

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
INGENIERÍA EN INFORMÁTICA
PLAN 2008
PROGRAMA DE ESTUDIOS
Resolución 08/08/09-00 Acta 751/14/04/2008

I. - IDENTIFICACIÓN

- | | |
|------------------------------------|---------------------------------------|
| 1. Materia | : Algoritmos y Estructuras de Datos I |
| 2. Semestre | : Primer |
| 3. Horas semanales | : 9 horas |
| 3.1. Clases teóricas | : 5 horas |
| 3.2. Clases prácticas | : 4 horas |
| 4. Total real de horas disponibles | : 144 horas |
| 4.1. Clases teóricas | : 80 horas |
| 4.2. Clases prácticas | : 64 horas |

II. - JUSTIFICACIÓN

Los algoritmos constituyen elementos fundamentales para el funcionamiento del computador, configurando la base del software. El ingeniero informático debe estar en condiciones de construir, verificar y evaluar algoritmos aplicando un enfoque metodológico que garantice la obtención de sistemas informáticos completos, eficientes y confiables, a un costo razonable.

III. - OBJETIVOS

1. Proporcionar conceptos y técnicas requeridas para el análisis de problemas y el planteamiento de soluciones a través de algoritmos.
2. Identificar las etapas por las que atraviesa un algoritmo durante su ciclo de vida: análisis, diseño, codificación, prueba y mantenimiento.
3. Presentar las estructuras de datos básicas, ejemplificando su aplicación en diferentes casos prácticos.
4. Apoyar el desarrollo de otras materias del semestre mediante el planteamiento de ejercicios relacionados.
5. Entrenar al estudiante en el uso de un lenguaje estructurado, de uso general y extendido, como herramienta para la conversión de algoritmos en programas.
6. Proporcionar los fundamentos teóricos y prácticos básicos para posteriores estudios en programación.

IV. - PRE - REQUISITO

No tiene.

V. - CONTENIDO

5.1. Unidades programáticas

1. Presentación.
2. Datos, Tipos de datos y operaciones.
3. Introducción a las estructuras de datos.
4. Elementos componentes de un algoritmo.
5. Ciclo de vida de un algoritmo.
6. Técnicas de solución de problemas mediante algoritmos.
7. Construcción de programas.

5.2. Desarrollo de las unidades programáticas.

1. Presentación.
 - 1.1. Identificación de algoritmos en la vida cotidiana.
 - 1.2. Sistemas de procesamiento de la Información: Hardware y Software.
 - 1.3. Concepto de algoritmo y programa.
 - 1.3.1. Pseudo-código y Lenguajes de programación.
 - 1.3.2. Ejemplos descriptivos.
2. Datos, Tipos de datos y operaciones.
 - 2.1. Datos de tipo numérico, carácter y lógico.
 - 2.2. Operaciones definidas para cada tipo. Representación y operadores.
 - 2.3. Reglas de precedencia.
 - 2.4. Funciones internas.
 - 2.5. Ejercicios de aplicación.
3. Introducción a las estructuras de datos.
 - 3.1. Arreglos unidimensionales (vectores).
 - 3.2. Arreglos multidimensionales (matrices).
4. Elementos componentes de un algoritmo.
 - 4.1. Acciones Básicas.
 - 4.1.1. Asignación.
 - 4.1.2. Entrada y salida.

