el tipo abstracto a partir de ella, usar las estructuras de datos adecuadas.	A3 A4	B1 B3	C3 C6
Usar las estructuras de datos adecuadas y programar los algoritmos de manipulación para solucionar problemas reales.	A3 A4	B1 B3	C3 C6
Asumir la necesidad de una buena especificación y un buen diseño como pasos previos a la codificación.	A4	B3	C6
Interiorizar las buenas prácticas de programación.	A4	B3	

Contenidos	
Tema	Subtema
Gestión dinámica da memoria	Organización de la memoria de un programa. Definición de variables de tipo puntero. Reserva y destrucción dinámica de memoria. Asignación y comparación.
Introducción a los Tipos Abstractos de Datos	La abstracción en programación: Concepto, Evolución a través de la historia de la programación, TAD y Programación orientada a objetos. La modularidad en programación. 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.
Listas	Especificación informal del TAD Lista. Implementación del TAD Lista. El TAD Lista ordenada: especificación e implementación. Multilistas y listas multiordenadas: concepto, representaciones y usos.
Pilas	Especificación informal del TAD Pila. Implementación del TAD Pila. Aplicaciones en computación.
Colas	Especificación informal del TAD Cola. Implementación del TAD Cola. Variantes de las colas. Colas de prioridad. Aplicaciones en computación.
Árbores	Definición de árbol y terminología TAD Árbol binario: Especificación informal, Implementación. Recorridos de árboles.
Árboles de búsqueda	Árboles binarios de búsqueda. Árboles AVL.

Planificación				
Metodologías / pruebas	Competencias	Horas presenciales	Horas no presenciales / trabajo autónomo	Horas totales
Sesión magistral	A3 A4 B1 B3	30	30	60
Solución de problemas	A3 B1 B3 C6	10	14	24
Prácticas de laboratorio	A4 B1 B3 C3 C6	20	26	46
Prueba objetiva	A3 A4 B1 B3	3	15	18
Atención personalizada		2	0	2

(*Los datos que aparecen en la tabla de planificación són de carácter orientativo, considerando la heterogeneidad de los alumnos



Metodoloxías	
Metodoloxías	Descrición
Sesión magistral	En las clases presenciales 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.
Solución de problemas	<p>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 que orienten los alumnos. A medida que se avance en el desarrollo teórico se formulará la resolución de problemas por parte de los alumnos, constituidos en grupos de trabajo. Dicha actividad, así como la discusión y participación activa en clase, se valorarán en la nota final.</p> <p>Tanto en las clases de problemas como en los ejemplos mostrados durante las exposiciones teóricas, cuando estos impliquen el desarrollo de código o pseudocódigo este se realizará mostrando los sucesivos pasos del diseño descendente. 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.</p> <p>Como actividades no presenciales, se formularán 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.</p>
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 para que el alumno lo lea con detenimiento y analice en profundidad, detallará el problema y las especificaciones, que deberán respetarse estrictamente. Posteriormente, el cometido 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.
Prueba objetiva	Evaluación sumativa del alumno mediante un examen final al término del cuatrimestre. Este será eminentemente práctico para que el alumno pueda demostrar que hay adquirido los conocimientos necesarios de abstracción y diseño de TADs y se adiestró lo suficiente como para poseer las habilidades precisas para resolver supuestos prácticos que impliquen la aplicación de dichas estructuras.

