

Fecha de aprobación del programa: 30/10/2018

Vencimiento: 30/10/2020

# Estructuras de Datos

**Actividad curricular:** Estructuras de Datos**Carrera:** Licenciatura en Informática**Área:** Teoría de la computación**Prerrequisitos:** Introducción a la programación**Carga Horaria:**

- Carga horaria total 128 horas
- Carga horaria práctica: 96 hs
  - Formación Experimental: 16hs
  - Resolución de problemas: 80hs
- 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 abstractos de datos. Tipos algebraicos: maybe, either, enumerativos, listas, árboles binarios, árboles generales. Estructuras de datos. Estructuras contenedoras: pilas, colas, diccionarios, heaps, árboles balanceados, contenedores basados en representaciones numéricas. Representación de datos en memoria. 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 recorrido, ordenamiento, búsqueda, ordenamiento y actualización. Clasificación e implementación. Nociones básicas de algoritmos sobre grafos.

**Programa analítico:****UNIDAD 1: INTRODUCCIÓN****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.

**Instituto de  
Tecnología e Ingeniería**

- Trabajar con Arrays unidimensionales y Bidimensionales, creación, recorridos, inserción y eliminación de elementos.
- Realizar Implementaciones dinámicas y estáticas.

**UNIDAD 4: 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 5: 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 6: 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 7: ÁRBOLES****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 8: GRAFOS****Objetivos Específicos**

- Trabajar con Grafos y sus aplicaciones. TAD grafo.



**Instituto de  
Tecnología e Ingeniería**

- 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:**

Consistirá en dos 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 establezcan en el cronograma correspondiente.

- La calificación de cada evaluación se determinará en la escala 0 a 10, con los siguientes valores: 0, 1, 2 y 3: insuficientes; 4 y 5 regular; 6 y 7 bueno; 8 y 9 distinguido; 10 sobresaliente. La materia podrá aprobarse mediante: régimen de promoción directa, exámenes finales regulares y exámenes libres.
- Régimen de promoción directa (sin examen final): los/las estudiantes deberán aprobar las materias con siete (7) o más puntos de promedio entre todas las instancias evaluativas, sean éstas parciales o sus recuperatorios, debiendo tener una nota igual o mayor a seis (6) puntos en cada una de éstas. Todas las instancias evaluativas tendrán una posibilidad de recuperación. En el caso de los ausentes en la fecha original, el recuperatorio operará como única fecha de examen. El examen recuperatorio permite mantener la chance de la promoción siempre y cuando respete las condiciones de calificación respectiva.
- Exámenes finales regulares: para aquellos/as estudiantes que hayan obtenido una calificación de al menos de 4 (cuatro) y no se encuentren en las condiciones de promoción, deberán rendir un examen final que se aprobará con una nota no inferior a 4 (cuatro) puntos.
- La asistencia no debe ser inferior al 75% en las clases presenciales.

94