



GUÍA DOCENTE

INFORMÁTICA

Grado en Física

Curso 2011/2012



I.- DATOS INICIALES DE IDENTIFICACIÓN

Informática es la única asignatura relativa a la materia del mismo nombre dentro de los estudios de Grado en Física. Esto nos lleva a plantear una asignatura de informática con una visión amplia que nos proporcione los conocimientos básicos y esenciales en informática y en programación, poniendo un especial énfasis en la utilidad y utilización que, para el área de las ciencias básicas puedan tener los conocimientos impartidos.

Nombre de la asignatura:	Informática
Nombre de la materia:	Informática
Carácter:	Formación básica, cuatrimestral
Créditos ECTS:	6
Titulación:	GRADUADO/A EN FÍSICA
Ubicación temporal:	Primer curso. Primer cuatrimestre
Profesor/a responsable:	Ariadna Fuertes Departament d'Informàtica

II.- INTRODUCCIÓN A LA ASIGNATURA

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 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 Informática como herramienta que le permita abordar posteriormente problemas progresivamente más complejos, tanto desde el punto de vista analítico como numérico, y realizar análisis de datos experimentales obtenidos en los diferentes laboratorios de la titulación.

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.

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 y que adquiriera las nociones básicas para poder utilizar las herramientas de computación simbólica y métodos numéricos que necesitarán en otras materias de la titulación.

**III.- VOLUMEN DE TRABAJO**

De acuerdo con la asignación de horarios propuesta por el centro, la previsión del volumen total de trabajo, se distribuirá como indicaremos a continuación. Las sesiones teórico-prácticas tendrán una duración de 1 hora impartidas en dos días a la semana durante un cuatrimestre, dedicando 2/3 del tiempo a los conceptos teóricos y 1/3 del tiempo a la resolución de problemas por parte del estudiante. Las sesiones de laboratorio, en el aula de informática, serán de tres horas y se realizarán durante 10 semanas.

TIPO DE ACTIVIDAD	DESCRIPCIÓN	HORAS
Asistencia a clases	magistrales teórico-prácticas: 2 horas/semana x 15 semanas	30
Asistencia a clases de laboratorio en Aula de Informática	Realización de ejercicios mediante ordenador. Trabajo individual tutorizado por el profesor/a 3 h/ sesión x 10 sesiones	30
Preparación de trabajos Horas de trabajo del estudiante sometidas a evaluación	Resolución de ejercicios y problemas, individualmente y en grupo, propuestos para hacer en casa	15
	Realización de ejercicios mediante ordenador, interpretación, conclusiones y realización de memorias para su comunicación.	25
Estudio-preparación contenidos teórico-prácticos	Teoría: 1,5 h/sem aprox. x 10 semanas Problemas: 3 h/sem ejercicios x 5 semanas	15 15
Estudio para preparación de exámenes:	11 h/examen (<i>aprox</i>) x 1 examen (8 h teoría-práctica + 3 h laboratorio)	11
Realización de exámenes:	4 h/examen (<i>aprox</i>) x 1 examen (2 h teoría-práctica + 2h laboratorio)	4
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.	5
TOTAL VOLUMEN DE TRABAJO		150

IV.- OBJETIVOS GENERALES

- Introducir al alumno en el conocimiento básico de la estructura de un ordenador, tanto a nivel hardware (microprocesador, memoria, etc.) como a nivel software (sistema operativo, programas de aplicación, etc.).
- Introducir al alumno en la metodología de la programación procedural mediante la introducción del concepto de lenguaje de programación y del concepto de algoritmo, así como de los diferentes métodos de resolución de problemas.



- Introducir al alumno en los tipos de datos, variables, constantes, estructuras de control y de datos que necesitaremos para desarrollar programas.
- Ser capaz de interpretar cálculos realizados por el ordenador y de desarrollar programas de software.
- Adquirir destrezas de resolución de problemas y resolverlos a nivel informático.
- Aprender a usar herramientas informáticas en el contexto del cálculo científico.

V.- CONTENIDOS MÍNIMOS

- Estructura interna del ordenador, detallando las partes físicas que lo componen (unidad central de proceso, memoria,...).
- Conocer también las partes lógicas que lo hacen funcionar (sistema operativo, programas, etc.).
- Concepto de algoritmo: Resolución de problemas mediante algoritmos. Análisis del problema. Diseño del algoritmo (diseño descendente o modular y refinamiento por pasos). Programación del algoritmo.
- Introducción de las estructuras básicas de un lenguaje de alto nivel: variables, constantes, estructuras de control, programación modular, recursividad, estructuras de datos, ficheros.
- Aprender a codificar algoritmos sencillos en un lenguaje de programación estructurado.
- Introducción en la programación básica mediante un lenguaje de alto nivel (C, C++,...)

