Facultad de Ciencias

GUIA DOCENTE DE LA ASIGNATURA

360 - Programación Avanzada

Máster Universitario en Matemáticas y Computación
Obligatoria. Curso 1

Curso Académico 2026 - 2027

1. DATOS IDENTIFICATIVOS					
Título/s	Máster Universitario en Matemáticas y Computación			Tipología y curso	Obligatoria. Curso 1
Centro	Facultad de Ciencias				
Módulo / materia	ELEMENTOS DE MATEMÁTICAS Y COMPUTACIÓN				
Código y denominación	360 - Programación Avanzada				
Créditos ECTS	3	Cuatrimestre		Cuatrimestral (1)	
Web	https://personales.unican.es/bernardp/programacion/index.html				
Idioma de impartición	Español	English Friendly	Sí	Forma de impartición	Presencial

Departamento	DPTO. MATEMATICA APLICADA Y CIENCIAS DE LA COMPUTACION
Profesor responsable	PILAR BERNARDOS LLORENTE
E-mail	pilar.bernardos@unican.es
Número despacho	E.T.S. de Ingenieros Industriales y de Telecomunicación. Planta: - 5. DESPACHO (S5018)
Otros profesores	JAVIER GONZALEZ VILLA

2. CONOCIMIENTOS PREVIOS
Programación básica

3. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS DEL PLAN DE ESTUDIOS TRABAJADAS
Competencias Genéricas
Capacidad para manejar las principales técnicas de computación científica
Competencias Específicas
Aplicar, analizar, diseñar y/o implementar algoritmos eficientes orientados a situaciones que admiten una modelización matemática.
Diseñar e implementar estructuras de datos adaptadas a la programación orientada a objetos.
Competencias Básicas
Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio
Competencias Transversales
Identificación de las fuentes y recursos de información relevantes para el tema seleccionado.

3.1 RESULTADOS DE APRENDIZAJE
Preparar conjuntos de prueba de software
Aplicar técnicas de programación orientada a objetos
Aplicar técnicas de programación funcional
Utilización de las principales estructuras de datos
Utilizar operaciones vectorizadas
Conocer técnicas de paralelismo

4. OBJETIVOS
Utilizar técnicas de programación avanzada
Utilizar estructuras de datos en programas orientados a objetos
Generalizar algoritmos y estructuras tipo a nuevos problemas
Localizar la información relevante de paquetes y librerías para su uso correcto en programas

5. MODALIDADES ORGANIZATIVAS Y MÉTODOS DOCENTES	
ACTIVIDADES	HORAS DE LA ASIGNATURA
ACTIVIDADES PRESENCIALES	
HORAS DE CLASE (A)	
- Teoría (TE)	10
- Prácticas en Aula (PA)	
- Prácticas de Laboratorio Experimental (PLE)	
- Prácticas de Laboratorio en Ordenador (PLO)	20
- Prácticas Clínicas (CL)	
Subtotal horas de clase	30
ACTIVIDADES DE SEGUIMIENTO (B)	

- Tutorías (TU)	7,5
- Evaluacion (EV)	3
Subtotal actividades de seguimiento	10,5
Total actividades presenciales (A+B)	40,5
ACTIVIDADES NO PRESENCIALES	
Trabajo en grupo (TG)	4
Trabajo autónomo (TA)	30,5
Tutorías No Presenciales (TU-NP)	
Evaluacion No Presencial (EV-NP)	
Total actividades no presenciales	34,5
HORAS TOTALES	75

