



Guía docente				
Datos Identificativos				2020/21
Asignatura (*)	Fundamentos de Informática		Código	610G04010
Titulación	Grao en Nanociencia e Nanotecnoloxía			
Descriptoros				
Ciclo	Periodo	Curso	Tipo	Créditos
Grado	2º cuatrimestre	Primero	Formación básica	6
Idioma	Castellano			
Modalidad docente	Híbrida			
Prerrequisitos				
Departamento	Ciencias da Computación e Tecnoloxías da Información			
Coordinador/a	Eiras Franco, Carlos	Correo electrónico	carlos.eiras.franco@udc.es	
Profesorado	Eiras Franco, Carlos	Correo electrónico	carlos.eiras.franco@udc.es	
	Romero Campo, Paula		paula.romero.campo@udc.es	
Web	moodle.udc.es			
Descripción general	Asignatura de formación básica que consta de 6 créditos y que introducirá al estudiante en las áreas fundamentales de esta materia. Los Sistemas Informáticos son sistemas capaces de almacenar y procesar información por medio de diferentes elementos software y hardware. Al concluir la asignatura, el estudiante habrá sido dotado con los conocimientos básicos en Sistemas Informáticos, necesarios para comprender y abordar con garantías asignaturas más avanzadas de cursos posteriores.			
Plan de contingencia	1. Modificaciones en los contenidos 2. Metodologías *Metodologías docentes que se mantienen *Metodologías docentes que se modifican 3. Mecanismos de atención personalizada al alumnado 4. Modificacines en la evaluación *Observaciones de evaluación: 5. Modificaciones de la bibliografía o webgrafía			

Competencias del título	
Código	Competencias del título
A7	CE7 - Interpretar los datos obtenidos mediante medidas experimentales y simulaciones, incluyendo el uso de herramientas informáticas, identificar su significado y relacionarlos con las teorías químicas, físicas o biológicas apropiadas.
A10	CE10 - Comprender la legislación en el ámbito del conocimiento y la aplicación de la Nanociencia y Nanotecnología. Aplicar principios éticos en este marco.
B2	CB2 - Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio
B3	CB3 - Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética
B4	CB4 - Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado
B5	CB5 - Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía



B6	CG1 - Aprender a aprender
B7	CG2 - Resolver problemas de forma efectiva.
B10	CG5 - Trabajar de forma colaborativa.
B11	CG6 - Comportarse con ética y responsabilidad social como ciudadano/a y como profesional.
B12	CG7 - Comunicarse de manera efectiva en un entorno de trabajo.
C1	CT1 - Expresarse correctamente, tanto de forma oral como escrita, en las lenguas oficiales de la comunidad autónoma
C2	CT2 - Dominar la expresión y la comprensión de forma oral y escrita de un idioma extranjero
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
C5	CT5 - Entender la importancia de la cultura emprendedora y conocer los medios al alcance de las personas emprendedoras
C6	CT6 - Adquirir habilidades para la vida y hábitos, rutinas y estilos de vida saludables
C7	CT7 - Desarrollar la capacidad de trabajar en equipos interdisciplinarios o transdisciplinarios, para ofrecer propuestas que contribuyan a un desarrollo sostenible ambiental, económico, político y social.
C8	CT8 - Valorar la importancia que tiene la investigación, la innovación y el desarrollo tecnológico en el avance socioeconómico y cultural de la sociedad
C9	CT9 - Tener la capacidad de gestionar tiempos y recursos: desarrollar planes, priorizar actividades, identificar las críticas, establecer plazos y cumplirlos

Resultados de aprendizaje			
Resultados de aprendizaje		Competencias del título	
Conocer y comprender la importancia de los objetivos de la programación. Conocer los aspectos generales de los lenguajes y paradigmas de programación. Conocer el pseudocódigo y la sintaxis del lenguaje Python utilizado para describir algoritmos y programas. Conocer los pasos para la realización de un programa y sus principales componentes. Conocer los tipos de datos básicos usados en el lenguaje Python. Conocer las estructuras de programación estructuras y las diferencias entre ellas. Conocer todos los aspectos relacionados con la realización de funciones y procedimientos.	A7	B2	C1
	A10	B3	C2
		B4	C3
		B5	C5
		B6	C6
		B7	C8
		B10	
		B11	
		B12	
Ser capaz de realizar el seguimiento de un algoritmo (en pseudocódigo) o programa (en lenguaje Python), explicar qué realiza y encontrar posibles errores. Ser capaz de resolver pequeños algoritmos y programas. A partir de la formulación de un problema de pequeña-mediana envergadura, saber realizar el programa para resolverlo, teniendo en cuenta los objetivos de la programación. Realizar la descomposición adecuada implementando las funciones y procedimientos necesarios correctamente. Conocer el potencial de las librerías de funciones libres y ser capaz de escribir programas valiéndose de ellas, adaptándolas a las necesidades del problema.	A7	B2	C3
		B3	C8
		B4	
		B5	
		B6	
		B7	
Tener autonomía para establecer configuraciones sencillas de equipos en lo referente a usuarios, permisos y redes.	A7	B2	C3
		B5	C5
		B6	C8
		B7	
Conocer los conceptos, principios y técnicas básicas relacionadas con las bases de datos. Tener capacidad de modelar y diseñar bases de datos relacionales. Tener capacidad de manejar bases de datos relacionales mediante la ejecución de sentencias en un lenguaje de consultas.	A7	B2	C3
		B5	C7
		B6	C9
		B7	

