

Asignación de memoria

La **asignación de memoria** consiste en el proceso de asignar memoria para propósitos específicos, ya sea en tiempo de compilación o de ejecución.

La **asignación estática de memoria** consiste en el proceso de asignar memoria en tiempo de compilación antes de que el programa asociado sea ejecutado, a diferencia de la **asignación dinámica de memoria** donde la memoria se asigna a medida que se necesita en tiempo de ejecución.

Estructuras Dinámicas de Datos

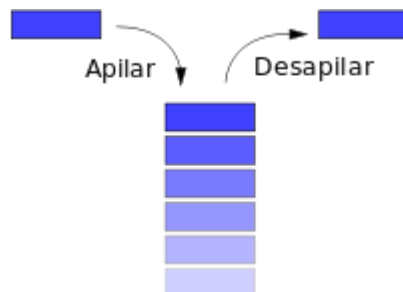
Una **estructura de datos dinámica** es aquella en la que el tamaño ocupado en memoria puede modificarse durante la ejecución del programa.

Pilas

Una **pila** (*stack* en inglés) estructura de datos que permite almacenar y recuperar datos, siendo el modo de acceso a sus elementos de tipo LIFO (del inglés *Last In, First Out*, «último en entrar, primero en salir»). Esta estructura se aplica en multitud de supuestos en el área de informática debido a su simplicidad y capacidad de dar respuesta a numerosos procesos.

Para el manejo de los datos cuenta con dos operaciones básicas: **apilar** (*push*), que coloca un objeto en la pila, y su operación inversa, **retirar** (o desapilar, *pop*), que retira el último elemento apilado.

En cada momento solamente se tiene acceso a la parte superior de la pila, es decir, al último objeto apilado. La operación **retirar** permite la obtención de este elemento, que es retirado de la pila permitiendo el acceso al anterior (apilado con anterioridad), que pasa a ser el que ahora está en el tope de la pila.



Las **pilas** suelen emplearse en los siguientes contextos:

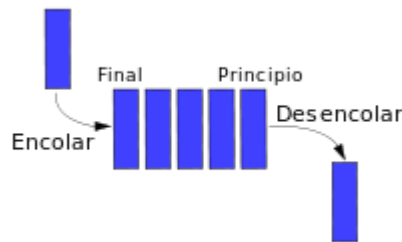
- Evaluación de expresiones en notación postfija (notación polaca inversa).
- Reconocedores sintácticos de lenguajes independientes del contexto.
- Implementación de recursividad.

En un sistema operativo cada proceso tiene un espacio de memoria (pila) para almacenar valores y llamadas a funciones.

Por analogía con objetos cotidianos, una operación **apilar** equivaldría a colocar un plato sobre una pila de platos, y una operación **retirar** equivaldría a retirarlo.

Colas

Una **cola** es una estructura de datos, caracterizada por ser una secuencia de elementos en la que la operación de inserción *push* se realiza por un extremo y la operación de extracción *pull* por el otro. También se le llama estructura FIFO (del inglés *First In First Out*), debido a que el primer elemento en entrar será también el primero en salir.



La particularidad de una estructura de datos de cola es el hecho de que solo podemos acceder al primer y al último elemento de la estructura. Así mismo, los elementos solo se pueden eliminar por el principio y solo se pueden añadir por el final de la cola.

| | | | | | |
|----|----|---|-------|----|----|
| 15 | 20 | 9 | | 18 | 19 |
|----|----|---|-------|----|----|

1.Ejemplo de Cola

| | | | | | |
|----|----|---|-------|----|----|
| 15 | 20 | 9 | | 18 | 19 |
|----|----|---|-------|----|----|

2.Vamos a Insertar el 13 en la Cola.

| | | | | | | |
|----|----|---|-------|----|----|----|
| 15 | 20 | 9 | | 18 | 19 | 13 |
|----|----|---|-------|----|----|----|

3.Sacamos el frente de la Cola (15)

| | | | | | |
|----|---|-------|----|----|----|
| 20 | 9 | | 18 | 19 | 13 |
|----|---|-------|----|----|----|

Ejemplos de colas en la vida real serían: cola de personas en la caja de en un supermercado, cola de personas esperando para entrar a ver un partido de fútbol, cola de vehículos esperando para entrar en un estacionamiento. La idea esencial es que son todas líneas de espera.

