

Facultad de Ciencias

GUÍA DOCENTE DE LA ASIGNATURA

G47 - Programación

Doble Grado en Física y Matemáticas
Básica. Curso 1

Grado en Matemáticas
Básica. Curso 1

Curso Académico 2022-2023

1. DATOS IDENTIFICATIVOS

Título/s	Doble Grado en Física y Matemáticas Grado en Matemáticas		Tipología y Curso	Básica. Curso 1 Básica. Curso 1	
Centro	Facultad de Ciencias				
Módulo / materia	MATERIA AFIN BÁSICA MÓDULO BÁSICO				
Código y denominación	G47 - Programación				
Créditos ECTS	6	Cuatrimestre	Cuatrimestral (2)		
Web					
Idioma de impartición	Español	English friendly	Sí	Forma de impartición	Presencial

Departamento	DPTO. INGENIERÍA INFORMÁTICA Y ELECTRÓNICA				
Profesor responsable	MICHAEL GONZALEZ HARBOUR				
E-mail	michael.gonzalez@unican.es				
Número despacho	Facultad de Ciencias. Planta: + 3. DESPACHO PROFESORES (3055)				
Otros profesores	JOSE JAVIER GUTIERREZ GARCIA JOSE IGNACIO ESPESO MARTINEZ JOSE CARLOS PALENCIA GUTIERREZ ADOLFO GARANDAL MARTIN				

2. CONOCIMIENTOS PREVIOS

Los propios del bachillerato.

3. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS DEL PLAN DE ESTUDIOS TRABAJADAS

Competencias Genéricas

(Conocer) Demostrar poseer y comprender conocimientos en el área de las Matemáticas a partir de la base de la educación secundaria general, a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia del estudio de las Matemáticas.

(Autonomía) Aprender de manera autónoma nuevos conocimientos y técnicas.

(Leer) Leer textos científicos escritos tanto en español como en inglés.

Competencias Específicas

(Comprender) Comprender y utilizar el lenguaje matemático.

(Modelizar) Proponer, analizar, validar e interpretar modelos de situaciones reales sencillas, utilizando las herramientas matemáticas más adecuadas a los fines que se persigan.

(Resolver) Resolver problemas de Matemáticas, mediante habilidades de cálculo básico y otros, planificando su resolución en función de las herramientas de que se disponga y de las restricciones de tiempo y recursos.

(Utilizar software) Utilizar aplicaciones informáticas de análisis estadístico, cálculo numérico y simbólico, visualización gráfica, optimización u otras para experimentar en Matemáticas y resolver problemas.

(Desarrollar programas) Desarrollar programas que resuelvan problemas matemáticos utilizando para cada caso el entorno computacional adecuado.

Competencias Básicas

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.

Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

3.1 RESULTADOS DE APRENDIZAJE

-- Ser capaz de diseñar y codificar algoritmos sencillos en un lenguaje de programación.

- Contar con conocimientos de programación orientada a objetos.

- Conocer algoritmos básicos aplicables a datos elementales y estructurados (tales como recorridos, búsquedas, ordenación)

- Ser capaz de utilizar un entorno de desarrollo para codificar y ejecutar programas.

- Conocer los componentes de un sistema operativo y saber utilizarlos a nivel de usuario mediante comandos o desde el entorno de programación.

4. OBJETIVOS

Objetivos concretos: Conocimientos

- Conocer y comprender la sintaxis y semántica de las expresiones e instrucciones de un lenguaje de programación imperativo.
- Conocer las principales construcciones algorítmicas: secuencia, alternativa, iteración y recursión
- Conocer el concepto de clase y objeto como elementos constituyentes de los programas
- Conocer y saber utilizar los tipos de datos elementales, las tablas y las matrices, y conocer algoritmos básicos para su manipulación (recorridos, búsquedas, ordenación sencilla).
- Conocer los conceptos de método y paso de parámetros.
- Conocer los principios de la modularidad y abstracción para crear módulos de programa sencillos
- Conocer mecanismos de gestión de errores y excepciones
- Adquirir conocimientos básicos de programación orientada a objetos
- Conocer los principios de la entrada/salida: interactiva y con ficheros
- Conocer un sistema operativo a nivel de usuario

