Proyecto docente de la asignatura

Asignatura	INFORMÁTICA		
Materia	BÁSICAS, RAMA DE INGENIERÍA Y ARQUITECTURA		
Módulo			
Titulación	GRADO EN MATEMÁTICAS		
Plan	394	**Código**	40004
Periodo de impartición	Primer Cuatrimestre	**Tipo/Carácter**	FB
Nivel/Ciclo	Grado	**Curso**	1º
Créditos ECTS	6 ECTS		
Lengua en que se imparte	ESPAÑOL		
Profesor/es responsable/s	BELARMINO PULIDO JUNQUERA		
Datos de contacto (E-mail, teléfono…)	Email: belar@infor.uva.es Teléfonos: 983 185606		
Departamento	INFORMÁTICA (ATC, CCIA, LSI)		

1. Situación / Sentido de la Asignatura

1.1 Contextualización

La Informática es una disciplina que actualmente está presente en casi todos los ámbitos de nuestra vida, tanto en nuestro tiempo libre como en nuestra vida laboral, denominándose incluso nuestra sociedad como Sociedad de la Información.

El manejo de ordenadores a nivel de usuario se ha conformado en la actualidad como un requisito imprescindible en el desarrollo de cualquier actividad profesional. Este hecho es aún más relevante en el ámbito académico de las Ciencias, donde además de poder manejar un ordenador a nivel de usuario, se suelen solicitar conocimientos de programación científica.

Dentro de la Informática, la programación es un campo esencial, ya que además de permitirnos realizar el tratamiento automático de información mediante programas o aplicaciones, se puede considerar como una herramienta necesaria en casi todos los demás campos de la disciplina. La Programación nos permite resolver problemas que impliquen el tratamiento automático de información. En el campo de las Matemáticas, el conocimiento de una o más técnicas y lenguajes de programación es necesario para resolver múltiples problemas donde no existen soluciones analíticas y es necesario recurrir a soluciones numéricas donde los ordenadores y la programación juegan un papel decisivo.

1.2 Relación con otras materias

El conocimiento de técnicas y lenguajes de programación permitirá solucionar problemas en el ámbito de la Matemática Básica (solución de métodos numéricos), Estadística (Investigación Operativa), etc. Será la base de lo que se conoce como programación científica.

1.3 Prerrequisitos

No hay.

2. Competencias

2.1 Generales

- **Conocer y utilizar recursos informáticos de carácter general y tecnologías de la información y las comunicaciones como medios de comunicación, organización, aprendizaje e investigación.**

- Leer y comprender textos científicos tanto en lengua propia como en otras de relevancia en el ámbito científico, especialmente la inglesa.
- Gestionar de forma óptima, tanto en el trabajo individual como en equipo, el tiempo de trabajo y organizar los recursos disponibles, estableciendo prioridades, caminos alternativos e identificando errores lógicos en la toma de decisiones.
- Tener la capacidad de trabajar en equipo, aportando orden, abstracción y razonamiento lógico; comprobando o refutando razonadamente los argumentos de otras personas y contribuyendo con profesionalidad al buen funcionamiento y organización del grupo.

2.2 Específicas

- **Desarrollar programas que resuelvan problemas matemáticos utilizando para cada caso el entorno computacional adecuado.**

- Utilizar aplicaciones informáticas de análisis estadístico, cálculo numérico y simbólico, visualización gráfica, optimización u otras para experimentar en Matemáticas y resolver problemas.
- Saber abstraer las propiedades estructurales (de objetos matemáticos, de la realidad observada, y de otros ámbitos) distinguiéndolas de aquellas puramente ocasionales y poder comprobarlas con demostraciones o refutarlas con contraejemplos, así como identificar errores en razonamientos incorrectos.
- Capacitar para el aprendizaje autónomo de nuevos conocimientos y técnicas.
- Proponer, analizar, validar e interpretar modelos de situaciones reales sencillas, utilizando las herramientas matemáticas más adecuadas a los fines que se persigan.
- Planificar la resolución de un problema en función de las herramientas de que se disponga y de las restricciones de tiempo y recursos.

3. Objetivos

3.1. Objetivos Generales

1. Manejar una plataforma de sistema operativo a un nivel suficiente que permita desenvolverse con soltura en sus necesidades cotidianas que requieran el uso de computadoras.

2. Entender cómo representar la información mediante tipos de datos básicos. Integrar el comportamiento interactivo o de comunicación con el usuario en el proceso secuencial de un programa. Entender y manejar las estructuras fundamentales de control. Aprender a manejar las estructuras de datos básicas, así como escoger en cada caso la más adecuada y los algoritmos de manejo más eficientes.

