



UNIVERSIDAD
POLITÉCNICA
DE MADRID



GUÍA DE APRENDIZAJE

PROGRAMACIÓN

GRADO EN INGENIERIA Y SISTEMAS DE DATOS

CURSO ACADÉMICO	CÓDIGO	CURSO	CRÉDITOS	CARÁCTER	SEMESTRE
2026-27	95000503	1	6 ECTS	BÁSICA	PRIMER SEMESTRE
IDIOMA	CENTRO RESPONSABLE				
ESPAÑOL	E.T.S. DE INGENIEROS DE TELECOMUNICACION				
DEPARTAMENTO RESPONSABLE					
INGENIERÍA DE SISTEMAS TELEMÁTICOS					

COORDINADOR/A
ÁLVARO CARRERA BARROSO - a.carrera@upm.es

REQUISITOS OBLIGATORIOS

ASIGNATURAS REQUERIDAS

CONOCIMIENTOS PREVIOS RECOMENDADOS

ASIGNATURAS RECOMENDADAS

OTRAS RECOMENDACIONES

RESULTADOS

RD-1393/2007

Resultados de aprendizaje

- RA055** Comprender la importancia de los procedimientos de mantenimiento y actualización del software.
- RA054** Saber programar en un lenguaje de programación de ordenadores de uso extendido.
- RA056** Saber manejar los entornos y herramientas de desarrollo.

Competencias

- CB01** Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio
- CB02** 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
- CG05** Tener la capacidad de concebir y proponer soluciones creativas aplicando los métodos científico y de ingeniería para la definición y resolución de problemas formalizando los objetivos buscados y considerando los recursos disponibles.
- CE17** Que los estudiantes tengan la capacidad de utilizar los fundamentos de la programación, sistemas operativos, bases de datos, tecnología web y las redes y servicios de telecomunicación en proyectos de ingeniería de datos y sistemas.
- CE10** Que los estudiantes tengan la capacidad de construir sistemas, aplicaciones y servicios telemáticos, interconectados y multiplataforma a partir de su comprensión de la arquitectura web.
- CG04** Saber identificar y utilizar las herramientas de las Tecnologías de la Información y de las Comunicaciones más adecuadas para plantear y construir soluciones a problemas
- CB05** Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía
- CG09** Desarrollar la capacidad de aprendizaje a lo largo de la vida (lifelong learning) para adaptarse a un sector tecnológico en continua evolución.

CG03 Ser capaz de explicar de forma oral o escrita las soluciones planteadas para la resolución de un problema.

DESCRIPCIÓN

La programación es una herramienta básica para cualquier graduado en ingeniería. En concreto, en Ingeniería y Sistemas de Datos, los graduados deben ser capaces de superar retos tecnológicos en el contexto de sistemas que generan, analizan e intercambian cantidades masivas de datos.

La asignatura de Programación representa el primer contacto que tienen los estudiantes con esta disciplina que desarrollarán a lo largo de la titulación. En ella, se hace una introducción a la programación multiparadigma, y a los elementos del lenguaje de programación Python en particular. Python ha evolucionado para convertirse en uno de los lenguajes de programación más utilizados en ciencia e ingeniería de datos, por su flexibilidad, simplicidad y adecuación al procesamiento de grandes cantidades de datos. A la vez, se capacita al alumno en el uso de herramientas de desarrollo y depuración así como en el diseño y desarrollo incremental de proyectos.

La asignatura tiene 6 créditos ECTS oficiales. Esto se traduce en 162 horas de trabajo total, concentradas en unas 14 semanas (1 ECTS = 27 horas de trabajo, divididas en 10 horas de clase y 17 de trabajo individual del alumno). Este trabajo incluye la asistencia activa a las clases presenciales de grupo y de laboratorio, el estudio, realización de ejercicios y pruebas de autoevaluación, resolución de actividades y proyectos de laboratorio, y realización de las pruebas de evaluación progresiva.

La asignatura se imparte mediante b-learning, es decir, combinando la enseñanza presencial y la no presencial, para lo cual se utilizará el entorno virtual de aprendizaje Moodle.