VI.- DESTREZAS A ADQUIRIR

- Tener un conocimiento básico sobre la estructura interna de un ordenador tanto en el ámbito físico (CPU, memoria,...) como en el ámbito lógico (sistema operativo, programas,...), de manera que se pueda comprender el funcionamiento interno del ordenador.
- Tener una visión general de los lenguajes de programación: programación estructurada, orientación a objetos,...
- Aprender a usar la programación como herramienta básica para el trabajo científico y conocer la representación digital de los tipos de variables numéricas.
- Conocer los tipos de datos, variables, constantes, estructuras de control y estructuras de datos que tienen los lenguajes de programación procedurales para desarrollar programas.
- Usar el paradigma de programación procedural para resolver problemas mediante un ordenador.
- Aprender a codificar algoritmos sencillos en un lenguaje de programación estructurado.
- Introducción en la programación básica mediante un lenguaje de alto nivel (C, C++,...) y así conocer los detalles concretos de programación vistos en el módulo teórico (tipos de datos, variable, etc.) y practicar las diferentes estructuras de control, el uso de funciones para realizar un tratamiento modular de los problemas.
- Usar el sistema operativo y conocer sus posibilidades.
- Tener un conocimiento básico de paquetes integrados de software matemático de interés en Física así como en la utilización de programas básicos de tratamiento de los datos experimentales (por ejemplo, Matlab).

**VII.- HABILIDADES SOCIALES O TRANSVERSALES**

- 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.
- Modelización y resolución de problemas: Saber resolver problemas, siendo capaz de identificar los elementos esenciales de una situación y de realizar las aproximaciones requeridas con objeto de reducir los problemas a un nivel manejable.
- 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.
- Ser capaz de proseguir con el estudio de otras materias de la física gracias al bagaje adquirido en el contexto de esta materia.

VIII.- TEMARIO Y PLANIFICACIÓN TEMPORAL

La planificación que se muestra a continuación es, lógicamente, orientativa ya que, dependiendo del ritmo de adquisición de competencias de los alumnos y del grado de madurez de sus conocimientos previos, puede resultar conveniente (o necesario) reajustar el cronograma siguiente.

	TEMA	Horas
1	Introducción 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.	2
2	Algoritmos y programas Concepto de algoritmo. Resolución de problemas mediante algoritmos. Análisis del problema. Diseño del algoritmo: diseño descendente o modular y refinamiento por pasos. Representación de algoritmos: pseudocódigo y organigramas o diagramas de flujo. Tipos de datos simples. Estructuras de control: estructuras secuenciales, estructuras repetitivas, estructuras selectivas. Programación modular. Introducción a la recursividad.	12
3	Aritmética y representación de la información en el ordenador Sistemas de numeración. Conversión entre los diferentes sistemas de numeración (binario, octal, hexadecimal y decimal)	6



	Operaciones aritméticas y lógicas. Representación de la información en el ordenador: datos alfanuméricos, enteros y reales. Aritmética en coma flotante.	
4	Tipos y estructuras de datos Concepto de dato estructurado. Tipos de datos estructurados. Estructuras de datos contiguas: vectores, matrices, cadenas de caracteres y estructuras (o registros). Punteros y estructuras de datos dinámicas. Introducción a las estructuras lineales enlazadas: listas enlazadas.	6
5	Archivos Ficheros: definición y conceptos. Organización física y organización lógica. Operaciones sobre ficheros: creación, apertura y cierre. Lectura y escritura.	4
		30

Sesiones	PRÁCTICAS EN EL LABORATORIO DE INFORMÁTICA	Horas
1	Introducción a la programación. Estructura de un programa.	3
2	Tipos de datos: simple, constantes y variables. Operadores aritméticos, funciones de entrada y salida de datos.	3
3	Operadores relacionales y lógicos. Estructuras de control selectivas	3
4	Estructuras de control repetitivas	3
5	Programación modular. Implementación de funciones. Librerías de funciones	3
6	Vectores y matrices	3
7	Cadenas y registros	3
8	Funciones para el manejo de ficheros	3
9	Programa final	3
10	Introducción al uso paquetes integrados de software matemático de interés en Física (Matlab)	3
	TOTAL	30

IX.- BIBLIOGRAFÍA DE REFERENCIA

a) Bibliografía básica:

[A. Tucker, W. Bradley (1994)]. Fundamentos de informática. (MacGraw Hill).

[W. Savitch (2000)]. Resolución de problemas con C++. El objetivo de la programación (Prentice-Hall)

[L. Joyanes (2000)]. Programación en C++: Algoritmos, estructuras de datos y objetos (MacGraw Hill).

