

**UNIVERSIDAD NACIONAL DE ASUNCIÓN**  
**FACULTAD POLITÉCNICA**  
**INGENIERIA EN INFORMÁTICA**  
**PLAN 2008**  
**PROGRAMA DE ESTUDIO**

Resolución Nº 16/14/32-00 Acta Nº 968/11/07/2016

## **I. IDENTIFICACIÓN**

1. Asignatura	: Algoritmo y Estructura de Datos III
2. Semestre	: Tercero
3. Horas semanales	: 7 horas
3.1. Clases teóricas	: 5 horas
3.2. Clases prácticas	: 2 horas
4. Total real de horas disponibles	: 112 horas
4.1. Clases teóricas	: 80 horas
4.2. Clases prácticas	: 32 horas

## **II. JUSTIFICACIÓN**

Esta asignatura es una continuación de las anteriores sobre estructuras de datos y algoritmos, temas centrales de las ciencias de la computación y de cualquier carrera de informática.

Los algoritmos son fundamentales para las ciencias de la computación y para la ingeniería de software. La solución eficiente de un problema depende fundamentalmente del algoritmo elegido. Por lo tanto, es importante desarrollar la habilidad de entender un problema y saber identificar la técnica algorítmica adecuada para su solución teniendo en cuenta el costo computacional involucrado.

El curso presenta y analiza las estructuras de datos y los algoritmos fundamentales desarrollados en las últimas décadas, haciendo énfasis en problemas relacionados a búsqueda y ordenación - interna y externa -, búsqueda de texto, algoritmos de grafos y técnicas de diseño de algoritmos. Para cada algoritmo estudiado se realiza un análisis del costo computacional asociado y su correctitud de manera a que cada estudiante pueda desarrollar una intuición algorítmica para la resolución de problemas.

## **III. OBJETIVOS GENERALES**

1. Evaluar críticamente la aplicabilidad de un algoritmo y estructura de dato en el contexto de problemas concretos.
2. Comprender e implementar los algoritmos fundamentales en ciencias de la computación.
3. Evaluar el costo/beneficio que existen entre varios algoritmos que ofrecen la misma funcionalidad.
4. Aplicar diferentes técnicas de diseño de algoritmos para resolver problemas computacionales.
5. Emplear correctamente las técnicas de análisis de algoritmos como herramienta fundamental en la toma de decisiones de diseño e implementación de soluciones algorítmicas a problemas computacionales.

## **IV. OBJETIVOS ESPECÍFICOS**

### **1. Conocimientos**

- 1.1. Explicar el significado y evaluación de los casos de comportamiento de los algoritmos (mejor, peor, promedio).
- 1.2. Identificar factores que influyen en la selección de un algoritmo.
- 1.3. Describir y comprender los algoritmos fundamentales relacionados a ordenación, búsqueda y grafos.
- 1.4. Describir y diferenciar las diversas estrategias para el diseño de algoritmos de acuerdo a los requerimientos y restricciones.
- 1.5. Comprender las implicancias de los problemas difíciles y las técnicas de solución posibles.

### **2. Habilidades**

- 2.1. Aplicar herramientas teóricas y empíricas para validar la hipótesis de uso temporal y espacial de los algoritmos.
- 2.2. Analizar y describir asintóticamente el comportamiento de los algoritmos de forma a poder compararlos.
- 2.3. Implementar correctamente los algoritmos y estructura de datos teniendo en cuenta las restricciones de espacio y tiempo.
- 2.4. Seleccionar la estructura de datos adecuada de acuerdo al problema específico a resolver.
- 2.5. Identificar problemas difíciles y establecer las posibles formas de enfrentarlos.

### **3. Competencias**

- 3.1. Aplicar los conocimientos teóricos en la implementación de soluciones a problemas reales.
- 3.2. Capacidad de plantear y resolver problemas específicos con requerimientos y limitaciones establecidas.
- 3.3. Capacidad de trabajo en equipo.
- 3.4. Capacidad de abstracción, análisis y síntesis.
- 3.5. Habilidad en la búsqueda y comprensión de información utilizando las fuentes pertinentes.
- 3.6. Capacidad de comunicación oral y escrita.

## **V. PRE - REQUISITO**

1. Algoritmos y Estructuras de Datos II.
2. Lenguajes de Programación I

## VI. CONTENIDO

### 6.1. Unidades programáticas

1. Introducción.
2. Análisis de algoritmos.
3. Algoritmos de ordenación y búsqueda.
4. Ordenación externa.
5. Algoritmos de grafos y sus aplicaciones.
6. Técnicas de diseño de algoritmos.
7. Límites a la computación.