Objetivos concretos: Habilidades

- Diseñar pequeños algoritmos usando una notación de pseudocódigo
- Ser capaz de codificar y probar pequeños algoritmos usando un lenguaje de programación imperativo
- Utilizar un sistema de desarrollo para editar, compilar y ejecutar programas
- Utilizar un sistema operativo a nivel de usuario
- Crear módulos de programa, separando las fases de diseño e implementación
- Codificar en un lenguaje orientado a objetos un diseño modular
- Implementar programas sencillos que sean fiables y fáciles de entender
- Utilizar módulos de programa predefinidos para hacer un programa más complejo
- Aplicar estrategias de prueba sencillas para un módulo de programa
- Saber documentar un proyecto de programación

5. MODALIDADES ORGANIZATIVAS Y MÉTODOS DOCENTES

ACTIVIDADES	HORAS DE LA ASIGNATURA
ACTIVIDADES PRESENCIALES	
HORAS DE CLASE (A)	
- Teoría (TE)	20
- Prácticas en Aula (PA)	10
- Prácticas de Laboratorio Experimental(PLE)	
- Prácticas de Laboratorio en Ordenador (PLO)	30
- Prácticas Clínicas (CL)	
Subtotal horas de clase	60
ACTIVIDADES DE SEGUIMIENTO (B)	
- Tutorías (TU)	7,5
- Evaluación (EV)	7,5
Subtotal actividades de seguimiento	15
Total actividades presenciales (A+B)	75
ACTIVIDADES NO PRESENCIALES	
Trabajo en grupo (TG)	5
Trabajo autónomo (TA)	70
Tutorías No Presenciales (TU-NP)	
Evaluación No Presencial (EV-NP)	
Total actividades no presenciales	75
HORAS TOTALES	150

6. ORGANIZACIÓN DOCENTE													
CONTENIDOS		TE	PA	PLE	PLO	CL	TU	EV	TG	TA	TU- NP	EV- NP	Semana
1	BLOQUE TEMÁTICO 1: Programación en Python	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1 a 13
1.1	0. Presentación de la asignatura	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1
1.2	1. Introducción a los lenguajes de programación Lenguajes de programación. Compiladores e intérpretes. El ciclo de vida del software. El lenguaje Python. Concepto de algoritmo. Encapsulamiento de datos y algoritmos. Estructura de un programa. Funciones. Estilo de codificación.	2,00	0,00	0,00	0,00	0,00	0,50	0,50	0,00	5,00	0,00	0,00	1,2
1.3	2. Datos y expresiones. Números. Operaciones y expresiones. Variables. Booleanos. Strings. Uso de funciones matemáticas. Variables y paso de parámetros. Listas y tuplas.	4,00	2,00	0,00	4,00	0,00	1,00	1,00	0,00	8,00	0,00	0,00	2,3
1.4	3. Clases. Concepto de clase y objeto. Definición de clases. Creación y uso de objetos. Atributos y métodos de instancia y de clase. Espacios de nombres. Módulos y paquetes.	2,00	1,00	0,00	4,00	0,00	0,50	0,50	0,00	5,00	0,00	0,00	4, 5
1.5	4. Estructuras algorítmicas. Introducción. Instrucción condicional. Instrucción condicional múltiple. Instrucciones de bucle. Recursión. Descripción de algoritmos mediante pseudocódigo.	3,00	1,00	0,00	4,00	0,00	0,50	0,50	0,00	8,00	0,00	0,00	6, 7
1.6	5. Estructuras de Datos. Tablas. Algoritmos de recorrido. Algoritmos de búsqueda. Conjuntos. Tablas multidimensionales. El paquete NumPy. Diccionarios. Tipos enumerados.	3,00	3,00	0,00	6,00	0,00	1,50	1,70	1,00	16,00	0,00	0,00	8, 9
1.7	6. Tratamiento de errores. Excepciones. Tratamiento de excepciones. Patrones de tratamiento de excepciones. Lanzar Excepciones. Usar nuestras propias excepciones. Acciones de limpieza.	2,00	1,00	0,00	2,00	0,00	0,50	0,60	1,00	8,00	0,00	0,00	10
1.8	7. Entrada/salida. Escritura de texto con formato. Lectura de números con formato. Ficheros. Lectura de ficheros de texto. Escritura de ficheros de texto.	2,00	1,00	0,00	2,00	0,00	0,80	1,00	1,00	8,00	0,00	0,00	11, 12
1.9	8. Herencia y polimorfismo. Jerarquía de clases. Herencia. Clases abstractas. Polimorfismo.	1,00	1,00	0,00	2,00	0,00	1,00	0,50	1,00	8,00	0,00	0,00	13