**b) Bibliografía complementaria:**

[L. Joyanes, I. Zahonero (2001)]. Programación en C: Metodología, algoritmos y estructuras de datos (MacGraw Hill)

[H.M. Deitel, P.J. Deitel (1995)]. Como programar en C/C++. (Prentice Hall).

X.- CONOCIMIENTOS PREVIOS

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).

XI.- METODOLOGÍA**(i) Clases teóricas o teórico-prácticas de pizarra**

El desarrollo de la asignatura se estructura en dos sesiones teórico-prácticas a la semana en la que se abordan los aspectos conceptuales y formales de la materia y los aspectos asociados a la resolución de problemas como consecuencia de la aplicación de los conceptos teóricos, de casos particulares así como de técnicas específicas de resolución.

En esas sesiones se introducen los conceptos teóricos, se resuelve algún ejercicio de ejemplo y a continuación se presentan al alumno una serie de ejercicios que se corregirán en sesiones posteriores de clase. En una primera aproximación al tema, el profesor resolverá solo uno o dos ejercicios modelo para que el alumno los tome de referencia para desarrollar las clases prácticas.

(ii) Clases prácticas de pizarra participativas

La resolución de los ejercicios propuestos por el profesor se corregirán una vez los alumnos ya hayan hecho las sesiones prácticas correspondientes al tema a tratar. El objetivo es resolver los ejercicios en la pizarra cuando los alumnos ya se hayan planteado una posible solución del problema y de esa manera participen de una forma más activa en la resolución de los problemas.

La resolución de estos ejercicios la harán los alumnos en la pizarra y se comentará la solución propuesta. El profesor llevará una solución más elaborada que se proyectará después mediante el cañón proyector. Con esta técnica conseguimos que los alumnos puedan ver distintas soluciones del mismo problema.

Indicar que la lista de problemas propuestos se entregará a los alumnos de forma anticipada y se indicará cuáles de ellos se corregirán en las próximas clases. La lista de ejercicios será suficientemente amplia de manera que abarque desde ejercicios y programas sencillos hasta otros más complejos propuestos como opcionales.

(iii) Sesiones de laboratorio en aula de informática

Las sesiones de laboratorio serán de 3 horas cada sesión y en ellas se aprenderá el 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.

Para estas sesiones, los alumnos habrán repasado los principales conceptos que van a ser usados en el desarrollo de la práctica y que están en el enunciado del boletín de prácticas resumidos. Así mismo, el alumno debería haber leído y comprendido los enunciados de los ejercicios propuestos y haber reflexionado sobre la posible solución de los mismos. Durante el tiempo de la práctica, los alumnos resolverán los ejercicios propuestos y preguntarán al profesor aquellos aspectos de los ejercicios que no entiendan.

Por cada materia que el profesor considere adecuada se pedirá al alumno que resuelva un ejercicio propuesto que servirá como pequeño examen y que se realizará durante la primera media hora de clase.

Para que esto sea posible, se deberá disponer de un ordenador por alumno para la realización de las prácticas en el laboratorio de informática.

Las últimas sesiones de prácticas los alumnos resolverán un proyecto de programación que incluirá los conceptos aprendidos durante el curso y en las sesiones de prácticas anteriores. Este "proyecto final" ha de ser suficientemente grande y complejo como para que sea necesaria su descomposición modular. Además en este proyecto han de ponerse en práctica todos los conceptos que han ido aprendiendo de manera individual a lo largo del curso.

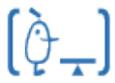
XII.- EVALUACIÓN DEL APRENDIZAJE

A lo largo del curso se irán valorando los trabajos realizados por parte de los estudiantes, presentación de resultados en informes o memorias, cuestiones propuestas y discutidas en el aula, presentación oral, de problemas resueltos por ellos y cualquier otro método que suponga una interacción y realimentación del trabajo del estudiante.

Además, los alumnos tendrán además una nota de prácticas correspondiente a la evaluación de las pequeñas pruebas tipo examen así como de las prácticas realizadas en el laboratorio y del trabajo o proyecto final propuesto.

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.

En primera convocatoria, la nota definitiva de la asignatura resultará de ponderar: 10% de la nota obtenida por los ejercicios resueltos y entregados al profesor; el 40% de la nota obtenida en una prueba escrita final de carácter teórico-práctico; 35% de la nota obtenida en las pruebas cortas realizadas durante las prácticas, y el 15% restante de la nota se corresponderá a las prácticas entregadas y al proyecto final.



En segunda convocatoria sólo se tendrá en cuenta el examen realizado, aunque existe la posibilidad de considerar las notas de las partes aprobadas en primera convocatoria.