Guia docente					
Datos Identificativos					2019/20
Asignatura (*)	Introducción a la programación			Código	614522001
Titulación	Mestrado Universitario en Bioinformática para Ciencias da Saúde				
Descriptores					
Ciclo	Periodo	Curso		Tipo	Créditos
Máster Oficial	1º cuatrimestre	Primero		Optativa	6
Idioma	Castellano				
Modalidad docente	Presencial				
Prerrequisitos					
Departamento	Ciencias da Computación e Tecnoloxías da InformaciónComputación				
Coordinador/a	Cabrero Canosa, Mariano Javier		Correo electrónico	mariano.cabrero@udc.es	
Profesorado	Cabrero Canosa, Mariano Javier		Correo electrónico	mariano.cabrero@udc.es	
Web	moodle.udc.es				
Descripción general	En esta asignatura se pretende que los estudiantes sin formación en programación adquieran las nociones básicas para la realización de programas. Se usará el lenguaje de programación Python y sobre él se estudiarán los diferentes tipos de datos que podemos usar y las estructuras de control básicas que se utilizan para realizar un programa software.				

Competencias / Resultados del título	
Código	Competencias / Resultados del título
A3	CE3 - Analizar, diseñar, desarrollar, implementar, verificar y documentar soluciones software eficientes sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales en el campo de la Bioinformática
B1	CB6 - Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación
B5	CB10 - Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida auto dirigido o autónomo.
B8	CG3 - Ser capaz de trabajar en un equipo, en especial de carácter interdisciplinar
C3	CT3 - 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	CT6 - 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 / Resultados del título		
Interiorizar las buenas prácticas de programación.	AP3	BP5 BP8	
Usar las estructuras de datos adecuadas y programar los algoritmos de manipulación para solucionar problemas reales.	AP3	BP1 BP8	
Capacidad para realizar programas sencillos en el computador empleando un lenguaje de alto nivel.	AP3	BP1 BP5 BP8	CP3 CP6
Ser capaz de diseñar, evaluar, comparar y analizar soluciones algorítmicas básicas a problemas usuales en Bioinformática.	AP3	BP1	CP6

Contenidos	
Tema	Subtema

1. Introducción	a. Algoritmos. Representación. Acciones primitivas/no primitivas b. Programas. Proceso de construcción c. Lenguajes de programación: máquina, bajo nivel, alto nivel d. Compiladores. Intérpretes e. Entornos de desarrollo y herramientas: Python
2. Conceptos básicos	a. Estructura de un programa b. Constantes, Variables. c. Tipos de datos: entero, real, lógico, carácter, ? d. Estructuras simples: listas (arrays), cadenas, ? e. Operadores y expresiones (aritméticas, lógicas) f. Declaración de variables e constantes g. Entrada y salida estándar
3. Sentencias de control	a. Secuencial b. Alternativa c. Repetitiva: while, for
4. Funciones	a. Definición, declaración y llamada de función b. El ámbito de las variables c. Paso de argumentos d. Recursividad e. Módulos
5. Ficheros	a. Apertura y cierre b. Lectura y escritura de datos c. Acceso directo a los datos
6. Introducción a estructuras abstractas	a. Listas b. Pilas c. Colas d. Árboles
7. Introducción a la orientación a objetos	a. Clases b. Objetos c. Propiedades d. Métodos e. Concepto de herencia
8. Excepciones	a. Tipos b. Captura c. Lanzamiento d. Creación
9. Librerías científicas en Python	a. SciPy b. NumPy c. Matplotlib d. BioPython

Planificación				
Metodologías / pruebas	Competencias / Resultados	Horas lectivas (presenciales y virtuales)	Horas trabajo autónomo	Horas totales
Sesión magistral	A3 B1 B5	15	30	45
Estudio de casos	A3 B1 B5	1	2	3
Prueba mixta	A3	3	15	18
Solución de problemas	A3 B8 C3 C6	20	60	80
Atención personalizada		4	0	4