2	BLOQUE TEMÁTICO 2: Herramientas	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1 y 14-15
2.1	9. Uso de sistemas operativos. Uso de la memoria USB. Introducción. Sistemas operativos comunes. El sistema de ficheros. Uso de la memoria USB. El intérprete de órdenes. Ejecución de programas. Guiones (scripts).	0,00	0,00	0,00	4,00	0,00	0,60	0,70	1,00	2,00	0,00	0,00	0,00	14, 15
2.2	10. Uso de un entorno integrado de desarrollo de programas. Entorno de desarrollo de programas. Gestión de proyectos. Analizar, cargar y ejecutar el programa. La depuración. Generación de documentos.	0,00	0,00	0,00	2,00	0,00	0,60	0,50	0,00	2,00	0,00	0,00	0,00	1
TOTAL DE HORAS		20,00	10,00	0,00	30,00	0,00	7,50	7,50	5,00	70,00	0,00	0,00	0,00	
Esta organización tiene carácter orientativo.														

TE	Horas de teoría
PA	Horas de prácticas en aula
PLE	Horas de prácticas de laboratorio experimental
PLO	Horas de prácticas de laboratorio en ordenador
CL	Horas de prácticas clínicas
TU	Horas de tutoría
EV	Horas de evaluación
TG	Horas de trabajo en grupo
TA	Horas de trabajo autónomo
TU-NP	Tutorías No Presenciales
EV-NP	Evaluación No Presencial

7. MÉTODOS DE LA EVALUACIÓN

Descripción	Tipología	Eval. Final	Recuper.	%
Problemas	Otros	No	Sí	10,00
Calif. mínima	0,00			
Duración				
Fecha realización	A lo largo del curso			
Condiciones recuperación	Examen de prácticas en el periodo de recuperación			
Observaciones	Se realizará una evaluación continuada de problemas resueltos en casa. En cada problema se requerirá un breve informe y una presentación al grupo durante la que se debatirá sobre la solución y otras posibles alternativas. Las presentaciones al grupo y la participación en el debate serán evaluables y servirán para subir o bajar la nota obtenida en el informe. Se entiende que la no asistencia a clase impide participar en los debates y por ello bajará la nota obtenida en el informe. En cada informe habrá ejercicios evaluables y otros no evaluables.			
Examen final	Examen escrito	Sí	Sí	50,00
Calif. mínima	4,00			
Duración	4 horas			
Fecha realización	junio			
Condiciones recuperación	Examen de recuperación			
Observaciones	Se realizarán dos exámenes escritos de cuestiones y problemas, uno en el periodo de evaluación ordinario y uno en el periodo de recuperación. Duración: 4 horas, en dos partes: una de cuestiones teórico-prácticas, y la otra para resolver un problema de programación. Se pueden llevar apuntes y libros a ambas partes. No se pueden llevar dispositivos electrónicos tales como computadores, tabletas, móviles, etc. Cada parte es la mitad de la nota del examen.			
Participación en clase de teoría	Otros	No	No	5,00
Calif. mínima	0,00			
Duración	Todo el curso			
Fecha realización	Evaluación continuada			
Condiciones recuperación				
Observaciones	Teoría: Se realizará una evaluación continuada basada en: pruebas breves (escritas o en Moodle) y participación en actividades colaborativas en Moodle			
Prácticas	Evaluación en laboratorio	No	Sí	35,00
Calif. mínima	4,00			
Duración				
Fecha realización	A lo largo del curso			
Condiciones recuperación	Examen de prácticas en el periodo de recuperación			

