

Fecha de aprobación: 30/06/2025

Guía docente de la asignatura

Fundamentos de la Programación (23111M5)

Grado	Grado en Información y Documentación	Rama	Ciencias Sociales y Jurídicas				
Módulo	Tecnologías de la Información y Edición Digital	Materia	Fundamentos de Programación				
Curso	3º	Semestre	2º	Créditos	6	Tipo	Optativa

PRERREQUISITOS Y/O RECOMENDACIONES

Prerrequisitos no hay. No se supone conocimiento alguno sobre programación.
Recomendación: practicar lo aprendido en clase, no se estudia de memoria en el último momento.

En el caso de utilizar herramientas de IA para el desarrollo de la asignatura, el estudiante debe adoptar un uso ético y responsable de las mismas. Se deben seguir las recomendaciones contenidas en el documento de "Recomendaciones para el uso de la inteligencia artificial en la UGR" publicado en esta ubicación:

<https://ceprud.ugr.es/formacion-tic/inteligencia-artificial/recomendaciones-ia#contenido0>

BREVE DESCRIPCIÓN DE CONTENIDOS (Según memoria de verificación del Grado)

- Introducción a la programación.
- Algoritmos y programas.
- Conceptos generales de programación dirigida a objetos.
- Métodos y estructuras de control.
- Estructuras de datos: cadenas, listas, conjuntos, diccionarios
- Lectura y almacenamiento de datos en ficheros.
- Adquisición de pensamiento computacional (descomposición, algoritmo y abstracción) para la resolución de problemas.
- Prácticas de programación en sistemas de información.

COMPETENCIAS ASOCIADAS A MATERIA/ASIGNATURA

COMPETENCIAS GENERALES

- CG02 - Demostrar capacidad de organización y planificación
- CG05 - Tener conocimientos de informática relativos al ámbito de estudio



- CG07 - Resolver problemas
- CG08 - Tomar decisiones
- CG10 - Ser capaz de trabajar en un equipo de carácter interdisciplinar
- CG16 - Aprender de forma autónoma
- CG17 - Saber adaptarse a nuevas situaciones
- CG18 - Tener creatividad
- CG19 - Mostrar capacidades de liderazgo

COMPETENCIAS ESPECÍFICAS

- CE11 - Utilizar y poner en práctica métodos, técnicas y herramientas informáticas y de redes de comunicación (hardware y software) para la implantación, desarrollo y explotación de sistemas de información.

RESULTADOS DE APRENDIZAJE (Objetivos)

- Comprender los conceptos relacionados con el desarrollo de algoritmos y su aplicación a los sistemas de información.
- Conocer los diferentes paradigmas de programación.
- Saber utilizar herramientas de edición, compilación, y ejecución para desarrollar programas. Conocer las técnicas de depuración, prueba y corrección de programas.
- Saber escoger la mejor estructura de datos para resolver un problema.
- Ser capaz de manejar diferentes niveles de abstracción para estructurar el software a desarrollar.
- Diseñar e implementar algoritmos para la resolución de problemas.

PROGRAMA DE CONTENIDOS TEÓRICOS Y PRÁCTICOS

TEÓRICO

Tema 1. Introducción a la programación de ordenadores

- Resolución de problemas con ordenadores.
- Algoritmo.
- Metodología de la programación.
- Lenguajes de programación. Traductores.
- Entornos integrados de desarrollo (IDE).

Tema 2. Tipos de datos y estructuras de programación

- Datos, tipos de datos y expresiones.
- Operaciones de entrada, salida y asignación.
- Estructura secuencial.
- Estructuras condicionales.
- Estructuras repetitivas.
- Primeros programas.

Tema 3. Conceptos generales de programación dirigida a objetos

- Clase y objeto.
- Métodos.

Tema 4. Estructuras de datos

- Estructuras Secuenciales (string)
- Estructuras Asociativas (listas, diccionarios, ...)



Tema 5. Acceso a ficheros

- Conceptos generales sobre ficheros
- Lectura de ficheros
- Escritura en ficheros

Tema 6. Funciones y procedimientos.

- Programación modular.
- Funciones y procedimientos.
- Parámetros formales y actuales.

PRÁCTICO

PROGRAMA DE PRÁCTICAS

- Manejo de un IDE, primeros programas
- Condicionales en Python.
- Bucles en Python.
- Datos estructurados en Python.
- Programación modular en Python.
- Uso básico de ficheros en Python.

