



Guía Docente				
Datos Identificativos				2024/25
Asignatura (*)	Introdución á programación	Código	614522001	
Titulación	Mestrado Universitario en Bioinformática para Ciencias da Saúde			
Descritores				
Ciclo	Período	Curso	Tipo	Créditos
Mestrado Oficial	1º cuatrimestre	Primeiro	Optativa	6
Idioma	Castelán			
Modalidade docente	Presencial			
Prerrequisitos				
Departamento	Ciencias da Computación e Tecnoloxías da InformaciónComputación			
Coordinación	Cabrero Canosa, Mariano Javier	Correo electrónico	mariano.cabrero@udc.es	
Profesorado	Cabrero Canosa, Mariano Javier	Correo electrónico	mariano.cabrero@udc.es	
Web	udconline.udc.gal			
Descrición xeral	Nesta materia preténdese que os estudantes sen formación en programación adquiren as nocións básicas para a realización de programas. Usarase a linguaxe de programación Python e sobre el estudaranse os diferentes tipos de datos que podemos usar e as estruturas de control básicas que se utilizan para realizar un programa software.			

Competencias / Resultados do título	
Código	Competencias / Resultados do título
A3	CE3 - Analizar , deseñar , desenvolver, implementar , verificar e documentar solucións software eficientes sobre a base dun coñecemento adecuado das teorías, modelos e técnicas actuais no eido da Bioinformática
B1	CB6 ? Posuír e comprender o coñecemento que fornecen unha base ou oportunidade de orixinalidade no desenvolvemento e / ou aplicación de ideas, a miúdo nun contexto de investigación.
B5	CB10 ? Que os estudantes posúan as habilidades de aprendizaxe que lles permitan continuar estudando dun xeito que terá de ser en gran parte auto-orientado ou autónomo.
B8	CG3 - Ser capaz de traballar en equipo, en especial de carácter interdisciplinar
C3	CT3 - 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
C6	CT6 - Valorar criticamente o coñecemento, a tecnoloxía e a información dispoñibles para resolver os problemas cos que deben enfrontarse

Resultados da aprendizaxe			
Resultados de aprendizaxe			Competencias / Resultados do título
Ser capaz de usar linguaxes de programación usuais en Bioinformática.	AP3	BP5 BP8	
Poder deseñar e escribir algoritmos básicos para o manexo das estruturas de datos fundamentais.	AP3	BP1 BP8	
Ser capaz de deseñar, escribir e corrixir programas de ordenador.	AP3	BP1 BP5 BP8	CP3 CP6
Ser capaz de deseñar, avaliar, comparar e analizar algoritmos básicos.	AP3	BP1	CP6

Contidos	
Temas	Subtemas



1. Introducción	<ul style="list-style-type: none"><li>a. Algoritmos. Representación. Accións primitivas/no primitivas</li><li>b. Programas. Proceso de construción</li><li>c. Linguaxes de programación: máquina, baixo nivel, alto nivel</li><li>d. Compiladores. Intérpretes</li><li>e. Entornos de desenvolvemento e ferramentas: Python</li></ul>
2. Conceptos básicos	<ul style="list-style-type: none"><li>a. Estructura de un programa</li><li>b. Constantes, Variables.</li><li>c. Tipos de datos: enteiro, real, lóxico, carácter, ?</li><li>d. Estructuras simples: listas (arrays), cadenas, ?</li><li>e. Operadores y expresiones (aritméticas, lógicas)</li><li>f. Declaración de variables e constantes</li><li>g. Entrada y salida estándar</li></ul>
3. Sentencias de control	<ul style="list-style-type: none"><li>a. Secuencial</li><li>b. Alternativa</li><li>c. Repetitiva: while, for</li></ul>
4. Funcións	<ul style="list-style-type: none"><li>a. Definición, declaración e chamada de función</li><li>b. O ámbito das variables</li><li>c. Paso de argumentos</li><li>d. Recursividad</li><li>e. Módulos</li><li>f. Análise de complexidade</li></ul>
5. Ficheiros	<ul style="list-style-type: none"><li>a. Apertura e peche</li><li>b. Lectura e escritura de datos</li><li>c. Acceso directo aos datos</li></ul>
6. Introducción á orientación a obxectos	<ul style="list-style-type: none"><li>a. Clases</li><li>b. Obxectos</li><li>c. Propiedades</li><li>d. Métodos</li><li>e. Concepto de herencia</li></ul>
7. Excepcións	<ul style="list-style-type: none"><li>a. Tipos</li><li>b. Captura</li><li>c. Lanzamento</li><li>d. Creación</li></ul>
8. Librerías científicas en Python	<ul style="list-style-type: none"><li>a. SciPy</li><li>b. NumPy</li><li>c. Matplotlib</li><li>d. BioPython</li></ul>
9. Introducción a estruturas abstractas	<ul style="list-style-type: none"><li>a. Listas</li><li>b. Pilas</li><li>c. Colas</li><li>d. Árbores</li></ul>