TEMARIO

1. Introducción a la programación

- 1.1. Introducción a la asignatura
- 1.2. Definiciones y terminología
- 1.3. Introducción al lenguaje Python: Visión global y herramientas
- 1.4. La vida de un programa

2. Variables, Objetos, tipos y expresiones

- 2.1. Variables y objetos. Tipado dinámico
- 2.2. Objetos enteros y strings. Operadores

2.3. Legibilidad del código

2.4. Expresiones y sentencias

3. Estructuras de datos I. Listas y tuplas

3.1. Objetos mutables e inmutables

3.2. Definición y uso de listas

3.3. Definición y uso de tuplas

3.4. Recorrido de listas y tuplas con bucles for

4. Estructuras de control

4.1. Booleans y operadores lógicos

4.2. Sentencias if/elif/else

4.3. Bucles for y while

4.4. Interacción con el usuario usando bucles while

5. Funciones

5.1. Definición y uso de funciones

5.2. Definición y uso de argumentos

5.3. Resultado de una función

5.4. Módulos

6. Estructuras de datos II. Diccionarios

6.1. Definición y uso de diccionarios

6.2. Recorrido de diccionarios

6.3. Diccionarios anidados

7. Clases

7.1. Definición de clases y uso de objetos

7.2. Herencia y polimorfismo

7.3. Interacción entre clases

8. Ficheros

8.1. Lectura y escritura en disco

8.2. Persistencia de objetos: pickle

8.3. Ficheros de texto estructurados: CSV y JSON

8.4. Excepciones

9. Introducción a la ingeniería de software

9.1. Pruebas unitarias: unittest

9.2. Depuración de código

9.3. Gestión de proyectos y paquetes

9.4. Interactive computing con IPython

ACTIVIDADES

Actividad evaluable



Actividad formativa

 Clases teóricas de la asignatura - Temas 1-9				SEMANAS 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 E																
PRESENCIALIDAD	DURACIÓN TOTAL	LUGAR	GRUPO																	
Presencial	42h 00m	Aula (en horario)	Normal																	
METODOLOGÍA																				
Lección Magistral																				

 Laboratorios sobre Proyectos - Parte 1				SEMANAS 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 E																
PRESENCIALIDAD	DURACIÓN TOTAL	LUGAR	GRUPO																	
Presencial	06h 00m	Laboratorio	Reducido																	
METODOLOGÍA																				
Prácticas de Laboratorio																				



UNIVERSIDAD
POLITÉCNICA
DE MADRID



Laboratorios sobre Proyectos - Parte 2

SEMANAS

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

E

PRESENCIALIDAD

DURACIÓN TOTAL

LUGAR

GRUPO

Presencial

06h 00m

Laboratorio

Reducido

METODOLOGÍA

Prácticas de Laboratorio

Examen Parcial 1

SEMANAS

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

E

PRESENCIALIDAD

DURACIÓN TOTAL

LUGAR

EVALUACIÓN

PONDERACIÓN

NOTA MÍNIMA

Presencial

02h 00m

Aula de examen

Progresiva

20%

0

TEMAS

T1T2T3T4T5T6T7

METODOLOGÍA

Examen Escrito

RESULTADOS EVALUADOS

RA054RA055CB01CE17CE10CB02CB05CG03CG04CG05

Examen Parcial 2

SEMANAS

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

E

PRESENCIALIDAD

DURACIÓN TOTAL

LUGAR

EVALUACIÓN

PONDERACIÓN

NOTA MÍNIMA

Presencial

02h 00m

Aula de examen

Progresiva

60%

0

TEMAS

T1T2T3T4T5T7T6T8T9

METODOLOGÍA

Examen Escrito

RESULTADOS EVALUADOS

CE10CE17CB01CB02CB05CG03CG04CG05CG09RA055RA054



Examen Práctico

SEMANAS

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

E

PRESENCIALIDAD	DURACIÓN TOTAL	LUGAR	EVALUACIÓN	PONDERACIÓN	NOTA MÍNIMA
Presencial	02h 00m	Laboratorio	Progresiva	20%	0

TEMAS

T1

T2

T4

T3

T5

T6

T7

T8

T9

METODOLOGÍA

Examen de Prácticas

RESULTADOS EVALUADOS

RA054

RA055

RA056

CE10

CE17

CB01

CB02

CB05

CG03

CG04

CG05

CG09



Examen Global

SEMANAS

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

E

PRESENCIALIDAD	DURACIÓN TOTAL	LUGAR	EVALUACIÓN	PONDERACIÓN	NOTA MÍNIMA
Presencial	02h 00m	Aula de examen	Global	80%	0

