



Estructuras de Datos

Datos de la asignatura:

Nombre de la asignatura: Estructuras de Datos

Carreras: *Tecnicatura en Informática*

Campo: *Ciencias Básicas*

Carga Horaria:

- Carga horaria total: 128 horas
- Carga horaria semanal: 8 horas por semana

Objetivos:

Que el estudiante:

- Comprenda la noción de dato y de estructuras de datos, y su importancia e interrelación estrecha con la estructura algorítmica de un programa.
- Entienda la diferencia entre acceso aleatorio y acceso secuencial.
- Conozca la idea de interface de una estructura de datos, y sea capaz de utilizarla productivamente para la solución de problemas.
- Conozca la interfaz de distintas estructuras de datos básicas (pilas, colas, listas, árboles, hashing, etc.) y las utilice adecuadamente
- Comprenda y utilice la noción de estructura contenedora, y la capacidad de realizar combinaciones complejas utilizándolas (i.e. pila de lista de registros, etc.)
- Se familiarice con las nociones de ámbito y de pasaje de parámetros por valor o referencia.
- Maneje alguno de los principios básicos de diseño de la interfaz de una estructura de datos (separación en constructores e inspectores de una interfaz, ecuaciones entre combinaciones de constructores, etc.), y pueda reconocerlos en situaciones prácticas junto con su utilidad.
- Comprenda el concepto de asignación dinámica de memoria, y pueda hacer programas que hagan un manejo dinámico explícito de memoria en forma adecuada.
- Entienda la noción de implementación de una estructura de datos, y de su eficiencia, y sea capaz de implementar las interfaces vistas anteriormente con distintas alternativas variadas en eficiencia.



Instituto de Tecnología e Ingeniería

- Se familiarice con las tareas de compilar y vincular programas para lograr un ejecutable.
- Pueda resolver problemas mediante programas recursivos, y entienda la diferencia entre una resolución recursiva y otra iterativa.

Contenidos mínimos:

- Recursión sobre listas y árboles. Programas recursivos. Tipos algebraicos: maybe, either, enumerativos, listas, árboles binarios, árboles generales.
- Estructuras contenedoras: pilas, colas, diccionarios, heaps, árboles balanceados, contenedores basados en representaciones numéricas.
- Nociones de representación e invariante de representación y su utilidad en el diseño e implementación de estructuras de datos.
- Uso imperativo de estructuras de datos. Iteración en listas y árboles. Modelo de memoria imperativo: stack/heap, asignación de memoria. Punteros. Variables por referencia.
- Listas encadenadas y sus variantes. árboles implementados con punteros. Binary heaps implementadas con arrays.
- Hashing. Análisis de eficiencia e implementación. Algoritmos de ordenamiento. Clasificación e implementación. Nociones básicas de algoritmos sobre grafos.

**Programa analítico:****UNIDAD 1: INTRODUCCIÓN AL LENGUAJE JAVA.****Objetivos Específicos**

- Incorporar conceptos básicos necesarios para escribir, compilar y ejecutar un programa en este lenguaje de programación.
- Utilizar reglas sintácticas. Tipos de datos primitivos y referenciales.
- Definir funciones y variables. Operadores. Estructuras de control.
- Proporcionar ejemplos y realizar ejercicios.

UNIDAD 2: TIPOS DE DATOS ABSTRACTOS.**Objetivos Específicos**

- Incorporar concepto de abstracción de Datos y Encapsulamiento de Datos.
- Incorporar concepto de Tipo Abstracto de Dato (TDA).
- Diferenciar entre tipo de dato y tipo abstracto de dato.
- Realizar requerimientos y diseño de un Tipo Abstracto de Dato (TDA).
- Trabajar con tipos de datos abstractos (TDA).

UNIDAD 3: FUNCIONES RECURSIVAS, ARRAYS UNIDIMENSIONALES Y BIDIMENSIONALES.**Objetivos Específicos**

- Incorporar principios de la recursión. Funcionamiento de la recursión. Trabajar en ejercicios con funciones recursivas.
- Trabajar con Arrays unidimensionales y Bidimensionales, creación, recorridos, inserción y eliminación de elementos.
- Realizar Implementaciones dinámicas y estáticas.

UNIDAD 4: MANEJO DE ARCHIVOS BINARIOS Y DE TEXTO.**Objetivos Específicos**

- Incorporar concepto de archivo.
- Realizar clasificación por acceso: entrada, salida, entrada/salida.
- Trabajar con acceso de los registros: directo y secuencial.
- Realizar operaciones sobre archivos: Acceso, lectura, escritura y cierre de archivos.
- Proporcionar ejemplos y realizar ejercicios.



UNIDAD 5: PILAS DEFINICIÓN Y TERMINOLOGÍA DE PILA.

