FICHA IDENTIFICATIVA

DATOS DE LA ASIGNATURA

Código: 34241
Nombre: Fundamentos de programación para la física
Ciclo: Grado
Créditos ECTS: 6
Curso académico: 2025-26

TITULACIONES

Titulación	Centro	Curso	Periodo
1105 - Grado en Física	Facultat de Física	1	Primer cuatrimestre

MATERIAS

Titulación	Materia	Carácter
1105 - Grado en Física	Informática	FORMACIÓN BÁSICA

COORDINACIÓN

FUERTES SEDER ARIADNA

GRIMALDO MORENO FRANCISCO

RESUMEN

Se trata de una asignatura considerada de formación básica de 6 créditos ECTS de duración cuatrimestral ubicada en el primer curso, primer cuatrimestre de la titulación por lo que no hay relación con otras materias previas de la titulación pero sí con el resto de materias dado que debe asentar las bases para que el estudiante sea capaz de conocer las posibilidades del computador como herramienta de cálculo, de simulación y de análisis de medidas experimentales en la Física y así pueda servirse del ordenador en la resolución de problemas conociendo sus usos potenciales y sus limitaciones.

El objetivo de esta materia consiste en proporcionar al estudiante una formación básica en programación, orientada a su aplicación en Física, de manera que se use el pensamiento computacional como metodología y como herramientas que le permita abordar posteriormente problemas progresivamente más complejos, tanto desde el punto de vista analítico como numérico en modelización de leyes físicas y simulaciones de comportamientos de sistemas físicos, y realizar análisis de datos experimentales obtenidos en los diferentes laboratorios del Grado en Física.

Por tanto, se trata de que el estudiante consiga un conocimiento suficiente del diseño de algoritmos mediante programación estructurada, así como de las estructuras de datos fundamentales y de este modo

aprenda a usar el ordenador como herramienta para la resolución de problemas físicos e ilustrar conceptos de matemáticas.

En lo que se refiere a la parte práctica, en esta asignatura se tratará de que el alumno adquiera habilidades de desarrollo de programas en un lenguaje de programación estructurado de propósito general y uso extendido, para ello se le propondrá como ejercicios a desarrollar la realización diferentes algoritmos y programas que resuelvan problemas de Física y de cálculo numérico de manera que adquiriera las nociones básicas para poder utilizar las herramientas de computación simbólica y métodos numéricos que necesitarán en otras materias del Grado en Física.

CONOCIMIENTOS PREVIOS

RELACIÓN CON OTRAS ASIGNATURAS DE LA MISMA TITULACIÓN

No se han especificado restricciones de matrícula con otras asignaturas del plan de estudios.

OTROS TIPOS DE REQUISITOS

Es aconsejable tener experiencia en el manejo de ordenadores personales, sistemas operativos y en el uso de programas informáticos básicos (como por ejemplo algún procesador de textos u hoja de cálculo).

COMPETENCIAS / RESULTADOS DE APRENDIZAJE

1105 - Grado en Física

Capacidad de aprendizaje: Ser capaz de iniciarse en nuevos campos de la Física y de la ciencia y tecnología en general, a través del estudio independiente.

Modelización y resolución de problemas: Ser capaz de identificar los elementos esenciales de un proceso/situación y de establecer un modelo de trabajo del mismo. Ser capaz de realizar las aproximaciones requeridas con el objeto de reducir un problema hasta un nivel manejable. Pensamiento crítico para construir modelos físicos.

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.

Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

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

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 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.

Resolución de problemas: Ser capaz de evaluar claramente los órdenes de magnitud, de desarrollar una percepción de las situaciones que son físicamente diferentes pero que muestran analogías, permitiendo, por lo tanto, el uso de soluciones conocidas a nuevos problemas.

Resolución de problemas y destrezas informáticas: Ser capaz de interpretar cálculos de forma independiente, incluso cuando sea necesario un pequeño PC o un gran ordenador, incluyendo el desarrollo de programas de software.

Saber aplicar los conocimientos adquiridos a la actividad profesional, saber resolver problemas y elaborar y defender argumentos, apoyándose en dichos conocimientos.

