

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
INGENIERIA EN
AERONÁUTICA
PROGRAMA DE ESTUDIOS

Resolución 15/03/07-00 Acta 926/16/02/2015 Anexo 05.1

I. - IDENTIFICACIÓN

- | | | |
|------|----------------------------------|-------------|
| 1. | Asignatura | : Algoritmo |
| 2. | Semestre | : Segundo |
| 3. | Horas semanales | : 5 horas |
| 3.1. | Clases teóricas | : 3 horas |
| 3.2. | Clases prácticas | : 2 horas |
| 4. | Total real de horas disponibles: | 85 horas |
| 4.1. | Clases teóricas | : 51 horas |
| 4.2. | Clases prácticas | : 34 horas |

II. - JUSTIFICACIÓN

Actualmente la programación de computadoras es una tarea no exclusiva de las carreras de Informática, y en especial con el avance tecnológico, su aprendizaje es esencial para cualquier profesional del área que involucre tecnología y más aún en carreras que están fuertemente ligadas a la computación desde la perspectiva del hardware y a la "microprogramación".

Esta asignatura introduce al estudiante a los fundamentos de la programación de computadoras de manera a capacitarlo sobre todo en el análisis y resolución de problemas de carácter general a través de computadoras y adentrarlo en el uso, por medio de prácticas de programación, de algún lenguaje de programación de propósito general.

El curso pone énfasis en la resolución de problemas algorítmicos conjugando conceptos fundamentales asociados a los lenguajes de programación del tipo imperativo.

III. - OBJETIVOS

1. Desarrollar habilidades de análisis y resolución de problemas algorítmicos.
2. Aplicar correctamente técnicas de programación estructurada y modularizada a las soluciones algorítmicas encontradas.
3. Fortalecer conocimientos de los conceptos asociados a los lenguajes de programación y su relación con las soluciones algorítmicas.
4. Hacer uso correcto de las construcciones presentes en los lenguajes de programación teniendo en cuenta recursos computacionales y tiempo de respuesta.
5. Fortalecer la capacidad de expresar soluciones algorítmicas y su materialización en algún lenguaje de programación concreto.
6. Valorar la importancia del aprendizaje y comprensión de la resolución de problemas algorítmicos en el contexto de la carrera de Ingeniería en Electrónica.
7. Reconocer la importancia del código autodocumentado y de estilo de programación en la escritura de los programas.

IV. - PRE - REQUISITO

1. Álgebra

V. - CONTENIDO

5.1. Unidades programáticas

1. Introducción a la resolución de problemas computacionales.
2. Identificadores, tipo de datos simples, operadores y expresiones.
3. Estructura general de un programa.
4. Cadenas.
5. Subprogramas.
6. Tipo de datos estructurados y definidos por el programador
7. Archivos
8. Introducción a la programación con lenguaje C.

5.2. Desarrollo de las unidades programáticas

1. Introducción a la resolución de problemas computacionales.
 - 1.1. Problemas, algoritmos y programas.
 - 1.2. Análisis del problema.
 - 1.3. Características y representación de algoritmos (Pseudocódigo, Diagrama de flujos, Diagramas de
 - 1.4. Conceptos básicos sobre lenguajes de programación.
2. Identificadores, tipo de datos simples, operadores y expresiones
 - 2.1. Identificadores. Uso y formación.
 - 2.2. Constantes literales y nominales del tipo numérica, de cadena y lógicas.
 - 2.3. Tipo de datos simples: numérico, lógico y cadena.
 - 2.4. Variables. Concepto asociado.
 - 2.5. Operación de asignación.
 - 2.6. Funciones internas.
 - 2.7. Operaciones de entrada y de salida.

