



Guía Docente				
Datos Identificativos				2019/20
Asignatura (*)	Fundamentos da Programación II	Código	614G02009	
Titulación				
Descritores				
Ciclo	Período	Curso	Tipo	Créditos
Grao	2º cuatrimestre	Primeiro	Formación básica	6
Idioma	Castelán			
Modalidade docente	Presencial			
Prerrequisitos				
Departamento	Ciencias da Computación e Tecnoloxías da Información			
Coordinación	Alonso Pardo, Miguel angel	Correo electrónico	miguel.alonso@udc.es	
Profesorado	Alonso Pardo, Miguel angel	Correo electrónico	miguel.alonso@udc.es	
	Cabrero Canosa, Mariano Javier		mariano.cabrero@udc.es	
	Hernandez Pereira, Elena Maria		elena.hernandez@udc.es	
Web	moodle.udc.es			
Descrición xeral	<p>Nesta materia preséntanse as técnicas de deseño de programas, incluíndo os fundamentos da orientación a obxectos, así como as estruturas de datos básicas en computación e os seus principios de uso.</p> <p>A materia pertence ao bloque Programación e Algoritmos, polo que a relación máis estreita dáse coas outras dos materias deste mesmo bloque: Fundamentos de Programación I (que se pode considerar predecesora directa) e Deseño e Análise de Algoritmos (que se pode considerar sucesora directa). Con respecto aos outros bloques, as relacións máis directas danse con Bases de Datos e Sistemas para Procesamento de Datos. Outro bloque temático de materias relacionadas é o que forman aquelas de Fundamentos Matemáticos, e dentro deste grupo, especialmente a materia Matemática Discreta.</p>			

Competencias / Resultados do título	
Código	Competencias / Resultados do título

Resultados da aprendizaxe			
Resultados de aprendizaxe	Competencias / Resultados do título		
Comprender e saber programar utilizando orientación a obxectos		B1 B5 B6	C1
Coñecer e saber usar linguaxes de programación de relevancia actual		B1 B5 B6	C1
Comprender os principios básicos de almacenamento de datos e a súa manipulación	A5	B1 B6	C1
Coñecer e saber utilizar as estruturas de datos estándar en computación e os algoritmos máis relevantes para manipularlas	A5	B5 B6	C1
Identificar a estrutura de datos máis adecuada para un problema determinado	A5	B5 B6	C1

Contidos	
Temas	Subtemas
Técnicas de deseño de programas	Abstracción e especificación Módulos funcionais e de datos Manexo de excepcións y eventos



Orientación a obxectos	Clases e obxectos. Métodos. Clases e funcións Herencia Interfaces e Polimorfismo
Utilización das estruturas de datos básicas en computación	Listas Pilas Colas Colas de Prioridade Dicionarios Árbores Árbores Binarias de Busca Tablas Hash Grafos

Planificación				
Metodoloxías / probas	Competencias / Resultados	Horas lectivas (presenciais e virtuais)	Horas traballo autónomo	Horas totais
Sesión maxistral	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
Proba obxectiva	A5 B1 B5 B6	3	7.5	10.5
Atención personalizada		2	0	2

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

Metodoloxías	
Metodoloxías	Descrición
Sesión maxistral	Nas clases presenciais de teoría, o profesor realizará unha breve descrición dos contidos temáticos e dos obxectivos básicos perseguidos, co fin de dotar o alumno dunha visión global da materia. Ademais tratará de establecer interrelacións con outros conceptos previamente adquiridos, de forma que se poida establecer unha liña temporal, e expoñerá a bibliografía recomendada. Seguidamente pasará a desenvolver os contidos teóricos, utilizando como método a clase maxistral.
Prácticas de laboratorio	As clases de prácticas obrigan á realización de prácticas de programación nunha linguaxe de alto nivel. Impoñeráse unha periodicidade na súa entrega para fomentar o estudo continuo. O enunciado das prácticas, que se proporcionará coa suficiente antelación para que o alumno o lea con detemento e analice en profundidade, detallará o problema e as especificacións, que deberán respectarse estritamente. Posteriormente, o labor do profesor será o de supervisar as sesións de prácticas, solucionando dúbidas e corrixindo erros de interpretación, malos hábitos de programación e erros de sintaxe, etc.



