



Título:	Tipos de Estructuras de Datos
Tipo:	Monografía
Autor:	Franklin Alexis Díaz Díaz
Fecha:	Noviembre 2018
Palabras claves:	Lista, pila, cola, árbol
Aprobado por:	Decano de la Facultad de Ingeniería Mg. Ing. Enrique Durand Bazán
Firma:	

INTRODUCCIÓN

La principal característica de las estructuras de datos es la facilidad de variar su tamaño. Por ende, se distinguen de las estructuras básicas como arreglos y matrices, quienes una vez definido su tamaño, no se puede modificar de manera automático, sino por medio de funcionalidades que uno implementa.

Las estructuras de datos realizan una asignación dinámica de su propia capacidad de almacenamiento, para lo cual el compilador del programa asigna una cantidad de memoria fija para mantener la dirección de la estructura asignado dinámicamente, en lugar de asignar una dirección para la estructura misma.

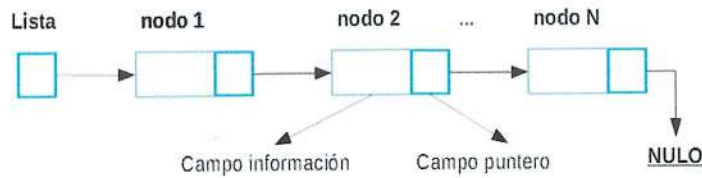
Cuando se asigna memoria de forma dinámica a un objeto de tipo estructura, se devuelve la referencia a la zona de la memoria que le ha sido asignado.

Dentro de las estructuras de datos, existen listas, pilas, colas árboles, entre otros y derivaciones de las mencionadas, que trabajan de manera dinámica en la asignación de la memoria. Cada una de ellas tiene funcionalidades y métodos distintos, lo cual implica que el uso que se les da, es diferente para cada uno de ellos.



Estructura de datos tipo Lista

Es un tipo de estructura de datos que almacenan datos de una forma organizada de forma dinámica, donde cada elemento de la lista apunta al siguiente a excepción del último elemento que no tiene sucesor y apunta a null. Las listas permiten representar un grupo de elementos presentados como una secuencia.

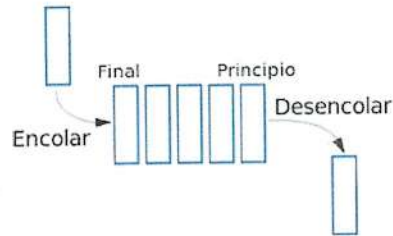


CLASE NODO	CLASE LISTA
<pre> public class Nodo { Object valor; Nodo siguiente; public Nodo(Object valor) { this.valor = valor; this.siguiente = null; } public void enlazarSiguiente(Nodo n) { this.siguiente = n; } public Nodo obtenerSiguiente() { return siguiente; } public Object getValor() { return valor; } } </pre>	<pre> public class Lista { Nodo cabeza; int cantidad; public Lista() { cabeza = null; cantidad = 0; } public void agregarPrimero(Object obj) { if(cabeza==null) cabeza = new Nodo(obj); else { Nodo temp = cabeza; Nodo nuevo = new Nodo(obj); nuevo.enlazarSiguiente(temp); cabeza = nuevo; cantidad++; } } public void eliminarNodo(int pos) { if(pos == 0) { cabeza = cabeza.obtenerSiguiente(); } else { int contador = 0; Nodo temporal = cabeza; while(contador < pos - 1) { temporal = temporal.obtenerSiguiente(); contador++; } temporal.enlazarSiguiente (temporal.obtenerSiguiente().obtenerSiguiente()); cantidad--; } } } </pre>



Estructura de datos tipo Cola

Una estructura de datos cola es lugar para almacenar datos, donde estos datos se almacenan una detrás de otra (imagínese la cola de un banco) y para acceder a cualquier dato, empieza a recorrer desde el primer elemento.



Funcionamiento de una Cola

Una estructura de datos cola funciona como una cola de personas a la espera de su turno para ser atendidos por alguien o algo (imagínese un cajero automático), donde la primera persona de la cola es la primera en ser atendida y cuando llega una persona se tiene que ubicar al final de la cola y esperar a ser atendida..

Operaciones a realizar en una Pila

- Insertar un objeto en la cola (encolar).
- Eliminar un objeto de la cola (desencolar).
- Obtener el primer objeto de la cola.
- Buscar algún objeto almacenado en la cola.
- Devolver el número de objetos almacenados en la cola.
- Otros

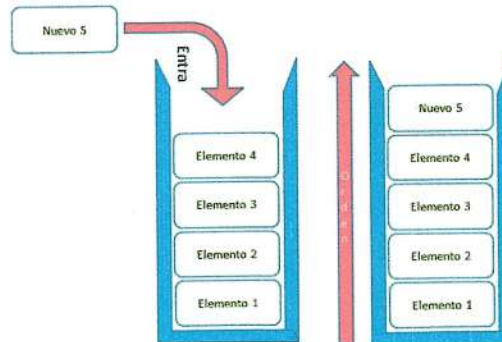
CLASE NODO	CLASE COLA
<pre>public class Nodo { Artículo art; Nodo siguiente; public Nodo(Artículo art) { this.art = art; this.siguiente = null; } }</pre>	<pre>public class Cola { public Nodo cabeza, ultimo; public Cola() { cabeza = null; ultimo = null; } public void encolar(Artículo art) { Nodo nuevo = new Nodo(art); if(cabeza == null) cabeza = nuevo; else ultimo.siguiente = nuevo; ultimo = nuevo; } }</pre>



	}
--	---

Estructura de datos tipo Pila

La estructura de datos pila o stack (por su denominación en inglés), es una estructura lineal que solo tienen un único punto de acceso mediante el cual se ingresan, eliminan o consultan los objetos de la misma. Utiliza el acrónimo LIFO (Last In First Out), que significa el último en entrar, es el primero en salir.