SEMINARIOS

Seminario 1. Diseño de soluciones a problemas mediante el ordenador.

Seminario 2. Alternativas, instalación y uso de IDEs.

Seminario 3. Buenas costumbres y convenios en la construcción de programas.

Seminario 4. Diseño de soluciones modulares a problemas mediante el ordenador.

Seminario 5. Acceso a bases de datos desde programas.

Seminario 6. Uso de herramientas de trabajo colaborativo.

BIBLIOGRAFÍA

BIBLIOGRAFÍA FUNDAMENTAL

- Python para todos. Raúl Gonzalez Duque. Creative Commons Atribución-NoComercial
- Andrés Marzal Varó, Isabel Gracia Luengo, Pedro García Sevilla. Introducción a la Programación con Python 3. Publicacions de la Universitat Jaume I, 2014.
<http://repositori.uji.es/xmlui/bitstream/10234/102653/1/s93.pdf>
- Python para informáticos. Explorando la información. Charles Severance
- Curso de Python para principiantes. Eugenia Bahit. Creative Commons Atribución-NoComercial.

BIBLIOGRAFÍA COMPLEMENTARIA

- Python for Everyone. C. Horstmann, R Necsaise en Wiley
- Langtangen, H.P., A Primer on Scientific Programming with Python(3ª ed), Springer, 2012.

ENLACES RECOMENDADOS

Python: <http://www.python.org/>

Para el IDE: <https://anaconda.org/>

<https://thonny.org/>



METODOLOGÍA DOCENTE

- MD01 - Clases de teoría (lección magistral)
- MD02 - Clases de problemas y/o de prácticas
- MD03 - Seminarios
- MD04 - Tutorías
- MD05 - Trabajo autónomo del alumnado

EVALUACIÓN (instrumentos de evaluación, criterios de evaluación y porcentaje sobre la calificación final)

EVALUACIÓN ORDINARIA

La nota final del alumno se calculará a partir de las calificaciones que obtenga en las siguientes partes:

PARTE TEÓRICA: La ponderación de esta parte es del 30%.

La evaluación de esta parte consta de distintas pruebas evaluables programadas durante el curso:

1. respuesta a cuestiones (10%)
2. cuestionarios (10%)
3. actividades realizables en el aula (10%)

PARTE PRÁCTICA: La ponderación de esta parte es del 70%.

La evaluación de esta parte se realizará mediante un examen multi-pregunta en ordenador sobre los contenidos de la materia impartida.

Este examen se realizará en su convocatoria oficial ordinaria.

Finalmente, para aquellos estudiantes que hayan asistido menos del 80% de las clases o bien no tengan aprobada la parte teórica (mediante evaluación continua), podrán ser evaluados de la parte teórica mediante unas preguntas adicionales, en el mismo acto académico de la convocatoria oficial.

EVALUACIÓN EXTRAORDINARIA

La evaluación en la convocatoria extraordinaria consistirá en una prueba teórico-práctica delante del ordenador, donde se evaluará tanto teoría como prácticas, la ponderación en este caso es teoría 30% y práctica 70%.

Para aquellos alumnos que en la convocatoria ordinaria siguieran el procedimiento de evaluación continua y cumplieran con la asistencia a las prácticas, si lo desean, podrán conservar su calificación de teoría para la convocatoria extraordinaria.

La suma de teoría, prácticas deberá ser 5 como mínimo para superar la asignatura.

EVALUACIÓN ÚNICA FINAL

Esta modalidad de evaluación se realizará en un único acto académico y consistirá en una prueba teórico-práctica delante del ordenador, donde se evaluará tanto teoría como prácticas, manteniendo el porcentaje de teoría 30% y de práctica 70%.

INFORMACIÓN ADICIONAL

Todas las clases se realizan en aula de ordenadores ya que tras las exposiciones breves de





conceptos en clase por parte del profesor se pone en práctica inmediatamente en ejercicios en el entorno de programación. Todas las resoluciones de problemas se hacen directamente en el entorno de programación.

Información de interés para estudiantado con discapacidad y/o Necesidades Específicas de Apoyo Educativo (NEAE): [Gestión de servicios y apoyos \(https://ve.ugr.es/servicios/atencion-social/estudiantes-con-discapacidad\)](https://ve.ugr.es/servicios/atencion-social/estudiantes-con-discapacidad).

