Guia docente				
Datos Identificativos				2022/23
Asignatura (*)	Fundamentos de Programación II		Código	614G02009
Titulación	Grao en Ciencia e Enxeñaría de Datos			
Descriptores				
Ciclo	Periodo	Curso	Tipo	Créditos
Grado	2º cuatrimestre	Primero	Formación básica	6
Idioma	Castellano			
Modalidad docente	Presencial			
Prerrequisitos				
Departamento	Ciencias da Computación e Tecnoloxías da Información			
Coordinador/a	Hernandez Pereira, Elena Maria	Correo electrónico	elena.hernandez@udc.es	
Profesorado	Alonso Pardo, Miguel angel Cabrero Canosa, Mariano Javier Hernandez Pereira, Elena Maria Vilares Calvo, David	Correo electrónico	miguel.alonso@udc.es mariano.cabrero@udc.es elena.hernandez@udc.es david.vilares@udc.es	
Web	campusvirtual.udc.gal			
Descripción general	En esta asignatura se presentan las técnicas de diseño de programas, incluyendo los fundamentos de la orientación a objetos, asi como las estructuras de datos básicas en computación y sus principios de uso. La asignatura pertenece al bloque Programación y Algoritmos, por lo que la relación más estrecha se da con las otras asignaturas de este mismo bloque: Fundamentos de Programación I (que se puede considerar predecesora directa) y Diseño y Análisis de Algoritmos (que se puede considerar sucesora directa). Con respeto a los otros bloques, las relaciones más directas se dan con Bases de Datos y Sistemas para Procesamiento de Datos. Otro bloque temático de asignaturas relacionadas es el que forman aquellas de Fundamentos Matemáticos, y dentro de este grupo, especialmente la materia Matemática Discreta.			

Competencias del título	
Código	Competencias del título
A5	CE5 - Conocimiento de estructuras de datos y algoritmos básicos y capacidad para utilizarlos eficientemente en la resolución de un problema.
B1	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
B5	CB5 - Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía
B6	CG1 - Ser capaz de buscar y seleccionar la información útil necesaria para resolver problemas complejos, manejando con soltura las fuentes bibliográficas del campo.
C1	CT1 - 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.

Resultados de aprendizaje			
Resultados de aprendizaje	Competencias del título		
Comprender y saber programar utilizando orientación a objetos		B1 B5 B6	C1
Conocer y saber usar lenguajes de programación de relevancia actual		B1 B5 B6	C1

Comprender los principios básicos de almacenamento de datos y su manipulación	A5	B1 B6	C1
Conocer y saber utilizar las estructuras de datos estándar en computación y los algoritmos más relevantes para manipularlas	A5	B5 B6	C1
Identificar la estructura de datos más adecuada para un problema determinado	A5	B5 B6	C1

Contenidos	
Tema	**Subtema**
Técnicas de diseño de programas	Abstracción y especificacación Módulos funcionales y de datos Manejo de excepciones y de eventos
Orientación a objetos	Clases y objetos. Métodos Clases y funciones Herencia Interfaces y Polimorfismo
Utilización de las estructuras de datos básicas en computación	Listas Pilas Colas Colas de Prioridad Diccionarios Árboles Árboles Binarios de Búsqueda Tablas Hash Grafos

Planificación				
Metodologías / pruebas	**Competéncias**	**Horas presenciales**	**Horas no presenciales / trabajo autónomo**	**Horas totales**
Sesión magistral	A5 B1 B5 B6	30	24	54
Prácticas de laboratorio	A5 B1 B5 B6 C1	20	36	56
Solución de problemas	A5 B1 B5 B6 C1	10	17.5	27.5
Prueba objetiva	A5 B1 B5 B6	3	7.5	10.5
Atención personalizada		2	0	2
(*)Los datos que aparecen en la tabla de planificación són de carácter orientativo, considerando la heterogeneidad de los alumnos				

Metodologías	
Metodologías	**Descripción**
Sesión magistral	En las clases 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.
Prácticas de laboratorio	Las clases de prácticas obligan a la realización de prácticas de programación 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, 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.

Solución de problemas	En las clases 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. 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. 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 descendiente. 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. 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.
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 adquirió los conocimientos necesarios de diseño de programas, orientación a objetos y utilización de estructuras de datos, y que entrenó lo suficiente como para poseer las habilidades precisas para resolver supuestos prácticos que impliquen la aplicación de dichos conocimientos.

Atención personalizada	
Metodologías	**Descripción**
Prácticas de laboratorio Solución de problemas	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 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. La atención personalizada se realizará preferentemente a través de las herramientas corporativas de la UDC (email, Teams, etc.) respetando los horarios de tutorías.

Evaluación			
Metodologías	**Competéncias**	**Descripción**	**Calificación**
Prácticas de laboratorio	A5 B1 B5 B6 C1	Realización según las condiciones establecidas en el enunciado de cada práctica. Necesario aprobar las practicas para superar la asignatura.	40
Prueba objetiva	A5 B1 B5 B6	Realización obligatoria. Necesario aprobar la prueba para superar la asignatura.	60

Observaciones evaluación

Trabajos prácticos y solución de problemas - De acuerdo con el artículo 14, apartado 4, de la normativa*, el plagio de los trabajos conllevará una nota global de NO APTO (0), tanto al estudiante que presente material copiado como a quien lo facilitase, y la calificación de SUSPENSO en la convocatoria anual. - Si las prácticas u otras actividades se llevan a cabo en grupos, todos los miembros del grupo responderán de forma solidaria del trabajo realizado y entregado y sus posibles consecuencias. 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 obligación del estudiante comunicar su situación al profesorado. No presentado - Quien no realice la prueba objetiva en el período oficial de evaluación tendrá la condición de ?No presentado? (NP). * Normativa de avaliación, revisión e reclamación das cualificacións dos estudos de grao e máster universitario, aprobada polo Consello de Goberno da Universidade da Coruña o 19 de decembro de 2013.

Fuentes de información	
Básica	- Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser (2013). Data Structures and Algorithms in Python. John Wiley & Sons - Kenneth A. Lambert (2013). Fundamentals of Python: Data Structures. Course Technology
Complementária	- Bradley N. Miller, David L. Ranum (2013). Problem Solving with Algorithms and Data Structures using Python. Franklin, Beedle & Associates - Benjamin Baka (2017). Python Data Structures and Algorithms: Improve application performance with graphs, stacks, and queues. Packt Publishing Sitio web da contorna de desenvolvemento Spyder: https://www.spyder-ide.org/Sitio web da plataforma para ciencia de datos Anaconda: https://www.anaconda.com/ Libro de Miller & Ranum: https://runestone.academy/runestone/static/pythonds/index.html Sitio web da contorna de desenvolvemento Spyder: https://www.spyder-ide.org/Sitio web da plataforma para ciencia de datos Anaconda: https://www.anaconda.com/Libro de Miller & Ranum: https://runestone.academy/runestone/static/pythonds/index.html

Recomendaciones
Asignaturas que se recomienda haber cursado previamente
Matemática Discreta/614G02002 Fundamentos de Programación I/614G02004
Asignaturas que se recomienda cursar simultáneamente
Asignaturas que continúan el temario
Diseño y Análisis de Algoritmos/614G02011
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