TEMAS

T1

T2

T3

T4

T5

T6

T7

T8

T9

METODOLOGÍA

Examen Escrito

RESULTADOS EVALUADOS

RA054

RA055

CE10

CE17

CB01

CB02

CB05

CG03

CG04

CG05

CG09

Examen Global Práctico

SEMANAS

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

E

PRESENCIALIDAD

DURACIÓN TOTAL

LUGAR

EVALUACIÓN

PONDERACIÓN

NOTA MÍNIMA

Presencial

02h 00m

Laboratorio

Global

20%

0

TEMAS

T1T2T3T5T4T6T7T8T9

METODOLOGÍA

Examen de Prácticas

RESULTADOS EVALUADOS

RA054RA055RA056CE10CE17CB01CB02CB05CG03CG04CG05CG09



Examen Extraordinario

SEMANAS

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

E

PRESENCIALIDAD

DURACIÓN TOTAL

LUGAR

EVALUACIÓN

PONDERACIÓN

NOTA MÍNIMA

Presencial

02h 00m

Aula de examen

Extraordinaria

80%

0

TEMAS

T2T1T3T4T5T6T7T8T9

METODOLOGÍA

Examen Escrito

RESULTADOS EVALUADOS

RA054RA055CE10CE17CB01CB02CB05CG03CG04CG05CG09

Examen Extraordinario Práctico

SEMANAS

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

E

PRESENCIALIDAD

DURACIÓN TOTAL

LUGAR

EVALUACIÓN

PONDERACIÓN

NOTA MÍNIMA

Presencial

02h 00m

Laboratorio

Extraordinaria

20%

0

TEMAS

T6T5T4T3T2T1T7T8T9

METODOLOGÍA

Examen de Prácticas

RESULTADOS EVALUADOS

RA054RA055RA056CE10CE17CB01CB02CB05CG03CG04CG05CG09

PLANIFICACIÓN SEMANAL

Actividad evaluable

Actividad formativa

Semana con actividad

E

 Período de exámenes

Actividad	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	E
Clases teóricas de la asignatura - Temas 1-9	.	.	.	.	.	.		.	.	.	.	.	.			
Laboratorios sobre Proyectos - Parte 1				.	.	.										
Laboratorios sobre Proyectos - Parte 2								.	.	.	.					
Examen Parcial 1							.									
Examen Parcial 2																.

Actividad	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	E
 Examen Práctico													.			
 Examen Global																.
 Examen Global Práctico																.
 Examen Extraordinario																.
 Examen Extraordinario Práctico																.

CRITERIOS DE EVALUACIÓN

Criterios Generales

La evaluación de la asignatura comprobará la adquisición de las competencias y resultados de aprendizaje asociados a la programación estructurada y orientada a objetos, la resolución de problemas mediante algoritmos, el uso adecuado de estructuras de datos básicas y la capacidad para desarrollar, modificar, depurar y justificar programas en Python.

La calificación final de la asignatura se obtendrá a partir de dos componentes:

- Parte teórica: 80% de la calificación final.
- Parte práctica de laboratorio: 20% de la calificación final.

La parte teórica evaluará la comprensión de los conceptos de la asignatura y su aplicación a la resolución de problemas de programación. La parte práctica evaluará la capacidad del estudiante para trabajar sobre un proyecto software, comprender código existente, introducir modificaciones funcionales, corregir errores, compilar, ejecutar, probar y justificar las decisiones tomadas.

Para poder ser evaluado en la parte práctica, será requisito haber entregado en tiempo y forma las fases del proyecto de laboratorio establecidas durante el curso.

La detección de copia, plagio, uso no autorizado de herramientas externas o cualquier otra práctica fraudulenta en una prueba o entrega de evaluación supondrá el suspenso automático en la asignatura, sin perjuicio de las medidas disciplinarias que pudieran corresponder.

Cualquier prueba o entrega podrá requerir una comprobación oral complementaria por parte del profesorado, con el fin de verificar la autoría, la comprensión del trabajo presentado y la adquisición efectiva de las competencias evaluadas. Esta comprobación podrá aplicarse especialmente cuando existan indicios de uso no autorizado de sistemas de inteligencia artificial, generación automática de código o colaboración indebida.

Un estudiante se considerará presentado cuando participe en cualquiera de las pruebas que formen parte de la evaluación de la asignatura.