Ser capaz de reunir e interpretar datos relevantes para emitir juicios.

DESCRIPCIÓN DE CONTENIDOS

1. Introducción a los ordenadores y a la programación en Física

Conceptos básicos.

Estructura interna del computador: unidad de control, unidad aritmético-lógica, unidad de almacenamiento, unidad de entrada y unidad de salida.

Lenguajes y paradigmas de programación: lenguajes procedurales y lenguajes declarativos.

Sistema operativo.

Uso de la programación en Física

2. Programación en lenguajes de alto nivel

Algoritmos.

Características de los lenguajes de programación de alto nivel: Objetos y referencias, Tipo simples de datos, strings y listas y Entrada y Salida de datos.

Fases en la realización de un programa: Análisis del problema, Diseño del algoritmo y Programación del algoritmo. Ejemplos de problemas en Física.

3. Programación estructurada

Teorema de la programación estructurada.

Diseño de programas estructurados. Estructuras de control: Estructura secuencial, Estructura condicional y Estructura iterativa.

4. Ficheros

Conceptos básicos de archivos: Tipo de acceso, Ficheros lógicos y ficheros físicos y Ficheros binarios y de texto.

Procesamiento de ficheros.

5. Programación modular

Definición de modulo: Programación modular, Definición de subprogramas: Funciones, Parámetros de un subprograma y Ámbito de identificadores.

Recursividad.

Iniciación a la resolución mediante ordenador de problemas de física de complejidad moderada.

6. Tipos de datos estructurados

Tratamiento de cadenas y listas.

Colecciones.

Introducción a las Clases.

7. Prácticas de laboratorio de Informática

Introducción al entorno de trabajo

Construcción de programas. Tipos, Expresiones aritméticas y E/S

Estructuras condicionales

Estructuras de iterativas

Ficheros

Funciones

Cadenas

Listas

Registros

Listas de Registros

VOLUMEN DE TRABAJO (HORAS)

ACTIVIDADES PRESENCIALES

Actividad	Horas
Teoría	30,00
Laboratorio	30,00
Total horas	**60,00**

ACTIVIDADES NO PRESENCIALES

Actividad	Horas
Asistencia a otras actividades	0,00
Elaboración de trabajos individuales o en grupo	40,00
Estudio y trabajo autónomo	0,00
Preparación de clases	30,00
Preparación de actividades de evaluación	20,00
Resolución de casos prácticos	0,00
Total horas	**90,00**

METODOLOGÍA DOCENTE

Docencia presencial 40%:

Clases teórico prácticas: Se abordan los aspectos conceptuales y formales de la materia y la resolución de problemas o casos como aplicación de los conceptos teóricos. Se basan principalmente en la lección magistral dialogada y el uso de herramientas docentes como representación gráfica de soluciones,

proyección de presentaciones, programas de cálculo, etc.

Sesiones de laboratorio en el Aula de Informática: Manejo de paquetes informáticos estándares, realización y ejecución de programas en los que se codifican algoritmos sencillos. Resolución de dudas surgidas al enfrentarse a los conceptos teóricos, a la resolución de problemas y a la realización de programas. Refuerzo de aspectos en los que se encuentran mayores dificultades, y verificación del progreso del estudiante en la materia, asociado a una componente de evaluación continua.

Trabajo personal del estudiante 60%:

- Estudio de los fundamentos teóricos.

- Resolución de problemas, cuestiones tipo test, y trabajos (individualmente o en grupo).

- Realización de ejercicios mediante ordenador, interpretación, conclusiones y realización de memorias para su comunicación.

- Tutorías individuales: consultas puntuales del estudiante al docente sobre dudas y dificultades encontradas en el estudio y en la resolución de problemas, o discusión sobre temas de interés, bibliografía, etc.

re temas de interés, bibliografía, etc.