- 4.1.3. Declaración de constantes, variables y arreglos.
- 4.2. Estructuras básicas.
 - 4.2.1. Alternativa.
 - 4.2.1.1. Simple.
 - 4.2.1.2. Doble.
 - 4.2.1.3. Múltiple.
 - 4.2.1.4. Anidamiento.
 - 4.2.2. Estructuras de repetición.
 - 4.2.2.1. Mientras.
 - 4.2.2.2. Desde.
 - 4.2.2.3. Repetir.
 - 4.2.2.4. Anidamiento.
 - 4.2.3. Estructura de un algoritmo.
 - 4.2.4. Partes componentes.
 - 4.2.5. Formalidades en la redacción.
- 4.3. Ejercicios de aplicación.
 - 4.3.1. Construcción de algoritmos ejemplificando los conceptos presentados.
- 5. Ciclo de vida de un algoritmo.
 - 6.1. Análisis del problema.
 - 6.2. Diseño de soluciones.
 - 6.3. Codificación.
 - 6.4. Prueba.
 - 6.5. Mantenimiento.
- 6. Técnicas de solución de problemas mediante algoritmos.
 - 6.6. Enfoque descendente.
 - 6.7. Descomposición en partes.
 - 6.8. Calidad en la redacción del algoritmo.
 - 6.9. Ejercicios de aplicación.
 - 6.1.1. Solución de problemas de mediana complejidad aplicando las técnicas estudiadas.
 - 6.1.2. Prueba de escritorio.
- 7. Construcción de programas.
 - 6.10. Presentación del lenguaje.
 - 6.11. Entorno de operación.
 - 6.12. Traducción de algoritmos en programas.
 - 6.13. Redacción y edición de programas en el computador.
 - 6.14. Técnicas de prueba y localización de errores.

VI. - ESTRATEGIAS METODOLÓGICAS

1. Presentación de conceptos y solución de problemas en presencia del profesor, utilizando pizarra y proyección de transparencias.
2. Formación de grupos y trabajo en equipo para resolver problemas durante las clases.
3. Propuesta de temas de investigación bibliográfica y ejercicios para desarrollo extra-clase.
4. Trabajos prácticos.

VII. - MEDIOS AUXILIARES

1. Pizarra.
2. Retroproyector y Transparencias.
3. Sala de máquinas para clases prácticas de programación.
4. Lenguaje de computación C ++.
5. Bibliografía de apoyo.

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. El puntaje acumulado en estas revisiones, conforme a la reglamentación de escalas, permitirá o no al alumno acceder al examen final, donde será evaluado sobre el total del contenido programático de la materia.

IX. - BIBLIOGRAFÍA

MATERIALES BIBLIOGRÁFICOS DISPONIBLES EN LA BIBLIOTECA DE LA FACULTAD POLITÉCNICA

- Deitel, H. M. & Deitel, P.J. (2004). *Cómo programar en C/C++ y Java*. (4° Ed.). México: Pearson Educación.
- Joyanes Aguilar, L. & Zahonero Martínez, I. (2010). *Programación en C, C++, java y UML*. México: McGraw-Hill.
- Joyanes Aguilar, L. & Zahonero Martínez, I. (2004). *Algoritmos y estructuras de datos: una perspectiva en c*. Madrid: McGraw-Hill.
- Joyanes Aguilar, L. (2001). *Programación en C: metodología, algoritmos y estructura de datos*. Madrid: McGraw-Hill.
- Joyanes Aguilar, L. (2008). *Fundamentos de programación: algoritmos, estructura de datos y objetos*. (4° Ed.). Madrid: McGraw-Hill.
- Laudén, K. C. (2004). *Lenguajes de programación: principios y prácticas*. (2 Ed.). México: Thomson.
- Pimiento Cárdenas, W. M. (2009). *Fundamentos de lógica para programación de computadores*. Bogotá: Universidad Piloto de Colombia.
- Tucker, A. & Noonan, R. (2003). *Lenguajes de programación: principios y paradigmas*. Madrid: McGraw-Hill.

RECURSOS DISPONIBLES A TRAVÉS DE CICC0

- ❑ Cachero Castro, C., Ponce de León Amador, P. J., & Saquete Boró, E. (2006). *Introducción a la programación orientada a objetos*. San Vicente del Raspeig: Digitalia. Recuperado de: <http://ebookcentral.proquest.com>
- ❑ Corbí Bellot, A. (2001). *Fundamentos de programación*. [Alicante]: Digitalia. Recuperado de: <http://ebookcentral.proquest.com>
- ❑ Llopis Pascual, F., Pérez López, E., & Ortuño Ortín, F. (2000). *Introducción a la programación: algoritmos y C/C++*. [Alicante]: Digitalia. Recuperado de: <http://ebookcentral.proquest.com>
- ❑ Llorens Largo, F. (2002). *Programación: formalización, análisis y reutilización de algoritmos matemáticos*. [Alicante]: Digitalia. Recuperado de: <http://ebookcentral.proquest.com>