3. Comprender el proceso general de la programación. Comprender y analizar el concepto de eficiencia o complejidad en algoritmos básicos. Tener la capacidad de elección de la estructura de datos adecuada para cada tipo de problema.

3.2. Objetivos Específicos

- Comprender el proceso general de la programación:
    - Entender cómo representar la información mediante tipos de datos básicos.
    - Integrar el comportamiento interactivo o de comunicación con el usuario en el proceso secuencial de un programa.
    - Entender y manejar las estructuras fundamentales de control.
    - Aprender a manejar las estructuras de datos básicas, así como escoger en cada caso la más adecuada y los algoritmos de manejo más eficientes.
    - Comprender y analizar el concepto de eficiencia o complejidad en algoritmos básicos.

4. Contenidos y/o bloques temáticos

Sólo existe un bloque temático que se corresponde con la asignatura:

Bloque 1: Fundamentos de Programación

Carga de trabajo en créditos ECTS: 6

a. Contextualización y justificación

Las de la asignatura.

b. Objetivos de aprendizaje

Los de la asignatura.

c. Contenidos

Tema 1. Introducción a la Programación: lenguajes y paradigmas.

Introducción a los lenguajes de programación. Fases de creación de un programa. Traducción: compilación e interpretación.

Tema 2. Nociones de entorno, procesador y acción.

Noción de entorno, acción y procesador. Acciones primitivas. Acciones compuestas. Noción de análisis descendente. Noción de algoritmo.

Tema 3. Tipos y Acciones elementales.

Objetos, constantes y variables. Tipos elementales. Expresiones. Acción de asignación. Composición de acciones. Descripción de un algoritmo. Los arrays.

Presentación del lenguaje C. Estructura de un programa en C. Tipos de datos escalares. Operadores básicos. Definición de array(s) de una dimensión.

Tema 4. Control de flujo del programa I: Esquemas Condicionales.

Esquemas condicionales (sintaxis, semántica y ejemplos): condicional simple, condicional compuesto y esquema condicional generalizado.

Esquemas condicionales en C: if (), if ()+ else, switch ().

Tema 5. Control de flujo del programa II: Esquemas Repetitivos.

Tipos de esquemas (sintaxis, semántica, ejemplos): mientras, repetir, para. Equivalencia entre esquemas.

Esquemas repetitivos en C: while(), do + while(), for ().

Tema 6. Algoritmos con nombre: funciones.

Parametrización de un algoritmo con nombre. Funciones. Los punteros.

Paso de parámetros en C: paso por valor y paso por referencia.

Tema 7. Estructuras de datos: Arrays y Registros.

Tipos estructurados vs tipos vectoriales. Tipos compuestos. El tipo string.

Recorrido secuencial de un array. Búsqueda secuencial. Arrays ordenados. Búsqueda dicotómica. Arrays de más de una dimensión: matrices. Operaciones elementales con matrices.

Definición de arrays en C. Recorrido de arrays en C. Paso de Arrays como argumentos. Arrays y punteros.

Tema 8. Ficheros.

Noción de fichero. Tipos de acceso a ficheros. Asociación de un fichero a un programa. Operaciones primitivas de acceso a ficheros. Operaciones de creación, recorrido, búsqueda y actualización de un fichero secuencial.

Ficheros en C. Operaciones básicas y recorrido de ficheros en C.

Tema 9. Memoria Dinámica.

Memoria estática vs memoria dinámica. Operaciones básicas de gestión de memoria dinámica. Memoria dinámica y punteros.

Tema 10. Introducción a los Tipos de Datos Abstractos

TDAs y su relación con la P.O.O.

d. Métodos docentes

Los generales de la asignatura.

e. Plan de trabajo

Se proporcionará actualizado la primera semana de clase. Se realizarán tres sesiones de teoría/práctica a la semana y se realizarán sesiones de dos horas de laboratorio en, aproximadamente, semanas alternas, hasta cubrir las 15 horas de laboratorio de la asignatura.