Observaciones	<p>Prácticas:</p> <p>Habrán cuatro prácticas evaluadas durante el transcurso de la asignatura. Se comunicará si una práctica es evaluada pasada la fecha de presentación.</p> <p>La media de las tres mejores calificaciones será la nota de prácticas.</p> <p>Para cada práctica se requerirá una breve memoria. Para poder aprobar las prácticas en el periodo ordinario se requiere haber entregado al menos todas las prácticas excepto una, con al menos un funcionamiento básico.</p> <p>Los criterios en las evaluaciones de los informes de las prácticas serán: Estilo de programación, eficiencia, claridad, documentación. Organización del informe, claridad en la exposición. Plazo de entrega: al evaluar una práctica se aplicarán penalizaciones por los retrasos en la entrega de los informes de las prácticas: 0.5 puntos por cada entrega retrasada hasta 1 semana, y 1 punto por cada entrega retrasada más de una semana.</p> <p>Si en el momento de evaluar una práctica aún no se ha entregado, además de aplicar las penalizaciones se evaluará otra práctica cercana en el tiempo, a criterio del profesor.</p> <p>La nota mínima de 4 para aprobar las prácticas se refiere a la media ponderada de los problemas y las prácticas.</p> <p>El examen de recuperación será de 3 horas de duración. En él habrá que resolver un ejercicio práctico usando los computadores del laboratorio. Se pueden llevar apuntes y libros. No se pueden llevar dispositivos electrónicos propios tales como computadores, tabletas, móviles, etc. Para poder presentarse a este examen se deberán entregar las memorias de todas las prácticas obligatorias de la asignatura menos una (tanto las evaluables como no las evaluables). Este examen tiene una calificación mínima de 4.</p>
<p>TOTAL 100,00</p>	
<p>Observaciones</p>	
<p>La nota de la asignatura consta de tres partes:</p> <p>a) participación en clase de teoría: 5% de la nota b) evaluación continua de problemas y prácticas: 45% de la nota Esta parte b) de la evaluación consta a su vez de dos partes: b.1) Problemas (10% de la nota) b.2) Prácticas (35% de la nota) c) Examen final: 50% de la nota</p> <p>La evaluación continua de la parte 'b):problemas y prácticas' se podrá superar en el periodo de recuperación mediante un examen de prácticas.</p> <p>Para superar la asignatura es preciso superar con una nota mínima de 4 tanto la parte b) de problemas y prácticas, como la parte c) del examen final. En caso de aprobar únicamente una de las dos partes en el periodo ordinario, se guardará la nota de esa parte para el periodo de recuperación.</p> <p>La realización de prácticas, problemas y exámenes es individual. Se considera realización fraudulenta:</p> <ul style="list-style-type: none"> - hacer problemas o prácticas en grupo - intercambiar ejercicios entre compañeros - que otras personas hagan los ejercicios o la mayor parte de estos 	
<p>Criterios de evaluación para estudiantes a tiempo parcial</p>	

Para los estudiantes a tiempo parcial la falta de asistencia a las clases de problemas no tendrá penalización.

La evaluación continua de problemas y prácticas se podrá superar por los estudiantes a tiempo parcial mediante un examen de prácticas (45%), tanto en el periodo ordinario como en el periodo de recuperación.

