República Bolivariana de Venezuela

Universidad Rafael Urdaneta

Facultad de Ingeniería

URU 2022 CATEDRA DE ESTRUCTURA DINAMICA DE DATOS



TRABAJO COLAS Y LISTAS

PROFESOR: ELI MORA

URU PERIODO 2022-B

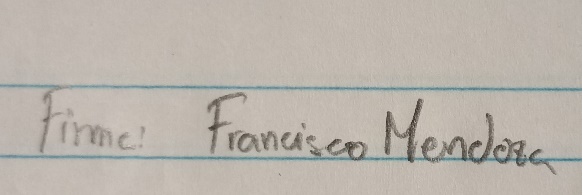
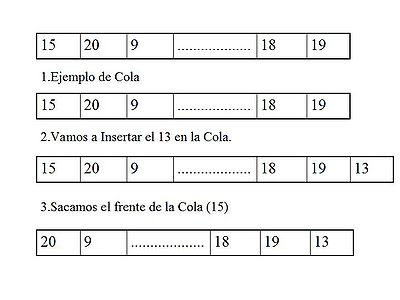


Foto de la cedula. Foto actual. Firma del autor.

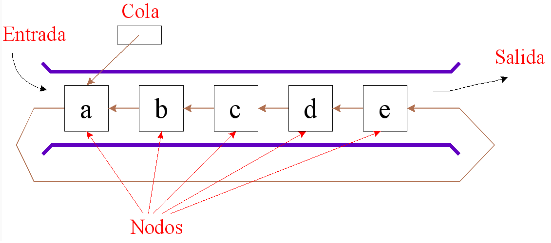
**Una cola:**

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 *pop* 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.

Las colas se utilizan en sistemas informáticos, transportes y operaciones de investigación (entre otros), dónde los objetos, personas o eventos son tomados como datos que se almacenan y se guardan mediante colas para su posterior procesamiento. Este tipo de estructura de datos abstracta se implementa en lenguajes orientados a objetos mediante clases, en forma de listas enlazadas.



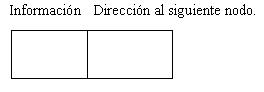
La particularidad de una estructura de datos de cola es el hecho de que sólo podemos acceder al primer y al último elemento de la estructura. Así mismo, los elementos sólo se pueden eliminar por el principio y sólo se pueden añadir por el final de la cola.



**Listas:**

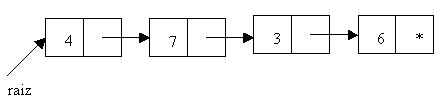
Una lista es un conjunto de nodos, cada uno de los cuales tiene dos campos: uno de información y un apuntador al siguiente nodo de la lista. Además, un apuntador externo señala el primer nodo de la lista.

Representación gráfica de un nodo:



La información puede ser cualquier tipo de dato simple, estructura de datos o inclusive uno o más objetos.  
La dirección al siguiente nodo es un puntero.

Representación gráfica de una lista:



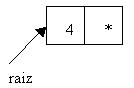
Como decíamos, una lista es una secuencia de nodos (en este caso cuatro nodos). La información de los nodos en este caso es un entero y siempre contiene un puntero que guarda la dirección del siguiente nodo.  
raiz es otro puntero externo a la lista que contiene la dirección del primer nodo.

El estado de una lista varía durante la ejecución del programa:

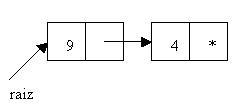


De esta forma representamos gráficamente una lista vacía.

Si insertamos un nodo en la lista quedaría luego:



Si insertamos otro nodo al principio con el valor 9 tenemos:



Lo mismo podemos borrar nodos de cualquier parte de la lista.  
Esto nos trae a la mente el primer problema planteado: el desarrollo del procesador de texto. Podríamos utilizar una lista que inicialmente estuviera vacía e introdujéramos un nuevo nodo con cada línea que tipea el operador. Con esta estructura haremos un uso muy eficiente de la memoria.