## Planificación

Metodoloxías / probas	Competencias / Resultados	Horas lectivas (presenciais e virtuais)	Horas traballo autónomo	Horas totais
Sesión maxistral	A3 B1 B5	15	30	45
Proba mixta	A3	3	0	3
Foro virtual	B1 B5 C3 C6	1	5	6
Solución de problemas	A3 B1 B5 B8 C3 C6	36	54	90



Atención personalizada		6	0	6
*Os datos que aparecen na táboa de planificación son de carácter orientativo, considerando a heteroxeneidade do alumnado				

Metodoloxías	
Metodoloxías	Descrición
Sesión maxistral	Actividade presencial para expoñer conceptos fundamentais da materia. Consistirá na exposición oral do profesor apoiada con medios multimedia. Durante a presentación tratarase de interactuar co alumno formulando preguntas dirixidas co fin de afianzar conceptos e facilitar a aprendizaxe. A proporción de uso desta metodoloxía será maior fronte a estudo de casos cando o número de estudantes sexa alto e será acordado con estes.
Proba mixta	Avaliación sumativa do alumno mediante un exame escrito cunha parte teórica con preguntas tipo test e unha parte práctica para resolver pequenos problemas de programación. A proba tratará de medir se o alumno adquiriu os conceptos fundamentais de programación e adestrouse o suficiente como para posuír as habilidades precisas para resolver supostos prácticos. O alumno poderá facer uso do ordenador para, ademais de contestar ás preguntas, consultar dúbidas acerca da sintaxe concreta de algún comando.
Foro virtual	
Solución de problemas	Esta actividade suporá o estudo de casos prácticos e exemplos ademais da realización de distintos exercicios de programación. Co fin de afianzar os conceptos teóricos presentaranse supostos prácticos, que nun principio serán resoltos polo profesor para que orienten os alumnos. A medida que se avance no desenvolvemento teórico formularase a resolución de problemas por parte dos alumnos. A proposta de actividades estará dispoñible ao alumno con suficiente antelación. O labor do profesor será a supervisión solucionando dúbidas e corrigindo erros de interpretación, malos hábitos de programación, erros de sintaxe, etc.

Atención personalizada	
Metodoloxías	Descrición
Solución de problemas	É fundamental a atención ao alumno para resolver cantas dúbidas de concepto ou de procedemento poidan xurdir durante a resolución dos supostos prácticos. Prestarase especial atención a aqueles alumnos que presenten maiores dificultades na súa aprendizaxe co fin de que o seu progreso non se vexa retardado respecto ao xeral do resto de estudantes.  As titorías realizaranse co apoio das ferramentas de comunicación dispoñibles (Teams, Moodle e correo electrónico)

Avaliación			
Metodoloxías	Competencias / Resultados	Descrición	Cualificación
Solución de problemas	A3 B1 B5 B8 C3 C6	Realización obrigatoria. É necesario entregar todos os traballos para superar a materia.	50
Proba mixta	A3	Realización obrigatoria. Necesario aprobar o exame para superar a materia. O exame constará dunha parte tipo test (40% da nota final) e unha parte práctica (60%).	45
Foro virtual	B1 B5 C3 C6	Seguemento continuado e obxectivable dunha participación activa no foro da materia.	5