- 2.8. Expresiones aritméticas, relacionales y lógicas.
- 2.9. Precedencia de operadores.
3. Estructura general de un programa.
 - 3.1. Concepto de programa.
 - 3.1.1. Partes fundamentales de un programa.
 - 3.1.2. Concepto de programación estructurada.
 - 3.2. Algoritmos con estructura lineal o secuencial.
 - 3.3. Algoritmos con estructura selectiva.
 - 3.3.1. Definición y uso de sentencias condicionales simple y escalonada.
 - 3.4. Algoritmos con estructuras iterativas.
 - 3.4.1. Definición y uso de sentencias repetitivas.
 - 3.5. Diseño y estilo de programación:
 - 3.5.1. Nombres de variables y constantes adecuados.
 - 3.5.2. Código mantenible: uso adecuado de identificadores y estructuras.
 - 3.5.3. Documentación interna de programas.
 - 3.5.4. Verificación y depuración de programas.
4. Cadenas.
 - 4.1. Propósito, uso y representación (juego de caracteres).
 - 4.2. Operaciones básicas con cadenas.
 - 4.3. Funciones predefinidas para manejo de cadenas.
5. Subprogramas.
 - 5.1. Filosofía de la programación modular.
 - 5.2. Propósito, sintaxis y funcionamiento.
 - 5.3. Funciones y procedimientos.
 - 5.4. Ámbito de variables: locales y globales.
 - 5.5. Paso de parámetros: por valor y por referencia.
6. Tipo de datos estructurados y definidos por el programador.
 - 6.1. Arreglos o vectores: sintaxis, definición y uso de arreglos unidimensionales y n-dimensionales.
 - 6.2. Registros: sintaxis, definición y uso. Acceso a campos.
 - 6.3. Tipo de datos definidos por el usuario.
 - 6.4. Tipos de datos estructurados como parámetros de subprogramas.
7. Archivos
 - 7.1. Concepto y representación
 - 7.2. Procesamiento de archivos secuenciales
8. Introducción a la programación con lenguaje C
 - 8.1. Historia e importancia del lenguaje C
 - 8.2. Compilación de programas en lenguaje C. Componentes de la plataforma de compilación de un entorno C.
 - 8.3. Construcciones más importantes del lenguaje: estructura de un programa en C, estructuras de control, tipo de datos, funciones, entrada/salida básica, punteros, cadenas, arreglos, registros (struct), enumerados, directivas básicas del preprocesador.
 - 8.4. Traducción y prueba de programas sencillos escritos en C.

VI. - ESTRATEGIAS METODOLÓGICAS

Las clases serán teórico/prácticas con resolución de ejercicios propuestos y discusión en clase. Se entregarán ejercitatorios semanales en carácter de trabajo práctico. Se harán prácticas en Sala de Máquinas usando el Entorno de Programación SL o similar y al final del curso se harán prácticas usando cualquier compilador de acceso gratuito de lenguaje C.

VII. - MEDIOS AUXILIARES

1. Tablero acrílico y pinceles.
2. Computador y proyector.
3. Herramienta de diseño de algoritmos.
4. Lenguaje de programación C.
5. Sala de máquinas para clases prácticas.

VIII. - EVALUACIÓN

El nivel de aprovechamiento se evaluará mediante un mínimo de dos revisiones parciales, ya sea en forma de examen escrito o presentación de trabajos prácticos de investigación sobre un tema relativo al contenido de la materia. El puntaje acumulado en estas revisiones, conforme a la reglamentación de escala, permitirá o no al alumno acceder al examen final, donde será evaluado sobre el total del contenido programático de la materia.

IX. - BIBLIOGRAFÍA

- ☐ El lenguaje de programación C. Kernighan/Ritchie
- ☐ Fundamentos de Programación. Algoritmos y Estructura de Datos. Luis Joyanes Aguilar - 2da. Edición. 1996
- ☐ Lenguaje C. Manual de Referencia. Herbert Schildt.
- ☐ Libros de referencia de Lenguaje C
- ☐ SL. Introducción al lenguaje-Referencia de subrutinas predefinidas-Ejemplos selectos. Segovia Silvero, Juan. Centro Nacional de Computación, Universidad Nacional de Asunción. 1999