Atención personalizada	
Metodoloxías	Descrición
Solución de problemas Prácticas de laboratorio Prueba objetiva	<p>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.</p> <p>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.</p>

Evaluación			
Metodoloxías	Competencias	Descrición	Calificación



Solución de problemas	A3 B1 B3 C6	Se valorarán los resultados, forma y condiciones de realización de diversos trabajos puntuables que se detallarán durante el curso y que se resolverán en las tutorías de grupos reducidos. La calificación solamente se sumará a la obtenida en las pruebas restantes cuando la materia esté superada.	10
Prácticas de laboratorio	A4 B1 B3 C3 C6	Realización obligatoria segun las condiciones establecidas en el enunciado de cada práctica. Necesario aprobar todas las prácticas para superar la materia.	20
Prueba objetiva	A3 A4 B1 B3	Realización obligatoria. Necesario aprobar el examen para superar a materia.	80
Otros			

Observaciones evaluación

Trabajos prácticos

- Sólomente los alumnos con calificación de NO APTO o NO PRESENTADO en prácticas en la primera oportunidad podrán entregar los trabajos de acuerdo al enunciado práctico que se proponga para la segunda oportunidad.
- De acuerdo al artículo 14, apartado 4, de la normativa*, el plagio de los trabajos prácticos llevará una nota global de NO APTO, tanto al estudiante que presente material copiado como a quien lo facilite, y por lo tanto la calificación de SUSPENSO en la convocatoria anual.

Matrícula a tiempo parcial

- Los alumnos matriculados a tiempo parcial tendrán que entregar las actividades evaluables en las condiciones y plazos específicos que se establecerán. Será deber del estudiante comunicar su situación al profesorado.

No presentado

- Tendrá la condición de "No presentado" (NP) quien no concurra a la prueba objetiva en el periodo oficial de evaluación.

Oportunidad adelantada de diciembre

- La evaluación de la oportunidad adelantada se basará exclusivamente en una prueba escrita.

* Normativa de evaluación, revisión y reclamación de las calificaciones de los estudios de grado y máster universitario, aprobada por el Consejo de Gobierno de la Universidad de A Coruña el 19 de diciembre de 2013.

Fuentes de información

Básica	<ul style="list-style-type: none"> - Narasimha Karumanchi (2017). Data Structures and Algorithms Made Easy, 5th Edition. CareerMonk Publications - Ignacio Zahonero y Luis Joyanes Aguilar (2004). Algoritmos y estructuras de datos: Una perspectiva en C. McGraw-Hill - Kyle Loudon (1999). Mastering Algorithms with C. O'Reilly Media
Complementaria	<ul style="list-style-type: none"> - Reema Thareja (2014). Data Structures Using C - Second Edition. Oxford University Press - Aaron M. Tenenbaum, Yedidyah Langsam & Moshe J. Augenstein (1989). Data Structures Using C. Prentice Hall - Richard F. Gilberg & Behrouz A. Forouzan (2005). Data Structures: A Pseudocode Approach with C (2nd Ed). Cengage Learning - Ignacio Zahonero y Luis Joyanes Aguilar (2005). Programación en C. Metodología, Algoritmos y Estructura de Datos, 2º Edición. McGraw-Hill - Luis Joyanes Aguilar, Andrés Castillo Sanz, Lucas Sánchez García e Ignacio Zahonero Martínez (2002). Programación en C. Libro de problemas. McGraw-Hill - Ignacio Zahonero, Lucas García Sánchez, Luis Joyanes Aguilar y Matilde Fernández Azuela (2005). Estructuras de datos en C (Serie Schaum). McGraw-Hill <p>Páxina oficial da contorna de desenvolvemento CLion: https://www.jetbrains.com/clion/</p>

Recomendaciones

Asignaturas que se recomienda haber cursado previamente

Programación I/614G01001
 Informática Básica/614G01002
 Matemática Discreta/614G01004



Asignaturas que se recomienda cursar simultáneamente
Asignaturas que continúan el temario
Algoritmos/614G01011 Bases de Datos/614G01013 Paradigmas de Programación/614G01014 Diseño Software/614G01015
Otros comentarios

(*) La Guía Docente es el documento donde se visualiza la propuesta académica de la UDC. Este documento es público y no se puede modificar, salvo cosas excepcionales bajo la revisión del órgano competente de acuerdo a la normativa vigente que establece el proceso de elaboración de guías