



Defensa y Cacheuta S/N - Villa el Libertador - Córdoba

Programación III

Horas reloj Anuales	128 hs
Horas Cátedra Anuales	192 hs C
Horas Cátedra Semanales	6 hs C

Lugar y Fecha

(Fundamentación)

2. OBJETIVOS

Se pretende que el alumno logre las siguientes cualidades:

- Capacidad para comprender y describir el problema que resuelve un algoritmo (el “qué”) y diferenciarlo de la manera en que lo resuelve (el “cómo”).
- Capacidad para analizar algoritmos y compararlos según su eficiencia en tiempo y en espacio.
- Capacidad y hábito de identificar abstracciones relevantes al abordar un problema computacional.
- Familiaridad con técnicas de diseño de algoritmos de uso frecuente.
- Familiaridad con la programación de algoritmos y estructuras de datos.

3. APRENDIZAJES Y CONTENIDOS

Unidad Didáctica N° 1 – Análisis de algoritmos.

Definición de algoritmo, análisis de algoritmos.

Diferencia entre mejor, esperado y peor caso de comportamiento de un algoritmo.

Ordenación por inserción, análisis, mejor caso, peor caso y caso medio.

Notación O, Omega y Theta. Propiedades elementales. Regla del límite. Jerarquía. Propiedades adicionales.

Formas empíricas de medir performance.

Relación espacio-tiempo en los algoritmos.

Unidad Didáctica N° 2 – Tipos abstractos de datos

Árboles binarios. Formas de implementación. Árbol binario de búsqueda.

Cola de prioridades. Heap.

Grafos y representación de grafos mediante listas de adyacencia y matriz de adyacencia.

Recorridos de grafos: Búsqueda en profundidad (DFS) y anchura (BFS). Estructuras de datos asociadas y análisis de eficiencia.

Unidad Didáctica N° 3 – Técnicas de diseño de algoritmos

Fuerza Bruta. Ejemplos: Bubble Sort, búsqueda secuencial. Problemas de eficiencia.

Backtracking: Ejemplos: N-Reinas.

Dividir y conquistar: Ejemplos: Quick Sort, Búsqueda binaria. Comparaciones de orden de

eficiencia.

Unidad Didáctica N° 4 – Técnicas Greedy

Problema del árbol de expansión minimal. Algoritmos Prim y Kruskal. Análisis de eficiencia.

Problema de ruta más corta: Algoritmo Dijkstra. Usos.

Unidad Didáctica N° 5 – Limitaciones de la computabilidad

Computabilidad. Problema de la parada. Enumerabilidad.

4. METODOLOGÍA

Se irán desarrollando los contenidos teóricos en clase, complementando con guías de ejercicios para ir afianzando los contenidos.

Se seleccionarán algoritmos representativos para que implementen en un lenguaje y analicen los problemas de implementación específicos.

5. INSTANCIAS Y CRITERIOS DE EVALUACIÓN

Se realizarán instancias de exámenes parciales escritos abordando los contenidos desarrollados en clase.

Se complementarán los exámenes parciales con trabajos prácticos consistentes en la implementación de algoritmos en un lenguaje de bajo nivel.

CONDICIÓN DE REGULARIDAD:

- Asistir al 75% de las clases

- Aprobar todas las evaluaciones parciales o sus respectivos recuperatorios con un mínimo de 4 puntos equivalente al (50%).

CONDICIÓN DE PROMOCIÓN:

- Asistir al 75% de las clases
- Aprobar todas las evaluaciones parciales con un mínimo de 6 puntos y promedio de 7 puntos.
- Aprobar el examen integrador final con un mínimo de 7 puntos.-

6. BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- D. Fridlender - Notas de Algoritmos y Estructuras de Datos II
- A. Levitin - Introduction to The Design & Analysis of Algorithms

BIBLIOGRAFÍA COMPLEMENTARIA

- J. Blanco, S. Smith, D. Barsotti - Cálculo de Programas