Solución de problemas	<p>Nas clases presenciais de problemas, co fin de afianzar os conceptos teóricos presentaranse supostos prácticos, que nun principio serán resoltos polo profesor para orientar os alumnos. A medida que se avance no desenvolvemento teórico formularase a resolución de problemas por parte dos alumnos, constituídos en grupos de traballo. A devandita actividade, así como a discusión e participación activa en clase, valoraranse na nota final.</p> <p>Tanto nas clases de problemas coma nos exemplos mostrados durante as exposicións teóricas, cando estes impliquen o desenvolvemento de código ou pseudocódigo este realizarase mostrando os sucesivos pasos do deseño descendente. Con isto pretendemos: a) que o alumno se acostume ao uso deste método, e b) evitar que se perda nos detalles de sintaxe e as características particulares da linguaxe, en lugar de fixar a súa atención na comprensión e deseño da solución.</p> <p>Como actividades non presenciais, formularanse exercicios adicionais que o alumno deberá resolver e comentar/corrixir co profesor durante as horas de titorías, colectivas e/ou individuais. Trátase de fomentar a participación dos alumnos e promover, na medida do posible, o diálogo aberto e a valoración de solucións.</p>
Proba obxectiva	<p>Avaliación sumativa do alumno mediante un exame final ao termo do cuadrimestre. Este será eminentemente práctico para que o alumno poida demostrar que adquiriu os coñecementos necesarios de deseño de programas, orientación a obxectos e utilización de estruturas de datos, e que adestrou o suficiente como para posuír as habilidades precisas para resolver supostos prácticos que implique a aplicación dos devanditos coñecementos.</p>

### Atención personalizada

Metodoloxías	Descrición
Prácticas de laboratorio Solución de problemas	<p>O desenvolvemento, tanto das clases maxistras coma das de resolución de problemas e os laboratorios de prácticas, realizarase atendendo ao progreso dos alumnos nas capacidades de comprensión e asimilación dos contidos impartidos. O avance xeral da clase compaxinarase cunha atención específica a aqueles alumnos que presenten maiores dificultades na tarefa da aprendizaxe e cun apoio adicional a aqueles outros que presenten maior desenvolvemento e desexen ampliar coñecementos.</p> <p>No que respecta ás titorías individuais, dado o seu carácter personalizado, non deben dedicarse a estender os contidos con novos conceptos, senón a aclarar os conceptos xa expostos. O profesor utilizaráas como unha interacción que lle permita extraer conclusións respecto ao grao de asimilación da materia por parte dos alumnos.</p>

### Avaliación

Metodoloxías	Competencias / Resultados	Descrición	Cualificación
Prácticas de laboratorio	A5 B1 B5 B6 C1	Realización segundo as condicións establecidas no enunciado de cada práctica.	20
Solución de problemas	A5 B1 B5 B6 C1	Valoraranse os resultados, forma e condicións de realización de diversos traballos puntuables que se detallarán durante o curso.	10
Proba obxectiva	A5 B1 B5 B6	Realización obrigatoria. Necesario aprobar a proba para superar a materia.	70

### Observacións avaliación



## Traballos prácticos e solución de problemas

- De acordo

ao artigo 14, apartado 4, da normativa\*, o plaxio dos traballos levará unha nota global de NON APTO, tanto ao estudante que presente material copiado como a quen o facilite, e a cualificación de SUSPENSO na convocatoria anual.

### Matrícula a tempo parcial

-

Os alumnos matriculados a tempo parcial terán que entregar as actividades avaliadas nas condicións e prazos específicos que se establecerán. Será obriga do estudante comunicar a súa situación ao profesorado.

### Non presentado

- Quen non concurra á proba obxectiva no período oficial de avaliación terá a condición de ?Non 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.

## Fontes de información

<b>Bibliografía básica</b>	- 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
<b>Bibliografía complementaria</b>	- 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: <a href="https://www.spyder-ide.org/">https://www.spyder-ide.org/</a> Sitio web da plataforma para ciencia de datos Anaconda: <a href="https://www.anaconda.com/">https://www.anaconda.com/</a> Libro de Miller & Ranum: <a href="https://runestone.academy/runestone/static/pythonds/index.html">https://runestone.academy/runestone/static/pythonds/index.html</a> Sitio web da contorna de desenvolvemento Spyder: <a href="https://www.spyder-ide.org/">https://www.spyder-ide.org/</a> Sitio web da plataforma para ciencia de datos Anaconda: <a href="https://www.anaconda.com/">https://www.anaconda.com/</a> Libro de Miller & Ranum: <a href="https://runestone.academy/runestone/static/pythonds/index.html">https://runestone.academy/runestone/static/pythonds/index.html</a>

## Recomendacións

### Materias que se recomenda ter cursado previamente

Matemática Discreta/614G02002

Fundamentos da Programación I/614G02004

### Materias que se recomenda cursar simultaneamente

### Materias que continúan o temario

Deseño e Análise de Algoritmos/614G02011

### 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