Criterios de Evaluación en Convocatoria Ordinaria. Evaluación progresiva

Criterios de evaluación

Durante el curso, el estudiante deberá trabajar de forma continuada a lo largo del cuatrimestre, asistiendo y participando en las clases teóricas y de laboratorio. A lo largo del curso podrán realizarse actividades interactivas mediante herramientas como Kahoot, Wooclap u otras equivalentes, con el objetivo de fomentar la participación activa y el aprendizaje continuo. La implicación del estudiante en estas dinámicas podrá ser valorada positivamente dentro de la evaluación progresiva.

Asimismo, el estudiante deberá entregar en fecha las fases del proyecto de laboratorio indicadas en los enunciados correspondientes:

- Semana 6: Proyecto, fase I.
- Semana 12: Proyecto, fase II.

La evaluación progresiva constará de las siguientes pruebas:

Parte teórica: 80% de la calificación final

- Semana 7: Examen Parcial 1, con una duración aproximada de 2 horas.
- Semana 15: Examen Parcial 2, con una duración aproximada de 2 horas.

La calificación de la parte teórica será la mayor de las dos siguientes opciones:

- Media ponderada de los dos exámenes parciales: Examen Parcial 1, 20%; Examen Parcial 2, 60%.
- Calificación del Examen Parcial 2, que pasará a representar el 80% de la calificación final, ignorándose en este caso la nota del Examen Parcial 1.

Parte práctica de laboratorio: 20% de la calificación final

- Semana 13: Prueba práctica individual de laboratorio, con una duración aproximada de 2 horas.

En la prueba práctica de laboratorio, el estudiante trabajará en Visual Studio Code sobre los proyectos desarrollados durante el curso. Durante la prueba deberá realizar modificaciones, ampliaciones, correcciones o adaptaciones sobre el código de sus proyectos, demostrando su capacidad para comprender el programa, modificarlo de forma autónoma, depurarlo, ejecutarlo correctamente y justificar técnicamente las decisiones adoptadas.

Para superar la asignatura mediante evaluación progresiva, el estudiante deberá obtener al menos 5,0 puntos sobre 10 en la media ponderada de las pruebas de evaluación, calculada de acuerdo con los porcentajes anteriores, y cumplir los requisitos de entrega de los proyectos de laboratorio.

Actividades

- Examen Parcial 1 (20%)
- Examen Parcial 2 (60%)
- Examen Práctico (20%)

| Criterios de Evaluación en Convocatoria Ordinaria. Prueba global**Criterios de evaluación**

Los estudiantes que no sigan la evaluación progresiva, o que no superen la asignatura por dicho itinerario, podrán acogerse a la evaluación global en la convocatoria ordinaria.

La evaluación global comprobará la adquisición de las mismas competencias y resultados de aprendizaje que la evaluación progresiva. Para ello, constará de dos partes:

- Prueba escrita de resolución de problemas y cuestiones teóricas: 80% de la calificación final.
- Prueba práctica individual de laboratorio: 20% de la calificación final.

La prueba escrita evaluará la comprensión de los contenidos teóricos de la asignatura y la capacidad para resolver problemas de programación.

La prueba práctica individual tendrá una duración aproximada de 2 horas y se realizará en Visual Studio Code. En ella, el estudiante deberá trabajar sobre los proyectos de laboratorio entregados, realizando modificaciones, ampliaciones, correcciones o adaptaciones solicitadas por el profesorado. Se evaluará la autonomía del estudiante, la comprensión del código, la corrección funcional de los cambios, la calidad de la solución y la capacidad para ejecutar y probar el programa.

Para poder realizar la prueba práctica, el estudiante deberá haber entregado previamente las fases del proyecto de laboratorio en los términos y plazos establecidos por el profesorado.

Para superar la asignatura mediante evaluación global, el estudiante deberá obtener al menos 5,0 puntos sobre 10 en la calificación final ponderada.

Actividades

- Examen Global (80%)
- Examen Global Práctico (20%)

Criterios de Evaluación en Convocatoria Extraordinaria

Criterios de evaluación

La convocatoria extraordinaria tendrá lugar en la fecha establecida por la Escuela. Podrán presentarse a ella los estudiantes que no hayan superado la asignatura en la convocatoria ordinaria.

La evaluación extraordinaria comprobará la adquisición de las competencias y resultados de aprendizaje de la asignatura mediante los mismos tipos de técnicas evaluativas utilizados en la convocatoria ordinaria. Por tanto, constará de dos partes:

- Prueba escrita de resolución de problemas y cuestiones teóricas: 80% de la calificación final.
- Prueba práctica individual de laboratorio: 20% de la calificación final.