Características de una Pila

- La única forma de acceder a los elementos es desde el la cima o último elemento de la pila.
- Posee pocas operaciones.
- Si desea acceder a un elemento que no está en la parte superior de la pila, deseche la pila en la pila adicional. Cuando se complete la operación del elemento, deseche los elementos de la pila adicional en la pila original.

Operaciones a realizar en una Pila

- push(): añade elemento al final
- pop(): elimina último elemento
- peek(): consulta el último elemento
- empty(): verifica si está vacía

Otros



CLASE NODO	CLASE PILA
<pre>public class Nodo { public Web web; public Nodo siguiente; public Nodo(Web web) { this.web = web; this.siguiente = null; } }</pre>	<pre>public class Pila { Nodo cima; int cantidad; public Pila() { cima = null; cantidad = 0; } }</pre>

}	<pre>public void push(Web web) { Nodo nodo = new Nodo(web); nodo.siguiete = cima; cima = nodo; cantidad++; } public Web peek() { if(cima == null) return null; return cima.web; } public void pop() { if(cima != null) { Nodo temporal = cima; cima = cima.siguiete; temporal.siguiete = null; cantidad--; } }</pre>
---	---

Estructura de datos tipo Árbol

La estructura de dato Árbol es posiblemente de las más utilizadas, pero a la vez una de las más complejas, ya que se caracterizan por almacenar sus nodos (objetos) en forma vertical y no en forma horizontal como las estructuras que se detalla líneas arriba.

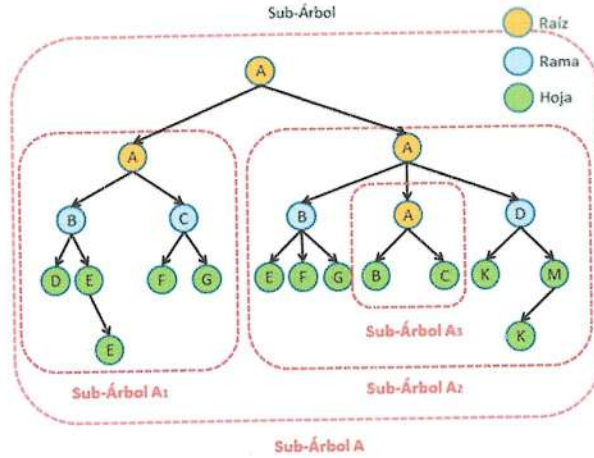
Compuesto por:

- Nodos: son los elementos del árbol.
- Nodo raíz: primer nodo del Árbol, y es único, lo que significa que no puede haber más de 1 nodo raíz.
- Nodo padre: son nodos que tienen al menos otro nodo dependiente.
- Nodo hijo: son nodos que tienen un nodo padre.
- Nodo hermano: son nodos que comparten al mismo nodo padre.
- Nodo hoja: son nodos que no tienen hijos, considerados los nodos de último nivel.
- Nodo rama: son nodos que no son nodo raíz y tiene por lo menos un nodo hijo.

Ámbitos de los árboles:

- Nivel: es cada nivel que aparece cuando los nodos van teniendo nodos hijos.
 - ✓ Sin un árbol carece de nivel, se le considera "árbol vacío"
 - ✓ El primer nivel le corresponde al nodo raíz.
 - ✓ Para hallar el nivel de cada nodo, se cuanta la cantidad de nodos que existen sobre él mismo, hasta llegar a la raíz aumentado en 1.
- Altura: corresponde al número máximo de niveles.

Los árboles está compuesto por subárboles que es la agrupación de nodos por niveles, y éstos a su vez van formando mas subárboles.



Existen 3 formas de recorrer los datos de un Árbol:

- **PreOrden:** inicia por el nodo raíz, de ahí recorre el subárbol del lado izquierdo y termina recorriendo el subárbol derecho. También puede iniciar siempre en nodo raíz, recorre subárbol derecho y termina por la izquierda.
- **PostOrden:** inicia por el subárbol izquierdo, luego recorre el subárbol derecho y termina en la raíz. También puede comenzar por el subárbol derecho.
- **InOrden:** inicia por el subárbol izquierdo, llega al nodo raíz y termina en el subárbol derecho. También puede comenzar por el subárbol derecho.

Operaciones a realizar en un Árbol

- Agregar
- Recorrer
- Eliminar
- Buscar menor, mayor
- Otros





BIBLIOGRAFÍA

- Drozdek, A. (2007). Estructura de Datos Y Algoritmos en Java. México: CENGAGE Learning.
- Estructuras de datos y algoritmos fundamentales. (2020). (n.p.): Editorial Digital del Tecnológico de Monterrey.
- FERNANDEZ AZUELA, M. A., SANCHEZ GARCIA, L. A., Joyanes Aguilar, L., Zahonero Martínez, I. (2005). Estructuras de Datos en C. España: MCGRAW-HILL.
- Garrido Carrillo, A., Fernández-Valdivia, J. (2006). Abstracción y Estructuras de Datos en C++. España: Delta Publicaciones.
- López, B. (2012). Estructuras de datos orientadas a objetos. Colombia: México.
- Ullman, J. D., Aho, A. V., Hopcroft, J. E. (1999). Estructuras de datos y algoritmos. México: Pearson Educación.
- Vancells i Flotats, J. Algoritmos y programas. UOC Papers, 2002. Digitalia, <https://www.digitaliapublishing.com/a/5797>
- Verdejo López, J. A., Martí Oliet, N., Ortega Mallén, Y. (2004). Estructuras de datos y métodos algorítmicos: ejercicios resueltos. España: Pearson Educación.