Contenidos	
Tema	Subtema
1. Sistemas operativos	- Sistemas de Ficheiros - Línea de comandos - Usuarios, perfiles y tipos de acceso



2. Introducción a la programación	- Qué es un lenguaje de programación - Tipos de datos y operadores - Sentencias de control - Funciones y procedimientos - Uso de librerías - Buenos hábitos de programación
3. Bases de datos relacionales	- Modelo relacional - Modelo entidad-relación - El lenguaje SQL
4. Redes	- Introducción y modelo de capas - Configuración de las capas de enlace y red

Planificación				
Metodologías / pruebas	Competencias	Horas presenciales	Horas no presenciales / trabajo autónomo	Horas totales
Prácticas a través de TIC	A7 B2 B6 B7 C3 C7 C8 C9	12	34	46
Seminario	A7 B2 B6 B7 C3 C8	8	8	16
Prueba mixta	A7 B2 B3 B4 B6 B7 B11 C3 C6 C9	2	0	2
Sesión magistral	A7 A10 B2 B3 B4 B5 B6 B7 B10 B11 B12 C1 C2 C3 C5 C6 C8	28	56	84
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
Prácticas a través de TIC	En las sesiones de prácticas el alumno realizará tareas de manejo de sistemas operativos, codificación de programas en lenguaje Python y diseño y manejo de bases de datos. Los enunciados de las prácticas se proporcionarán con la suficiente antelación para que los alumnos puedan aprovechar mejor su tiempo. Es misión del profesor supervisar el código generado por el alumno para resolver dudas, corregir malos estilos de programación y corregir errores.
Seminario	En las sesiones de seminario se realizarán ejercicios y prácticas con la finalidad de detectar en los alumnos lagunas de conocimiento en la materia impartida hasta ese momento, y dar las explicaciones y/o referencias necesarias para enmendarlas.
Prueba mixta	En el periodo de evaluación se realizará una prueba de conocimientos teóricos y de resolución de problemas prácticos para evaluar la adquisición de las competencias por parte del alumno.
Sesión magistral	En las sesiones de teoría el profesor describe los objetivos y contenidos de la materia, para dar una visión particular del tema a tratar y relacionarlo con otros dentro de la asignatura. Después se desarrolla el tema correspondiente mediante una sesión magistral, ayudándose de las herramientas técnicas disponibles, haciendo hincapié en ciertas cuestiones en las que el alumno debe profundizar su autoaprendizaje. El objetivo es que los alumnos adquieran los conocimientos informáticos necesarios para desarrollar adecuadamente su vida académica y profesional. Se utilizará Python como lenguaje de codificación.

Atención personalizada	
Metodologías	Descripción



Seminario	Tanto en las sesiones magistrales como en los laboratorios de prácticas y en las sesiones de seminario se llevará una atención personalizada del alumno, en distintos niveles según el tipo de clase, detectando el nivel de asimilación y comprensión de los temas explicados y las prácticas requeridas a implantar.
Prácticas a través de TIC	En las sesiones de seminario es donde se puede llegar más al alumno para conocer las lagunas que presente e indicarle el camino para cubrirlas. Los alumnos que tengan matrícula a tempo parcial deben hablar, al comienzo del curso, con el/los profesores encargados de su grupo.

Evaluación			
Metodologías	Competencias	Descripción	Calificación
Seminario	A7 B2 B6 B7 C3 C8	Las horas de seminarios contarán con mecanismos de evaluación de los progresos del alumno. Esta evaluación se reflejará en el 10% de la nota final.	10
Prácticas a través de TIC	A7 B2 B6 B7 C3 C7 C8 C9	A lo largo del curso se realizarán cuatro prácticas, cada una de las cuales tendrá un valor máximo del 10% de la nota final.	40
Prueba mixta	A7 B2 B3 B4 B6 B7 B11 C3 C6 C9	El examen final constará de cuestiones teóricas y de varios ejercicios de puesta en práctica de las competencias adquiridas. El mencionado examen final tendrá un valor máximo del 50% de la nota final.	50

Observaciones evaluación

Fuentes de información	
Básica	- Luis Joyanes Aguilar (2011). Fundamentos de programación: algoritmos, estructuras de datos y objetos. Madrid - Raúl González Duque (). Python para todos. http://edge.launchpad.net/improve-python-spanish-doc/0.4/0.4.0/+download/Python%20para%20todos.pdf - A. Silberschatz; H. Korth; S. Sudarshan (2019). Database System Concepts (7a edición). McGraw Hill - Alan Beaulieu (2009). Learning SQL (2ª Edición). O'Reilly - James F. Kurose, Keith W. Ross (2008). Redes de computadores: un enfoque descendente basado en Internet. Addison Wesley
Complementaria	- 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 - Mark Lutz (2013). Learning Python, Fifth Edition. O'Reilly Media, Inc

Recomendaciones
Asignaturas que se recomienda haber cursado previamente
Asignaturas que se recomienda cursar simultáneamente
Asignaturas que continúan el temario
Otros comentarios



Leer: Lea el tema a tratar antes de asistir a las sesiones teóricas. ¡Es muy importante! Atender:

Atienda en clase, no esté solo de cuerpo presente. Comprender:

Comprenda lo que se le dice en las sesiones de teoría y, si no, pregunte. Preguntar:

Pregunte todo lo que no comprenda, no se quede con dudas. Estudiar:

Estudie después de las sesiones, para retener lo comprendido. Practicar: Haga muchos ejercicios, los que se le pidan, sugieran, y otros por su cuenta, tanto en papel como en el ordenador.

(*) 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