Los estudiantes deberán entregar los proyectos de laboratorio antes de la fecha de realización de la prueba extraordinaria, en los términos indicados por el profesorado. La prueba práctica tendrá una duración aproximada de 2 horas y se realizará en Visual Studio Code, trabajando sobre los proyectos entregados. El estudiante deberá realizar modificaciones, ampliaciones, correcciones o adaptaciones solicitadas por el profesorado, demostrando que comprende el código presentado y que es capaz de evolucionarlo de forma autónoma.

Si durante la convocatoria ordinaria el estudiante obtuvo una calificación igual o superior a 5,0 puntos sobre 10 en la parte práctica de laboratorio, podrá optar por conservar dicha calificación para la convocatoria extraordinaria. En ese caso, únicamente deberá realizar la prueba escrita, que mantendrá un peso del 80% en la calificación final.

Para superar la asignatura en la convocatoria extraordinaria, el estudiante deberá obtener al menos 5,0 puntos sobre 10 en la calificación final ponderada.

Actividades

- Examen Extraordinario (80%)
- Examen Extraordinario Práctico (20%)

RECURSOS

Bibliografía

Python Crash Course

Autor: Eric Matthes Python Crash Course, 3rd Edition. O'Reilly, 2023. ISBN: 978-1718502703 Disponible online: <https://learning.oreilly.com/library/view/python-crash-course/9781098156664/> (accesible desde la red de la UPM o usando la VPN-UPM).

Web

Moodle

En la plataforma Moodle el alumno encontrará ejercicios propuestos y resueltos, enlaces interesantes y documentación de la asignatura con los contenidos específicos de la misma.

Bibliografía

Learning Python, 5th Edition

Autor: Mark Lutz Learning Python, 5th Edition O'Reilly Release Date: June 2013 ISBN: 9781449355739

Python Data Science Handbook: Tools and Techniques for Developers

Autor: Jake VanderPlas Python Data Science Handbook: Tools and Techniques for Developers O'Reilly 25 marzo 2016 ISBN: 9781491912058

Head First Python

Autor: Paul Barry Head First Python, 2nd edition (16 de diciembre de 2016) Editor: O'Reilly UK Ltd. ISBN: 9781491919538

Python Programming: An Introduction to Computer Science

Autor: John Zelle Python Programming: An Introduction to Computer Science, 3rd Edition (8 de agosto de 2016) Editor: Franklin, Beedle & Associates Inc ISBN: 9781590282755

OBJETIVOS DE DESARROLLO SOSTENIBLE



**Industria, innovación e
infraestructura**

OTRA INFORMACIÓN

La asignatura se ha diseñado de tal manera que su impartición sea flexible para adaptarse a diferentes situaciones. En principio, la asignatura se imparte en modo presencial, y es así como se ha descrito en esta guía. Por lo tanto, no se retransmitirán las clases presenciales, salvo por imperativos sanitarios.

En el caso de trasladar la docencia al modo mixto o modo no presencial:

- Tanto las clases presenciales, como las clases de laboratorio se pueden pasar a modo no presencial, con la misma temporización. Si hay algún alumno que no tiene ordenador portátil se puede gestionar el préstamo por parte de la Escuela.
- Igualmente, aunque las pruebas de evaluación se consideran los eventos más significativos de la asignatura para ser de impartición presencial, en caso de confinamiento se realizarán a través de la plataforma Moodle.

Esta asignatura de carácter técnico contribuye a alcanzar los Objetivos de Desarrollo Sostenible (ODS) (<https://sostenibilidad.upm.es/conoce-los-objetivos-de-desarrollo-sostenible/>); concretamente el "ODS9: Industria, innovación e infraestructura", ofreciendo una base de los conocimientos necesarios para desarrollar las diferentes metas de dicho objetivo, tales como: "9.1 Desarrollar infraestructuras fiables, sostenibles, resilientes y de calidad", o "9.5 Aumentar la investigación científica y mejorar la capacidad tecnológica industrial". Por lo tanto, las tareas de la asignatura se orientan a la consecución del ODS9 a través del aprendizaje de los alumnos en las diferentes habilidades. Dicho aprendizaje se da en el ámbito de los lenguajes de programación que son básicos para alcanzar los objetivos mencionados con anterioridad.