Objetivos Específicos

- Trabajar con ADT pila. Realizar Operaciones básicas.
- Realizar implementación dinámica mediante punteros e implementación estática mediante arreglos. Realizar comparación de las implementaciones.
- Poder aplicar los contenidos.

UNIDAD 6: COLAS

Objetivos Específicos

- Incorporar definición y terminología de colas
- Trabajar con ADT cola. Realizar Operaciones básicas.
- Realizar implementación dinámica: simple y circular.
- Realizar implementación estática: simple, circular. Realizar comparación de las implementaciones.
- Poder aplicar los contenidos.

UNIDAD 7: LISTAS

Objetivos Específicos

- Trabajar con definición y terminología de lista.
- Trabajar con ADT lista ordenada. Realizar operaciones básicas.
- Realizar implementación dinámica e Implementación estática. Realizar comparación de las implementaciones.
- Poder aplicar los contenidos.

UNIDAD 8: ARBOLES

Objetivos Específicos

- Incorporar definición de árbol.
- Trabajar con árbol binario. Realizar representación de árboles binarios y árbol binario de expresión. Realizar representación de listas como árboles binarios.
- Trabajar con árboles y sus aplicaciones.
- Incorporar otros tipos de árboles: árboles AVL y árboles de búsqueda de m-vías.



UNIDAD 9: GRAFOS

Objetivos Específicos

- Trabajar con Grafos y sus aplicaciones. TAD grafo.
- Realizar recorridos, aplicaciones y ejemplos

Bibliografía obligatoria:

- Chris Okasaki, Purely Functional Data Structures. Cambridge University Press, 1998.
- Mark Allen Weiss , Data Structures and Algorithm in C. Addison-Wesley, 1997 (2da edición).

Bibliografía de consulta:

- Thomas Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein , Introduction to Algorithms. MIT Press, 2001 (3era edición).
- Peter A. Fejer y Dan A. Simovici, Mathematical Foundations of Computer Science. Vol.I: Sets, Relations, and Induction. Springer Verlag, 1991.

Uso del campus virtual.

El Campus Virtual es un espacio fundamental para el desarrollo de la asignatura. En el aula virtual se propondrá material educativo, apuntes de clase, bibliografía así como también el programa y cronograma de la asignatura y las guías de Trabajos Prácticos y ejercicios.

Modalidad de evaluación:

El sistema normal de evaluación consistirá en exámenes parciales con recuperatorios, según el cronograma previsto, de la totalidad de la materia descripta en el programa. Los mismos se realizarán en las fechas que, a tal efecto, se establezcan en el cronograma correspondiente.

Promoción:

Régimen de aprobación: La aprobación como alumno que promociona se rige por el sistema de exámenes parciales

Se requiere como condición de promoción, que el alumno

- Asista al 75 % de las clases dictadas.
- Firme los Trabajos Prácticos, con la entrega de dos aplicaciones
- Apruebe los dos exámenes parciales escritos (pautados en el cronograma) o la aprobación de los respectivos recuperatorios. Los exámenes parciales abarcan temas teóricos-prácticos



desarrollados en el período de clases. La elección y corrección de los temas de los parciales es responsabilidad del profesor a cargo del curso. Las notas deberán ser mayores o iguales a seis en ambos parciales y el promedio entre ambos mayor o igual que siete.

Regular

Régimen de aprobación: La aprobación como alumno regular se rige por el sistema de exámenes finales, previa regularidad de la asignatura

Se requiere como condición de regularidad, que el alumno

- Asista al 75 % de las clases dictadas.
- Firme los Trabajos Prácticos, con la entrega de dos aplicaciones
- Apruebe los dos exámenes parciales escritos (pautados en el cronograma) o la aprobación de los respectivos recuperatorios. Los exámenes parciales abarcan temas teóricos-prácticos desarrollados en el período de clases. La elección y corrección de los temas de los parciales es responsabilidad del profesor a cargo del curso. Las notas deberán ser mayores o iguales a cuatro y menores a seis

Examen Final: Será escrito u oral y abarca temas de desarrollo teórico y prácticos

Para la evaluación de exámenes parciales (y/o recuperatorios) y final se tendrá en cuenta: El

procedimiento, los desarrollos y los resultados de los ejercicios y problemas planteados

- La justificación y análisis de los resultados
- La clara y correcta expresión de las ideas.
- Cabe señalar que luego de cada unidad temática el alumno desarrollará una autoevaluación que le permitirá ir ajustando su proceso de aprendizaje.

Libre:

Examen: El examen será escrito para la parte práctica y de aprobarse ésta se realizará en forma escrita y oral la parte teórica.

Para la evaluación se tendrá en cuenta:

Para la parte práctica: La resolución correcta de los ejercicios propuestos.

El razonamiento seguido en la elaboración de los mismos, la exactitud y la precisión.

**Instituto de
Tecnología e Ingeniería**

Para la parte teórica: El correcto desarrollo de los temas propuestos y capacidad para conectar temas e interrelacionarlos.