Tema	Teoría	Aula	Laboratorio	Seminario	Evaluación
1. 1. Conceptos fundamentales	2	2	0		
1. 2. Introducción programación	2	1	1		
2. Nociones de entorno, procesador y acción	2	2	0		
3. Tipos y acciones elementales	6	4	2		
4. Condicionales	4	2	2		
5. Esquemas repetitivos	6	4	2		
6. Algoritmos con nombre: funciones	6	4	2		
7. Arrays	10,5	7,5	2		1
8. Ficheros	10	5	4		1
9. Memoria Dinámica	3	3	0		
10. Introducción TDA	2	2			
0. Seminario de lógica proposicional	2			2,5	

f. Evaluación

La general de la asignatura.

g. Bibliografía básica

1. *Introducción a la programación. 1, Algorítmica y lenguajes*. Joëlle Biondi, Gilles Clavel; con la colaboración de Silvia Estrems para la corrección de los ejercicios; versión castellana de Nuria Castell Ariño" Barcelona : Masson, 1985 (1ª ed.)
2. *Programación estructurada en C*, José Rafael García-Bermejo Giner. Prentice-Hall. 2008. Sitio web de apoyo: http://maxus.fis.usal.es/FICHAS_C.WEB/MAIN.htm
3. *El lenguaje de programación C*. Brian W. Kernighan, Dennis M. Ritchie; Prentice-Hall, 1991 (2ª ed.)

h. Bibliografía complementaria

1. *Fundamentos de Informática y Programación en C*. Diego R. Llanos Ferraris. Ed. Paraninfo. 2010.
2. *Fundamentos de programación: algoritmos y estructura de datos y objetos*. Luis Joyanes Aguilar. Madrid: MacGraw-Hill, 2003 (3ª ed.)
3. *Fundamentos de programación: libro de problemas*. Luis Joyanes Aguilar, Luis Rodríguez Baena, Matilde Fernández Azuela. Madrid: MacGraw-Hill, D.L. 2003 (2ª ed.)
4. *El libro del C: primer lenguaje.* Claude Delannoy ; traducción y revisión de Amadeu Brugués. París: Eyrolles, 1995

5. Métodos docentes y principios metodológicos

- Clases Teóricas: método expositivo con técnica de la pregunta.
- Se realizarán sesiones de problemas dentro de las clases teóricas para afianzar los conceptos que se están viendo.
- Seminarios.
- Clases prácticas en el laboratorio de ordenadores de la Facultad.

6. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teóricas	20	Estudio y trabajo autónomo individual	70
Clases prácticas	15	Estudio y trabajo autónomo grupal	20
Laboratorios	15		
Prácticas externas, clínicas o de campo			
Seminarios	6		
Otras actividades	4		
Total presencial	**60**	**Total no presencial**	**90**

7. Sistemas y características de la evaluación

La evaluación consistirá en la realización de, al menos, dos pruebas de contenidos teórico/prácticos, con resolución de problemas.

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Solución de Problemas (cuestiones teórico-prácticas)	100%	Consistirá en solucionar pequeños ejercicios de programación en lenguaje algorítmico y/o C.

CRITERIOS DE CALIFICACIÓN

- **Convocatoria ordinaria:**
  - Durante el curso se realizará evaluación continua. La nota final de la convocatoria ordinaria se obtendrá con la suma ponderada de las calificaciones obtenidas en cada parte.
  - Habrá uno o dos ejercicios de evaluación intermedios con preguntas teórico/prácticas. El número se decidirá en función de la disponibilidad de recursos de laboratorio y de la coordinación final con el resto de asignaturas del curso. Si hay un único ejercicio durante el cuatrimestre, éste supondrá el 30% de la nota final y se realizará en el aula al finalizar el tema 5 de la asignatura. Si hubiese dos pruebas, el promedio de sus notas supondrá el 40% de la nota final de la asignatura. La segunda prueba se realizaría tras finalizar el tema 7 y se realizaría en los laboratorios. Ninguno de las pruebas intermedias elimina materia, pues los contenidos de la asignatura son incrementales.
  - Habrá una prueba final de cuestiones teórico-prácticas que supondrá el 70% ó el 60% de la nota final, según haya una o dos pruebas intermedias, respectivamente. Esta prueba se realizará en la fecha oficial de examen de la asignatura y en ella se evaluará toda la materia relacionada con la programación.

- **Convocatoria extraordinaria:**
  - Si no se supera la asignatura durante la convocatoria ordinaria, se realizará un único examen extraordinario en julio donde se evaluará toda la materia de la asignatura y su superación supondrá el 100% de la nota en la convocatoria extraordinaria. Se podrá conservar la nota del examen del bloque I para la convocatoria extraordinaria en caso de haberlo superado y si el estudiante así lo manifiesta.

8. Consideraciones finales

- Se recomienda disponer de un ordenador personal y conexión a Internet complementarias a las que proporcionará el centro.
- Se utilizará un compilador gratuito de C que será proporcionado por los profesores de la asignatura al comienzo de las clases de laboratorio.
- Las fechas definitivas de ejercicios de evaluación serán comunicadas al inicio del cuatrimestre.