Observacións avaliación



## Non presentado

Terá a condición de Non presentado (NP)

quen non presente ningún traballo práctico nin concorra á proba

obxectiva no período oficial de avaliación. Por conseguinte, quen presente calquera traballo práctico e/ou realice a

proba obxectiva considerarase "Presentado" e será avaliado. Traballos prácticos

No caso de existir nalgún tema, as actividades de gamificación substituirán ás entregas de prácticas e suporán a mesma cualificación na nota

global. Soamente os alumnos con cualificación de NON PRESENTADO na primeira

oportunidade poderán entregar os traballos propostos durante o curso

para a segunda oportunidade. En caso de SUSPENSO na primeira

oportunidade, só se poderán entregar de novo os traballos suspensos que

sexan así calificados polo profesor. O retraso na entrega dos

traballos levará consigo unha penalización na nota que aparecerá

recollida na planificación docente na páxina web. En ningún caso as notas dos traballos gárdanse para os seguintes cursos académicos. Oportunidade adiantada

A avaliación basearase exclusivamente na proba obxectiva. 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 obrigación do estudante comunicar a súa situación ao

profesorado. Cualificación exame

Os alumnos farán unha

proba escrita ao finalizar o cuadrimestre de acordo ao calendario oficial.

O exame constará dunha parte tipo test (40% da nota final) e unha parte

práctica (60%) de realización de pequenos programas. Nesta segunda parte o

alumno poderá consultar o manual de Python. Alumnos de segunda matrícula e posteriores

A avaliación basearase no recollido nesta guía. Dada a posibilidade de

non asistir presencialmente por incompatibilidade cos horarios de

segundo curso, realizarán a maiores unha serie de traballos prácticos

ademais dos propostos para os alumnos de primeira matrícula. Neste caso

exisirá asistencia a titorías, bien presencialmente ou virtualmente. Outras observacións Todos os aspectos relacionados con ?dispensa académica?,

?dedicación ao estudo?, ?permanencia? e ?fraude académica? rexeranse de acordo coa normativa académica vixente da UDC.

## Fontes de información

### Bibliografía básica

- Jesús J. García Molina, Francisco J. Montoya Dato, José L. Fernández Alemán, M<sup>a</sup> José Majado Rosales (2005).

Una introducción a la programación : un enfoque algorítmico. Thomson

- Luis Joyanes Aguilar (2008). Fundamentos de programación : algoritmos, estructuras de datos y objetos. McGraw Hill

- Raúl González Duque (). Python PARA TODOS.

<http://edge.launchpad.net/improve-python-spanish-doc/0.4/0.4.0/+download/Python%20para%20todos.pdf>

- Mark Lutz (2013). Learning Python, Fifth Edition. O'Reilly Media, Inc

- Vernon L Ceder (2010). The quick Python book. Greenwich : Manning

- Ljubomir Perkovic (2015). Introduction to Computing Using Python: An Application Development Focus, 2nd Edition.

Wiley

### Bibliografía complementaria

- Bill Lubanovic (2014). Introducing Python: Modern Computing in Simple Packages. O'Reilly Media

- Mitchell L Model (2009). Bioinformatics Programming Using Python. O'Reilly Media

## Recomendacións

### Materias que se recomenda ter cursado previamente



Materias que se recomenda cursar simultaneamente
--

Materias que continúan o temario
----------------------------------

Introdución ás bases de datos/614522002

Estruturas de datos e algoritmia para secuencias biolóxicas/614522013

Observacións
--------------

Segundo se recolle nas distintas normativas de aplicación para a docencia universitaria deberase incorporar a perspectiva de xénero nesta materia (usarase linguaxe non sexista, utilizarase bibliografía de autores/as de ambos sexos, propiciarse a intervención en clase de alumnos e alumnas...) Traballarase para identificar e modificar prexuízos e actitudes sexistas e influirase na contorna para modificalos e fomentar valores de respecto e igualdade. Deberanse detectar situacións de discriminación por razón de xénero e proporanse accións e medidas para corrixilas.

(\*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