Listas Enlazadas

Una **lista enlazada** es una de las estructuras de datos fundamentales, y puede ser usada para implementar otras estructuras de datos. Consiste en una secuencia de nodos, en los que se guardan campos de datos arbitrarios y una o dos referencias, enlaces o punteros al nodo anterior o posterior. El principal beneficio de las listas enlazadas respecto a los vectores convencionales es que el orden de los elementos enlazados puede ser diferente al orden de almacenamiento en la memoria, permitiendo que el orden de recorrido de la lista sea diferente al de almacenamiento.

Una lista enlazada es un tipo de dato autorreferenciado porque contienen un puntero o enlace (en inglés *link*, del mismo significado) a otro dato del mismo tipo. Las listas enlazadas permiten inserciones y eliminación de nodos en cualquier punto de la lista, pero no permiten un acceso aleatorio.

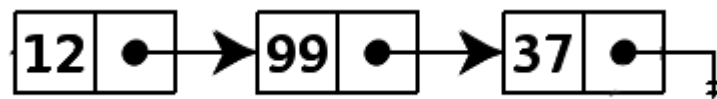
Las listas enlazadas son usadas como módulos para otras muchas estructuras de datos, tales como pilas, colas y sus variaciones.

El campo de datos de un nodo puede ser otra lista enlazada.

Existen diferentes tipos de listas enlazadas: listas enlazadas simples, listas doblemente enlazadas, listas enlazadas circulares y listas enlazadas doblemente circulares.

Listas simples enlazadas

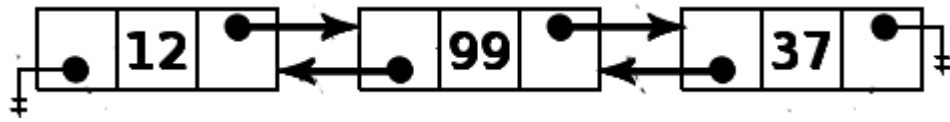
Es una lista enlazada de nodos, donde cada nodo tiene un único campo de enlace. Una variable de referencia contiene una referencia al primer nodo, cada nodo (excepto el último) enlaza con el nodo siguiente, y el enlace del último nodo contiene NULL para indicar el final de la lista.



Una lista simplemente enlazada que contiene tres valores enteros

Listas doblemente enlazadas

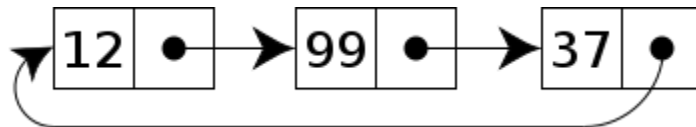
Un tipo de lista enlazada más sofisticado es la **lista doblemente enlazada** o **lista enlazadas de dos vías**. Cada nodo tiene dos enlaces: uno apunta al nodo anterior, o apunta al valor NULL si es el primer nodo; y otro que apunta al nodo siguiente, o apunta al valor NULL si es el último nodo.



Una lista doblemente enlazada que contiene tres valores enteros

Listas enlazadas circulares

En una lista enlazada circular, el primer y el último nodo están unidos. Esto se puede hacer tanto para listas enlazadas simples como para las doblemente enlazadas. Para recorrer una lista enlazada circular podemos empezar por cualquier nodo y seguir hasta que se regrese hasta el nodo original. Desde otro punto de vista, las listas enlazadas circulares pueden ser vistas como listas sin comienzo ni fin.



Una lista enlazada circular que contiene tres valores enteros