Unidad I: Introducción a las estructuras de datos

1.1 Tipos de datos abstractos (TDA)

Los tipos de datos abstractos (TDA) encapsulan datos y funciones que trabajan con estos datos. Los datos no son visibles para el usuario en un tipo de dato abstracto y el acceso a los datos es exclusivamente bajo el llamado a funciones, también llamadas métodos. Así, el tipo de dato abstracto es especificado por los métodos, no por los datos. En C++, los tipos de datos abstractos son representados por clases, las cuales presentan a pequeña deficiencia: el dato que representa el estado de un objeto de este tipo de dato abstracto es visible (algunas veces no accesible) en la parte private de la clase declarada para cada programa, la clase es reconocida mediante la vía#include. Ejemplos de tipos de datos abstractos son: stack, que, etc Los TDA por lo general manejan memoria dinámica, esto es, la asignación dinámica de memoria es una característica que le permite al usuario crear tipos de datos y estructuras de cuaLquier tamaño de acuerdo a las necesidades que se tengan en el programa

1.2 Modularidad

Modularidad en Ciencias de la computación es la característica por la cual un programa de computador está compuesto de porciones que se conocen como módulos. El diseño estructurado es la técnica de diseño de algoritmos en que se basa la programación modular, paradigma de programación que persigue desarrollar programas modulares.

La modularidad se basa en la descomposición de un problema en una serie de sub problemas; dividiéndolo en módulos que resultan de segmentar el problema en funciones lógicas que son perfectamente diferenciadas. Esta división exige la presencia de un módulo denominado módulo de base o principal a objeto de que controle y se relacione con los demás.

Es una técnica de programación que todavía se utiliza tanto para la construcción de algoritmos computacionales básicos así como apoyo al desarrollo de sistemas de gestión (en el diseño de diagramas modulares).

1.3 Uso de TDA

Un algoritmo es una secuencia finita de operaciones, organizadas para realizar una tarea determinada.

Las estructuras de datos son la forma en que se organizan los datos para ser usados.

Puede ser una colección de variables, posiblemente de diferentes tipos de datos, conectadas de un modo determinado.

Una estructura de datos bien organizada debe permitir realizar un conjunto de acciones sobre los datos de tal forma de minimizar el uso de los recursos y el tiempo empleado para efectuar la operación.

Abstracción

La abstracción es un mecanismo fundamental para la comprensión de fenómenos o situaciones que implican gran cantidad de detalles.

Abstracción es la capacidad de manejar un objeto (tema o idea) como un concepto general, sin considerar la enorme cantidad de detalles que pueden estar asociados con dicho objeto.

Ejemplo, se puede saber conducir un automóvil sin conocer el tipo del modelo o cómo está fabricado.

La abstracción se utiliza para suprimir detalles irrelevantes, mientras se enfatiza en los relevantes o significativos.

El beneficio principal de la abstracción es que facilita al programador pensar acerca del problema a resolver. Uno de los principios importantes del diseño de software es el de la abstracción y ocultación de la información.

Abstracción de datos es una técnica que permite inventar nuevos tipos de datos que sean más adecuados a una aplicación y, por consiguiente, facilitar la escritura del programa

1.4 Manejo de memoria estática.

Para implementar alguna estructura de datos, primero es necesario tener muy claro cómo va a ser el manejo de memoria.

La diferencia entre estructuras estáticas y dinámicas esta en el manejo de memoria.

En la memoria estática durante la ejecución del programa el tamaño de la estructura no cambia.

La estructura que maneja memoria estática son los vectores.

Un vector es una colección finita, homogénea y ordenada de elementos.

Es finita porque todo arreglo tiene un límite, homogénea porque todos los elementos son del mismo tipo y ordenada porque se puede determinar cuál es el enésimo elemento.

Un vector tiene dos partes: Componentes e índices

Los componentes hacen referencia a los elementos que forman el arreglo y los índices permiten referirse a los componentes del arreglo en forma individual.

Los arreglos se clasifican en:

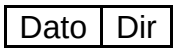
- Unidimensionales (vectores o listas)
- Bidimensionales (matrices o tablas)
- Multidimensionales

1.5 Manejo de memoria dinámica

En la memoria dinámica durante la ejecución del programa el tamaño de la estructura puede cambiar.

La *memoria dinámica*, es el espacio de almacenamiento que solicita una clase o método en tiempo de ejecución. De esa manera, a medida que el proceso requiere de más espacio se solicita al sistema operativo, sin que el proceso se preocupe por donde serán asignados los datos, ni que espacios de memoria nos entregara el sistema operativo.

Así como existen estructuras de datos estáticas (arreglos), también existen estructuras de datos dinámicas (listas y árboles), estas últimas son generadas a partir de un tipo de dato conocido como referencia (dirección de memoria). Para utilizar las referencias se requiere de un elemento llamado nodo, el cual se estructura de la siguiente manera.

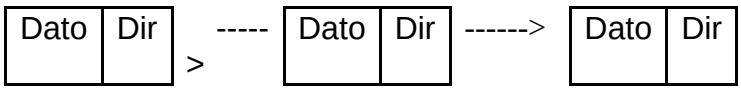


Nodo con una referencia

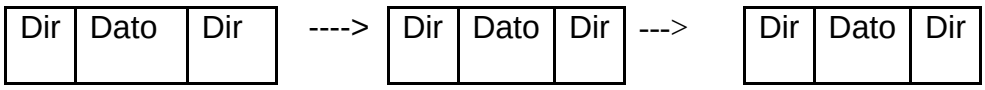
Nodo con dos referencias

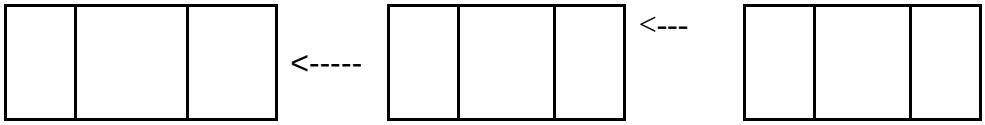
Las estructuras de datos que usan nodos pueden ser lineales o no lineales, dentro de las lineales se encuentran las listas simples y dobles y en las no lineales encontramos los árboles, estas estructuras se representan de la siguiente forma.

Lista simple.



Lista doble





Árbol

Dir	Dato	Dir