

**TITULO: INGENIERO TÉCNICO INDUSTRIAL**

<b>Asignatura:</b> Fundamentos de Informática	<b>Curso/Período:</b> 1º / 1 <sup>er</sup> Semestre
<b>Departamento:</b> Sistemas Informáticos	<b>Carácter/Créditos:</b> Troncal / 6
<b>Subject:</b>	<b>ECTS Credits:</b> Hours/week:

**Descriptores:**

Estructura de los computadores. Programación. Sistemas operativos. (BOE 23/jun/2004)

**Objetivos:**

La asignatura pretende introducir al alumno en el conocimiento de la Programación Estructurada como herramienta orientada a la resolución de problemas complejos. Para ello, el alumno deberá adquirir un método de razonamiento lógico en el planteamiento y resolución de problemas con su posterior aplicación a un lenguaje de programación.

El lenguaje de programación elegido ha sido C, porque permite seguir el paradigma de programación estructurada. Su conocimiento ayudará en otros aspectos de la formación del ingeniero, ya que se puede usar también como herramienta de cálculo o de acceso a bajo nivel de los instrumentos. Además, es un lenguaje de amplia aceptación en la industria.

**Programa:**

**Tema 1.- Algoritmo y Programas (3h teoría, 2h prácticas)**

- \*.- Estructura del ordenador, sistema operativo y programas
- \*.- Diseño de un programa.
- \*.- Programación modular.
- \*.- Programación Estructurada.

**Tema 2.- Introducción al C (3h teoría, 2h prácticas)**

- \*.- Conceptos básicos.
- \*.- El conjunto de caracteres de C.
- \*.- Identificadores y palabras reservadas.
- \*.- Tipos de datos.
- \*.- Constantes.
- \*.- Variables.
- \*.- Declaraciones.
- \*.- Expresiones.
- \*.- Instrucciones.
- \*.- Constantes simbólicas.
- \*.- Funciones básicas de entrada/salida: printf, scanf, getchar, putchar

**Tema 3.- Operadores y expresiones (3h teoría, 2h prácticas)**

- \*.- Operadores aritméticos.
- \*.- Operadores relacionales y lógicos.
- \*.- Operadores de asignación.
- \*.- El operador condicional.

#### **Tema 4.- Instrucciones de control de flujo (4h teoría, 4h prácticas)**

- \*.- Instrucción *if-else*.
- \*.- Instrucción *switch*.
- \*.- Instrucción *for*.
- \*.- Instrucción *while*.
- \*.- Instrucción *do-while*.

#### **Tema 5.- Funciones (4h teoría, 4h prácticas)**

- \*.- Introducción.
- \*.- Declaración y uso de funciones en C.
- \*.- La instrucción *return*.
- \*.- Prototipos de funciones.
- \*.- Paso de argumentos.
- \*.- Ámbito de declaración de variables.
- \*.- Recursividad.

#### **Tema 6.- Arrays y Cadenas de caracteres (4h teoría, 4h prácticas)**

- \*.- Introducción.
- \*.- Arrays unidimensionales.
- \*.- Declaración de arrays.
- \*.- Procesamiento de un array.
- \*.- Arrays multidimensionales.
- \*.- Lectura y escritura de los elementos de un array.
- \*.- Paso de arrays a funciones.
- \*.- Cadenas de caracteres.
- \*.- Funciones importantes de cadenas: *strcpy*, *strlen*, *strcmp*
- \*.- Funciones de entrada/salida: *gets*, *puts*, *fgets*

#### **Tema 7.- Punteros (4h teoría, 4h prácticas)**

- \*.- Conceptos básicos.
- \*.- Declaración de punteros.
- \*.- Punteros y funciones.
- \*.- Operaciones con punteros.
- \*.- Asignación dinámica de memoria.

#### **Tema 8.- Estructuras (3h teoría, 2h prácticas)**

- \*.- Definición de una estructura.
- \*.- Procesamiento de una estructura.
- \*.- Tipos de datos definidos por el usuario (*typedef*).
- \*.- Vectores de estructuras y punteros a estructuras
- \*.- Paso de estructuras a una función.
- \*.- Estructuras autorreferenciadoras.

#### **Tema 9.- Archivos de datos (4h teoría, 4h prácticas)**

- \*.- Apertura y cierre de un archivo.
- \*.- Lectura y escritura de archivos de texto: *fprintf*, *fscanf*
- \*.- Lectura y escritura de archivos binarios: *fread*, *fwrite*
- \*.- Acceso directo a los datos: *fseek*

## **PRACTICAS DE LABORATORIO:**

Durante las clases de Laboratorio se realizarán prácticas referentes a los contenidos teóricos explicados

Las prácticas se realizan durante el tiempo de clase y se entregan al final. En caso necesario el alumno puede terminar las prácticas fuera del horario de clase, pero tendrá que entregarlas en la siguiente clase.

### **Sistema de evaluación:**

La asistencia a clase es obligatoria según las normas de la ETSI (ICAI)

La calificación final de la asignatura, que es semestral, consta de tres partes con los siguientes porcentajes sobre la nota final:

Examen intercuatrimestral de NOVIEMBRE: Teoría (**25%** de la nota final)

Examen final de FEBRERO: Teoría + Práctica (**75%** de la nota final)

En la convocatoria extraordinaria de septiembre la nota del examen (Teoría y Práctica) constituye la nota final

### **Método docente:**

El método de trabajo combina una concepción flexible de la Lección Magistral con Prácticas dedicadas a la formulación y resolución de problemas. Con el objeto de promover el papel activo del alumno se proponen problemas para que sean resueltos de forma individual o en grupos reducidos. La evaluación de estos problemas se aprovecha para realizar un trabajo de orientación académica y seguimiento del aprendizaje de los alumnos.

Las sesiones en el laboratorio completan la formación del alumno, ya que permite afianzar los conocimientos generales previamente transmitidos en las sesiones teóricas y extenderlos a aplicaciones diversas.

### **Bibliografía:**

#### **Bibliografía básica:**

J.D. Muñoz Frías, R. Palacios, "Fundamentos de programación utilizando el lenguaje C",  
Ed. Universidad Pontificia Comillas. Madrid, España. 2006. ISBN: 84-8468-184-1

#### **Bibliografía complementaria:**

B.S. Gottfried, "Programación en C. Serie Schaum 2ª Edición revisada",  
Ed. McGraw-Hill, 2005. ISBN: 84-4819-846-8

J.L. Antonakos, K.C. Mansfield, "Programación estructurada en C"  
Ed. Prentice Hall, 1997. ISBN: 84-8966-023-9

B.W. Kernighan, D.M. Ritchie, "El lenguaje de programación C", segunda edición.  
Ed. Prentice-Hall, 1995. ISBN: 96-8880-205-0