

Programa Analítico

1. OBJETIVOS:

- Aprender las diferentes estructuras de datos que se usan para almacenar información.
- Aprender a implementar las estructuras de datos utilizando memoria estática y memoria dinámica.
- Saber elegir adecuadamente las estructuras a utilizar según sea el problema a resolver.
- Aplicar correctamente cada implementación en aplicaciones concretas.
- Desarrollar habilidades para la creación de software de alta calidad.

2. CONTENIDOS:

UNIDAD 1 : Análisis y diseño de Algoritmos.

- 1.1. Eficiencia de algoritmos. Tiempo de ejecución y uso de memoria.
- 1.2. Concepto de complejidad. Algoritmos de tiempo constante, logarítmico, lineal, cuadrático, cúbico y exponencial. Análisis asintótico. Cotas en el caso peor y en el caso promedio. Notación $O()$.
- 1.3. Cálculo de la complejidad temporal y espacial en algoritmos iterativos y en algoritmos recursivos. Balance entre tiempo y espacio en los algoritmos.

UNIDAD 2 : Tipo Abstracto de Dato (TAD).

- 2.1. Definición. Objetivos. Representación gráfica y/o matemática de un TAD. Especificación.
- 2.2. Operaciones primitivas: constructoras, analizadoras, modificadoras. Condiciones previas y posteriores.
- 2.3. Uso de TAD para la resolución de problemas.

UNIDAD 3: Pilas y Colas.

- 3.1. Pila: definición y ejemplos. Especificación. Operaciones primitivas. Aplicaciones: transformación de notación infija a postfija, evaluación de expresiones aritméticas. Implementación del TAD Pila usando memoria contigua y memoria no contigua. Análisis y comparación de las diferentes implementaciones.
- 3.2. Cola: definición y ejemplos. Especificación. Operaciones primitivas. Aplicaciones: cola de impresión, simulación. Distintas implementaciones del TAD Cola: arreglos, arreglo circular y memoria dinámica. Utilización de un nodo cabecera para mejorar la eficiencia de las operaciones. Bicolos y colas con prioridad.

UNIDAD 4: Listas Lineales.

- 4.1. Definición, ejemplos y aplicaciones. Propiedades. Especificación. Operaciones básicas. Aplicaciones.
- 4.2. Distintas representaciones en memoria: arreglos, encadenamiento simple, encadenamiento simple con nodo cabecera, encadenamiento simple con nodo centinela, encadenamiento doble, cursores. Estudio de la complejidad de las operaciones básicas en las distintas representaciones.
- 4.3. Lista circular y lista ordenada.

UNIDAD 5: Árboles.

- 5.1. Árboles binarios. Conceptos básicos: nodos, aristas, raíz, hojas, profundidad, altura. Propiedades.
- 5.2. Especificación. Operaciones primitivas. Recorridos. Aplicaciones: árboles de sintaxis y árboles de Huffman.
- 5.3. Implementaciones estática y dinámica del TAD Arbol Binario. Análisis de la complejidad espacial y temporal de ambas implementaciones.
- 5.4. Árboles Binarios de Búsqueda. Definición. Aplicaciones. Operaciones básicas. Implementación de las operaciones inserción, eliminación y búsqueda en un Arbol Binario de Búsqueda.
- 5.5. Árboles Binarios de búsqueda balanceados: árboles de Adelson, Velskii y Landis; rotaciones.
- 5.6. Árboles M-arios. Árboles B y B++. Creación de archivos índices.

UNIDAD 6: Tablas de dispersión.

- 6.1. Definición y conceptos básicos. Aplicaciones. Funciones de dispersión. Colisiones. Estrategias de resolución de colisiones. Densidad de empaquetamiento.
- 6.2. Dispersión abierta y cerrada. Exploración lineal y cuadrática. Dispersión doble.
- 6.3. Desbordamiento de la tabla. Expansión de la tabla. Dispersión extensible.

UNIDAD 7: Algoritmia de grafos.