EVALUACIÓN

Los sistemas de evaluación son los siguientes:

El trabajo que se espera que los alumnos realicen lo podemos clasificar en 2 tipos:

1. Trabajo autónomo de auto-aprendizaje.

2. Trabajo supervisado.

1.- El trabajo autónomo de auto-aprendizaje consiste en las actividades que el alumno realiza fuera del horario de docencia, de manera independiente o dirigida por el profesor, y que está orientado a adquirir conocimientos, capacidades y destrezas de forma autodidacta. Principalmente serán actividades que el profesor indique que se deben realizar durante el curso (lecturas recomendadas, realización de problemas, investigación de ciertos temas, etc.) pero que no serán evaluadas directamente (calificadas) por el profesor, aunque tendrán incidencia en otras actividades que sí que serán evaluadas y calificadas por el profesor. No obstante, estas actividades podrán ser revisadas en las tutorías a petición de los alumnos.

2.- El trabajo supervisado consiste en todas las actividades que realizará el alumno a petición del profesor y que éste monitorizará para evaluar el grado de superación de la asignatura. Estos trabajos serán

problemas o actividades prácticas individuales que se realizarán en el aula de informática.

Las características de estos trabajos son:

¿ Que serán evaluados por el profesor.

¿ Tendrán fecha de entrega, o se realizarán de manera presencial en el momento de ser planteadas.

¿ Serán de realización obligatoria por parte del alumno para la aplicación de la evaluación continua.

Dentro de estos trabajos se encontrarán un conjunto de pruebas objetivas individuales que consistirán en la resolución de un problema sencillo mediante la programación de un algoritmo informático. (N_p_presencial).

Además, los alumnos tendrán una nota de prácticas correspondiente a las prácticas realizadas en el laboratorio y al trabajo o proyecto final propuesto. (N_continua).

Al finalizar el curso se realizará un examen escrito que abarcará tanto los conocimientos teóricos como prácticos. Con este examen se evaluará, por una parte, la comprensión de los aspectos teórico-conceptuales y el formalismo asociado, tanto de forma general como a través de cuestiones o casos particulares sencillos. Y por otra parte, se valorará la capacidad de resolución de problemas aplicando el formalismo, así como la capacidad crítica respecto a los resultados obtenidos. En ambas partes se valorarán una correcta argumentación y una adecuada justificación. (N_examen).

En primera convocatoria, la calificación final se obtendrá mediante la siguiente ponderación de las distintas actividades y pruebas:

Nota Final = 35% N_p_presencial + 15% N_continua + 50% N_examen

Hará falta sacar un 4 en el examen para poder promediar y un 5 o más como nota final para aprobar.

En segunda convocatoria sólo se tendrá en cuenta el examen realizado, el cual constará de una parte escrita y otra en el aula de informática (50% cada parte), aunque existe la posibilidad de considerar las notas de las partes aprobadas en primera convocatoria si el profesor lo considerara oportuno.

BIBLIOGRAFÍA

Básica:

- [Kent D. Lee (2014)] Python Programming Fundamentals (Spinger). https://link.springer.com/book/10.1007/978-1-4471-6642-9

- [A. Marzal, I. Gracia, P. García (2014)] Introducción a la programación con Python 3. (Col·lecció Sapientia 93 www.sapientia.uji.esISBN: 978-84-697-1178-1) https://repositori.uji.es/items/992c7ee2-fef4-4061-9958-aefe932fd611
- [A. Downey, J. Elkner, C. Meyers (2002)] Aprenda a Pensar Como un Programador con Python (Green Tea Press). Traducido por M.A. Vilella, A. Arnal, I. Juanes, L. Amurrio, E. Andia, C. Ballardini. https://argentinaenpython.com/quiero-aprender-python/aprenda-a-pensar-como-un-programador-con-python.pdf

Complementaria:

- [Kent D. Lee, Steve Hubbard (2015)]; Data Structures and Algorithms with Python. Undergraduate Topics in Computer Science, Springer Verlag. https://link.springer.com/book/10.1007%2F978-3-319-13072-9
- [Benjamin Baka (2017)] Python Data Structures and Algorithms. Packt Publishing https://ebookcentral.proquest.com/lib/univalencia/detail.action?docID=4868549