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

Metodologías	
Metodologías	**Descripción**
Sesión magistral	Actividad presencial para exponer conceptos fundamentales de la materia. Consistirá en la exposición oral del profesor apoyada con medios multimedia. Durante la presentación se tratará de interactuar con el alumno formulando preguntas dirigidas con el fin de afianzar conceptos y facilitar el aprendizaje. La proporción de uso de esta metodología será mayor frente a estudio de casos cuando el número de estudiantes sea alto y será acordado con estos.
Estudio de casos	Actividad no presencial para ahondar en los conceptos fundamentales de la materia. Consistirá en el estudio personal del alumno, a través del material sugerido y proporcionado por el profesor. La proporción de uso de esta metodología será mayor frente a sesión magistral cuando el número de estudiantes sea bajo y será acordado con estos.
Prueba mixta	Evaluación sumativa del alumno mediante un examen escrito con una parte teórica con preguntas tipo test y una parte práctica para resolver pequeños problemas de programación. La prueba tratará de medir si el alumno adquirió los conceptos fundamentales de programación y se entrenó lo suficiente como para poseer las habilidades precisas para resolver supuestos prácticos. El alumno podrá hacer uso del ordenador para, además de contestar a las preguntas, consultar dudas acerca de la sintaxis concreta de algún comando.
Solución de problemas	Esta actividad supondrá el estudio de casos prácticos y ejemplos además de la realización de distintos ejercicios de programación. 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. La medida que se avance en el desarrollo teórico se formulará la resolución de problemas por parte de los alumnos. La propuesta de actividades estará disponible al alumno con suficiente antelación. La labor del profesor será la supervisión solucionando dudas y corrigiendo errores de interpretación, malos hábitos de programación, errores de sintaxis, etc.

Atención personalizada	
Metodologías	**Descripción**
Solución de problemas	Es fundamental la atención al alumno para resolvar cuantas dudas de concepto o de procedimiento puedan surgir durante la resolución de los supuestos prácticos. Se prestará especial atención a aquellos alumnos que presenten mayores dificultades en su aprendizaje con el fin de que su progreso no se vea ralentizado respecto al general del resto de estudiantes.

Evaluación			
Metodologías	**Competencias / Resultados**	**Descripción**	**Calificación**
Solución de problemas	A3 B8 C3 C6	Se valorará la participación del alumno así como la realización de diversos trabajos puntuables que se detallarán durante lo curso y que podrán resolverse en la clase o en la tutoría. No es necesario entregar todos los trabajos para aprobar, aunque sí para conseguir la máxima nota.	65
Prueba mixta	A3	Realización obligatoria. Necesario aprobar el examen para superar la materia. El examen constará de una parte tipo test (40% de la nota final) y una parte práctica (60%).	35

Observaciones evaluación

No presentado
- Tendrá la condición de "No presentado"
(NP) quien no presente ningún trabajo práctico ni concurra a la prueba
objetiva en el período oficial de evaluación. Por consiguiente, quien presente cualquier trabajo práctico y/o realice la prueba objetiva se considerará "Presentado" y será evaluado.
Trabajos prácticos
- Solamente los alumnos con calificación de NO PRESENTADO en la primera oportunidad podrán entregar los trabajos propuestos durante el curso para la segunda oportunidad. En caso de SUSPENSO en la primera
oportunidad, sólo se podrán entregar de nuevo los trabajos suspensos que
sean así reconocidos por el profesor.
- El retraso en la entrega de los
trabajos llevará consigo una penalización en la nota que aparecerá
recogida en la planificación docente en la página web.
- 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 cómo a lo que lo había facilitado, y por tanto la calificación de SUSPENSO en la convocatoria anual.
Primera y segunda oportunidad
- Las calificaciones obtenidas en actividades como solución de problemas serán válidas tan sólo para el curso académico en el que se realicen.
Oportunidad adelantada de Diciembre
- Para la evaluación de la oportunidad adelantada se aplicarán los mismos criterios.
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.
Calificación examen
Los alumnos harán una prueba escrita
al finalizar el cuatrimestre de acuerdo al calendario oficial. El examen constará de una parte tipo test (40% de la nota final) y una
parte práctica (60%) de realización de pequeños programas. En esta
segunda parte el alumno podrá consultar el manual de Python.
Alumnos de segunda matrícula y posteriores
- La evaluación se basará en lo recogido en esta guía. Dada la
posibilidad de no asistir presencialmente por incompatibilidad con los horarios
de segundo curso, se realizarán a mayores una serie de trabajos prácticos
además de los propuestos para los alumnos de primera matrícula. En este caso
se exigirá asistencia a tutorías, bien presencialmente o virtualmente.
* Normativa de evaluación, revisión y reclamación de las calificaciones de los estudios de grado y máster universitario, aprobadas por el Consejo de Gobierno de la Universidad da Coruña el 19 de diciembre de 2013.

Fuentes de información	
Básica	- Jesús J. García Molina, Francisco J. Montoya Dato, José L. Fernández Alemán, Mª 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). ntroduction to Computing Using Python: An Application Development Focus, 2nd Edition. Wiley

Complementária	- Bill Lubanovic (2014). Introducing Python: Modern Computing in Simple Packages. O'Reilly Media - Mitchell L Model (2009). Bioinformatics Programming Using Python. O'Reilly Media

Recomendaciones
Asignaturas que se recomienda haber cursado previamente
Asignaturas que se recomienda cursar simultáneamente
Asignaturas que continúan el temario
Introducción a las bases de datos/614522002 Estructuras de datos y algoritmia para secuencias biológicas/614522013
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