- 7.1. Grafos Dirigidos y No Dirigidos. Definición y ejemplos. Conceptos básicos. Especificación.
- 7.2. Caminos en un grafo. Recorrido de grafos: DFS y BFS. Circuitos de Euler y de Hamilton.
- 7.3. Grafos ponderados. Aplicaciones. Obtención del camino más corto. Algoritmos de Dijkstra y de Floyd.
- 7.4. Clausura transitiva. Algoritmo de Warshall.
- 7.5. Árbol abarcador de costo mínimo. Algoritmos de Prim y de Kruskal.

3. BIBLIOGRAFIA

3.1 BASICA

1. Estructuras de Datos y Análisis de Algoritmos en Java, A. M., Addison-Wesley - 2012.

bbffggvddrrgg

3.2 Adicional

1. Algoritmos en Java. Sedgewick. Pearson. 2003.
2. Estructuras de datos y algoritmos. Weiss, A. M. Addison-Wesley Iberoamericana-2000
3. Data structures, algorithms, and performance. Wood. Addison-Wesley. 1993.
4. Estructuras de datos con C++, Objetos, abstracciones y diseño. Koffman, Wolfgang. Mc. Graw Hill. 2006.
5. Estructuras de datos, algoritmos y programación orientada a objetos. Heileman Gregory. Mc. Graw Hill. 1998
6. Diseño y manejo de estructuras de datos en C. Villalobos, J. Mc. Graw Hill. 1996.
7. Estructuras de datos en C. Tenenbaum, Langsam, Augenstein. Prentice-Hall Hispanoamerica. 1993.
8. Estructuras de datos y algoritmos. Aho, Hopcroft, Ullman. Addison-Wesley. 1988.
9. The art of computer programming. Knuth. Addison Wesley Longman, Vol.1 y Vol.3. 1998.

4.METODOLOGIA DE LA ENSEÑANZA

Las clases combinarán los desarrollos teóricos con resolución de problemas y ejercitaciones prácticas desarrolladas en el laboratorio de computación. Durante el presente curso se utilizarán un lenguaje de lato nivel para el desarrollo de las aplicaciones.

En las clases teóricas se privilegiarán métodos pedagógicos activos que aseguren que el estudiante no es un mero receptor de conocimientos, sino que el aula es un laboratorio donde docentes y alumnos construyen en conjunto los nuevos conocimientos.

En las clases en laboratorio los alumnos deberán implementar los algoritmos desarrollados en las clases teóricas utilizando software orientado a la resolución de problemas relacionados con las estructuras de datos.

Los alumnos disponen desde principio de año de una guía de ejercicios, la que puede ser consultada del Portal de la Universidad, así también de apuntes y/o links que son cuidadosamente revisados los profesores que integran la cátedra. Las comunicaciones internas se realizarán por e-mail.

5. CRITERIOS DE EVALUACION

La evaluación se realiza a través de:

- ❖ Los exámenes parciales. Cada examen consta de dos partes:
 - Análisis y resolución de problemas, cuestionarios, seguimiento de algoritmos, etc.
 - Implementación de algoritmos en un lenguaje de alto nivel adecuado a la problemática tratada en este curso.
- ❖ El desarrollo de los trabajos prácticos.
- ❖ Las presentaciones de temas específicos.
- ❖ Evaluaciones que eventualmente se tomarán al finalizar cada unidad.
- ❖ El desempeño del alumno en las clases el cual se medirá por la asistencia y participación en las mismas.

5.1 Aprobación del curso práctico de la materia

Para poder aprobar la cursada de la materia se deberán cumplir las siguientes condiciones:

- ❖ Haber aprobado el parcial reglamentario o su recuperación.
- ❖ Cumplir con la condición de asistencia.
- ❖ Haber aprobado los trabajos prácticos.

5.2 Aprobación de la materia

Para la aprobación de la materia el alumno deberá rendir un examen final. En la nota final se tendrá en cuenta el concepto que se tenga del alumno.