Para estos alumnos el peso del examen de teoría será del 55% y no habrá evaluación de la participación en clase de teoría.

8. BIBLIOGRAFÍA Y MATERIALES DIDÁCTICOS

BÁSICA

Title: Python in a Nutshell: A Desktop Quick Reference 3rd Edition
by: Alex Martelli (Author), Anna Ravenscroft (Author), Steve Holden
Publisher: O'Reilly Media; 3 edition (May 4, 2017)
ISBN-10: 144939292X
ISBN-13: 978-1449392925

Tutorial de python 3:
<https://docs.python.org/3/tutorial/>
<http://docs.python.org.ar/tutorial/3/index.html>

Title: Introducción a la programación con Python 3
By: Andrés Marzal Varó, Isabel Gracia Luengo, Pedro García Sevilla
Editor: Universitat Jaume I, 2014
ISBN: 978-84-697-1178-1
<http://repositori.uji.es/xmlui/bitstream/handle/10234/102653/s93.pdf>

Complementaria
<p>Title: Python Crash Course: A Hands-On, Project-Based Introduction to Programming - 25 nov 2015 By: Eric Matthes Editor: No Starch Press; Edición: 1 (25 de noviembre de 2015) ISBN-10: 1593276036 ISBN-13: 978-1593276034</p>
<p>Title: Learning Python, 4th Edition By: Mark Lutz Publisher: O'Reilly Media Print: October 2009 Ebook: September 2009 Pages: 1216 Print ISBN: 978-0-596-15806-4 ISBN 10: 0-596-15806-8 Ebook ISBN: 978-0-596-80598-2 ISBN 10: 0-596-80598-5</p>
<p>Title: Python 3. Los fundamentos del lenguaje - 2ª edición Tapa blanda – 20 dic 2016 By: Sébastien Chazallet Publisher: Eni ISBN-10: 2409006140 ISBN-13: 978-2409006142</p>
<p>Title: Phyton. Paso A Paso Tapa blanda – 1 mar 2016 By: ANGEL PABLO HINOJOSA GUTIERREZ Editor: RA-MA S.A. Editorial y Publicaciones; Edición: 1ª ed., 1ª imp. (1 de marzo de 2016) ISBN-10: 8499646115 ISBN-13: 978-8499646114</p>
<p>Title: Aprenda a programar con Python 3 By: Zed A. Shaw Editor: ANAYA MULTIMEDIA; Edición: edición (19 de octubre de 2017) Colección: Títulos Especiales ISBN-10: 8441539413 ISBN-13: 978-8441539419</p>
<p>Title: How to Make Mistakes in Python By: Mike Pirnat Editor: O'Reilly, October 2015 ISBN139781491934470 https://www.oreilly.com/programming/free/how-to-make-mistakes-in-python.csp</p>

9. SOFTWARE				
PROGRAMA / APLICACIÓN	CENTRO	PLANTA	SALA	HORARIO
Spyder 5.1 o superior: Entorno de desarrollo para Python 3.9 o superior.	Facultad de Ciencias			
Sistema operativo Linux con descompresor zip, editor de texto gedit, editor de texto emacs, shell bash	Facultad de Ciencias			
OpenOffice/LibreOffice Writer	Facultad de Ciencias			
Librerías, paquetes y herramientas de Python: numpy, scipy, matplotlib, pip, tkinter, skimage, imageio	Facultad de Ciencias			
Herramienta pydoc para documentación de Python.	Facultad de Ciencias			

10. COMPETENCIAS LINGÜÍSTICAS	
<input checked="" type="checkbox"/> Comprensión escrita	<input type="checkbox"/> Comprensión oral
<input type="checkbox"/> Expresión escrita	<input type="checkbox"/> Expresión oral
<input type="checkbox"/> Asignatura íntegramente desarrollada en inglés	
Observaciones	
Lectura de documentación técnica	