6. ORGANIZACIÓN DOCENTE													
	CONTENIDOS	TE	PA	PLE	PLO	CL	TU	EV	TG	TA	TU-NP	EV-NP	Semana
1	Introducción a Python. Programación orientada a pruebas.	2,00	0,00	0,00	3,00	0,00	1,00	0,25	0,50	3,00	0,00	0,00	1-2
2	Programación orientada a objetos	2,00	0,00	0,00	5,00	0,00	2,00	0,75	0,75	7,00	0,00	0,00	2-3
3	Operaciones vectoriales. Paquete numpy	1,00	0,00	0,00	3,00	0,00	1,00	0,25	0,50	3,00	0,00	0,00	3-4
4	Programación funcional	2,00	0,00	0,00	5,00	0,00	1,50	0,75	0,75	7,00	0,00	0,00	4-5
5	Estructuras de datos	2,00	0,00	0,00	4,00	0,00	1,00	0,50	1,00	6,00	0,00	0,00	6-7
6	Aumento de prestaciones. Paralelismo	1,00	0,00	0,00	0,00	0,00	1,00	0,50	0,50	4,50	0,00	0,00	8
TOTAL DE HORAS		10,00	0,00	0,00	20,00	0,00	7,50	3,00	4,00	30,50	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.	%
Ejercicios y programas	Otros	No	Sí	40,00

Calif. mínima	0,00
Duración	
Fecha realización	A lo largo del curso
Condiciones recuperación	
Observaciones	Podrá recuperarse aumentando el trabajo de la asignatura

Controles parciales	Evaluación en laboratorio	No	Sí	60,00

Calif. mínima	0,00
Duración	
Fecha realización	Durante el curso. Recuperación al final de las clases
Condiciones recuperación	
Observaciones	

TOTAL	100,00

Observaciones

Si por medidas sobrevenidas de rango superior fuese imposible la evaluación presencial, se modificarán pesos y forma de las actividades de evaluación, pudiéndose suprimir alguna.

Criterios de evaluación para estudiantes a tiempo parcial

Quienes prevean y avisen a principio de curso de la dificultad de asistir a clase y/o realizar los ejercicios, podrán evaluarse al 100% con un único examen.

8. BIBLIOGRAFÍA Y MATERIALES DIDÁCTICOS

BÁSICA

Se proporcionarán materiales a lo largo del curso. El apoyo principal es la documentación oficial: https://docs.python.org/es/3/ (visitado el 20-5-25)

Complementaria

Kent D. Lee y Steve Hubbard
Data Structures and Algorithms with Python
Springer 2015
ISBN 978-3-319-13071-2 ISBN 978-3-319-13072-9 (eBook)

Hunt, John
Advanced guide to Python 3 programming
Cham, Switzerland : Springer Nature Switzerland AG, [2019]©2019.
ISBN: 978-3-030-25942-6

Lott, Steven F..
Mastering Object-oriented Python,
Packt Publishing, Limited, 2014
ISBN 978-1-78328-097-1

M. Scott Shell
An Introduction to NumPy and Scipy
2019
URL (20-5-2025): https://sites.engineering.ucsb.edu/~shell/che210d/numpy.pdf

An introduction to Python programming for scientists and engineers / Johnny Wei-Bing Lin, Hannah Aizenman, Erin Manette Cartas Espinel, Kim Gunnerson, Joanne Liu. -- Cambridge : Cambridge University Press, 2022.
ISBN 978-1-108-70112-9
BUC IND Monografías MP P C61E 216
https://catalogo.unican.es/cgi-bin/abnetopac?TITN=442733

Testing Python® - Applying Unit Testing, TDD, BDD, and Acceptance Testing
por Sale, David
2014 (https://go.exlibris.link/50T7rzVk)
Accesible online en Unican: https://elibro.net/es/ereader/unican/182725

R. Trobec et al.,
Introduction to Parallel Computing,
Springer Nature Switzerland AG 2018
URL (20-5-2025) https://link.springer.com/book/10.1007/978-3-319-98833-7

9. SOFTWARE

PROGRAMA / APLICACIÓN	CENTRO	PLANTA	SALA	HORARIO
Python+numpy				

10. COMPETENCIAS LINGÜÍSTICAS

☑ Comprensión escrita ☐ Comprensión oral

☐ Expresión escrita ☐ Expresión oral

☐ Asignatura íntegramente desarrollada en inglés

Observaciones