### 6.2. Desarrollo de las unidades programáticas

1. Introducción.
  - 1.1. Revisión de conceptos sobre estructuras de datos, tipos abstractos de datos, algoritmos, problema computacional, lenguajes, programas.
  - 1.2. Definición/Implementación de tipos abstractos de datos.
  - 1.3. Ejercicios de programación utilizando tipos abstracto de datos sencillos (listas, pilas, colas)
2. Análisis de algoritmos.
  - 2.1. Concepto de análisis de algoritmo.
  - 2.2. Análisis de casos: mejor, peor y promedio
  - 2.3. Análisis asintótico: cota inferior, superior y exacta. Notaciones asintóticas y propiedades.
  - 2.4. Cálculo de tiempo de ejecución de algoritmos.
  - 2.5. Análisis de algoritmos recursivos. Resolución de ecuación de recurrencia. Árbol de recurrencia y Teorema maestro.
  - 2.6. Evaluación de compromiso entre análisis temporal y espacial en algoritmos.
3. Algoritmos de ordenación y búsqueda.
  - 3.1. Árboles. Conceptos.
  - 3.2. Árboles binarios de búsqueda.
  - 3.3. Árboles binarios de búsqueda balanceados.
  - 3.4. Tablas de dispersión. Función de dispersión. Técnicas de manejo de colisiones.
  - 3.5. Algoritmos de ordenación interna.
    - 3.5.1. Basados en comparación
      - 3.5.1.1. Algoritmos: Inserción, Mezcla, Shell, Rápida (QuickSort) y por Montículo (HeapSort).
      - 3.5.1.2. Límite de los algoritmos basados en comparación.
    - 3.5.2. No basados en comparación
      - 3.5.2.1. Algoritmos: Conteo (CountingSort), Cubetas (BinSort) y RadixSort.
  - 3.6. Búsqueda de patrones en cadenas
    - 3.6.1. Definición del problema y aplicaciones.
    - 3.6.2. Algoritmos de búsqueda: fuerza bruta, KMP, Boyer-Moore y Rabin-Karp.
4. Ordenación externa.
  - 4.1. Algoritmos de ordenación por mezcla.
  - 4.2. Estrategias de ordenación externa multi-carreras.
  - 4.3. Árboles B y variantes.
5. Algoritmos de grafos y sus aplicaciones.
  - 5.1. Representación de grafos.
  - 5.2. Algoritmos en grafos:
    - 5.2.1. Búsqueda en profundidad
    - 5.2.2. Búsqueda en amplitud
    - 5.2.3. Orden topológico
    - 5.2.4. Camino mínimo: algoritmos de Dijkstra, Bellman-Ford y Floyd-Marshall.
    - 5.2.5. Árbol de Recubrimiento Mínimo: algoritmos de Prim y Kruskal.
6. Técnicas de diseño de algoritmos.
  - 6.1. Estrategia "divide y vencerás".
  - 6.2. Algoritmos voraces.
  - 6.3. Algoritmos con retroceso.
  - 6.4. Programación dinámica
  - 6.5. Algoritmos y estructura de datos aleatorios.
7. Límites a la computación.
  - 7.1. Problemas NP y NP-completos.
  - 7.2. Formas posibles de enfrentar problemas difíciles.

## VII. ESTRATEGIAS METODOLÓGICAS

1. Exposición a cargo del profesor de los aspectos principales de la unidad.
2. Resolución de ejercicios teóricos en clase y exposición oral de los resultados alcanzados.
3. Prácticas en laboratorio con guía personalizada.
4. Tareas semanales sobre capítulos desarrollados en la semana. Incluye: revisión bibliográfica, resolución de ejercicios teóricos y prácticos que incluyen usualmente tareas de programación.
5. Exposición y discusión de los temas en clase y utilizando el portal de Educación a Distancia "EDUCA".
6. Trabajo práctico final que incluye: presentación de reporte técnico y defensa oral.

## VIII. MEDIOS AUXILIARES

1. Pizarra y pinceles.
2. Proyector multimedia y laptop.
3. Ejercitatorios de clase proveídos por el profesor.
4. Salas de laboratorio con:
  - 3.1 Computadoras de escritorio o laptops con Windows o Unix
  - 3.2 Herramientas de desarrollo de software, compilador de lenguaje Java o algún otro lenguaje con soporte de orientación a objetos.
  - 3.3. Acceso al Portal de Educación a Distancia ("EDUCA") de la FPUNA y a Internet.

## IX. EVALUACIÓN

1. Los exámenes parciales requeridos por los reglamentos de la Facultad.
2. Presentación de tareas semanales.
3. Un trabajo práctico final.
4. La calificación final será establecida de acuerdo a la escala vigente en la Facultad.

## X. BIBLIOGRAFÍA

- Cormen T., Leiserson C., Rivest R. & Stein C. (2009). *Introduction to algorithm*. (3° Ed.). Cambridge: The MIT Press.
- Dasguta S., Papadimitriou H & Vazirani U.V. (2006). *Algorithms*. (s.l.): McGraw Hill. Recuperado de: <http://algorithmics.lsi.upc.edu>
- Kleinberg J. & Tardos E. (2006). *Algorithm Design*. Boston: Pearson Education.
- Levitin. A. (2007). *The introduction to the design and analysis of algorithms*. (2° Ed.). Boston: Pearson Education. Recuperado de: [http://www.vgloop.com/\\_files/1394454921-126688.pdf](http://www.vgloop.com/_files/1394454921-126688.pdf)
- Sedgewick R. & Wayne K. (2011). *Algorithms*. (4° Ed.). Westford Massachusetts: Addison-Wesley.
- Shafer C. A. (2013). *Data Structures and Algorithm Analysis*. Blacksburg: Department of Computer Science Virginia Tech. Recuperado de: <http://people.cs.vt.edu/~shaffer/Book/JAVA3elatest.pdf>
- Skiena S. (2008). *The Algorithm Design Manual*. (2° Ed.). New York: Springer. Recuperado de: [http://mimoza.marmara.edu.tr/~msakalli/cse706\\_12/SkiennaTheAlgorithmDesignManual.pdf](http://mimoza.marmara.edu.tr/~msakalli/cse706_12/SkiennaTheAlgorithmDesignManual.pdf)
- Skiena, S. & Revilla, M. (2003). *Programming Challenges: The programming Contest Training Manual*. New York: Springer-Verlag. Recuperado de: <https://pdfs.semanticscholar.org>
- Weiss, M. A. (2012). *Data Structures and Algorithm Analysis in Java*. (3° Ed.). (s.l.): Pearson Education.
- Weiss, Mark A. (2000). *Estructuras de datos en Java*. Madrid: Iberoamericana.

### 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. (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 CICCO

- 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>