**REPÚBLICA BOLIVARIANA DE VENEZUELA**

**UNIVERSIDAD RAFAEL URDANETA**

**FACULTAD DE INGENIERÍA**

**CÁTEDRA DE ESTRUCTURAS DINÁMICAS DE DATOS**

**URU 2022(B)**



**INFORME COLAS Y LISTAS**

**PROFESOR: ELI MORA**

**DIEGO RINCÓN C.I.: 29.929.768**

**MARACAIBO, 09/07/2022**

**1. ¿Qué son las listas?**

Una lista es una estructura de datos que utiliza memoria dinámica. Permite representar elementos del mismo tipo en secuencia, pero a diferencia de los arreglos, esta estructura tiene tamaño variable, lo cual significa que podemos insertar y eliminar datos en ejecución. Además, las posiciones en memoria de los elementos no son secuenciales. Necesita un puntero que apunte hacia el inicio de la lista y a partir de este, se puede ingresar al resto.

**2. ¿Cómo se insertan elementos en una lista?**

Para insertar un nuevo elemento en la lista, normalmente se debe jugar un poco con punteros auxiliares que nos permitan establecer los cambios de posición que habrá que hacer. En general, puede haber tres situaciones posibles para insertar un elemento en una lista: insertar al inicio, insertar al final, insertar por medio de la lista. Lo primero que se suele hacer en cualquier caso es crear el nodo con las especificaciones que necesitemos.

* **Insertar al inicio:** Para insertar un elemento al inicio de la lista, primero se crea el nodo que se desea utilizar y se comienza con las operaciones de puntero. En este caso, el nodo siguiente al nuevo nodo que creamos será el que actualmente es el primero, quedando ahora en segunda posición, ya que el nodo nuevo será el primero después del cambio. Una vez hecho esto, en caso de tener una variable que tenga la longitud de la lista, esta se aumenta.
* **Insertar al final:** Al insertar un elemento en cualquier otra posición que no sea la de inicio, primero se debe verificar que la lista no esté vacía. En caso de estarlo, simplemente se le asigna al puntero que apunta hacia el inicio de la lista el nuevo nodo creado por nosotros. Para insertar al final, se debe recorrer toda la lista haciendo uso de un nodo auxiliar; una vez que nos encontremos en el último nodo actual de la lista, su siguiente será nuestro nuevo nodo, siendo ahora este el último de la lista. De nuevo, se incrementa la variable de longitud de la lista en caso de haberla.
* **Insertar por medio de la lista:** Como antes, primero se debe verificar que la lista no esté vacía, ya que, de estarlo, simplemente hay que asignar el nuevo nodo como primero de la lista. Para poder insertar un elemento en una posición de la lista se debe tomar la posición deseada; luego, nos moveremos por la lista usando un puntero auxiliar mientras que la posición sea inferior a la solicitada y que la lista tenga los suficientes elementos, detendremos el recorrido en el nodo anterior a la posición solicitada, tomaremos la información de su siguiente nodo y se la asignaremos al nodo nuevo, luego se asigna el nodo nuevo como siguiente al que se tomó al inicio. Por último, se incrementa la longitud de la lista en caso de ser necesario.

**3. ¿Cómo se eliminan elementos en una lista?**

Al igual que como se insertan elementos, para eliminar se debe hacer uso de punteros auxiliares que nos permitirán hacer los intercambios y liberaciones de memoria correspondientes. Para esto, se suele utilizar una función que transforme los datos del nodo a borrar en NULL antes de liberar el espacio en memoria por el nodo. De la misma manera, se tienen tres situaciones generales en las que se pueden eliminar elementos de una lista: eliminar el inicio, eliminar el final, eliminar un elemento en medio de la lista. Antes de hacer cualquier eliminación, se verifica si la lista está vacía, pues en este caso no se podría ejecutar ninguna acción.

* **Eliminar el inicio:** Si se quiere eliminar el primer nodo de la lista, se debe crear una variable auxiliar que guarde al mismo, y luego se asigna el nodo siguiente a este como primero de la lista. Seguidamente, se ejecuta la función para limpiar el nodo y liberarlo. En caso de ser necesario, se disminuye la longitud de la lista.
* **Eliminar al final:** Al eliminar el nodo que está al final se debe verificar el caso en que la lista tenga un solo elemento, si es así, simplemente se guarda su valor en una variable auxiliar y luego se asigna NULL al puntero que apunta hacia el inicio de la lista, para finalizar liberando el nodo. En caso de haber varios elementos, se recorre la lista hasta llegar al penúltimo elemento, luego se asigna el valor del siguiente nodo, el cual sería el último, en una variable auxiliar, se le asigna NULL al valor siguiente del penúltimo nodo, siendo este ahora el último, y se libera el nodo guardado por el puntero auxiliar.
* **Eliminar un elemento al medio de la lista:** Para eliminar un elemento en una posición dada por el usuario, se presentan diversas situaciones, que la posición sea el primer elemento, que sea un elemento del medio o que la posición no exista dentro de la lista. Si la posición indicada es la del primero, se hace lo mismo que con la función de eliminar al inicio. En caso de que sea una posición existente y que sea diferente a la primera posición, se debe recorrer la lista hasta el elemento anterior a la posición indicada, luego se asigna en un puntero auxiliar el elemento a borrar, es decir, el que está en la posición indicada, y al anterior a este se le asigna el siguiente del nodo que se va a eliminar; luego, se libera el nodo en la posición indicada. Por último, en caso de que la posición indicada no exista, no se puede realizar ninguna acción.

Además de todos estos métodos, también es de importancia saber que las listas se pueden **recorrer** completamente sin afectar sus nodos, esto puede ser usado para aplicaciones que dependan del uso que queramos darle al programa, ya sea imprimir la información de los nodos, o más importante, vaciar la lista, que básicamente sería eliminar todos los elementos de la misma y luego liberar la estructura que contiene.

**4. ¿Qué son colas?**

En pocas palabras, una cola es una estructura de datos que utiliza memoria dinámica. Es un tipo de lista abierta en la que solo se puedan insertar elementos por un extremo y eliminarlos por el otro extremo. Se rige mediante el principio FIFO (First In First Out) y utiliza dos punteros, uno que apunta al primer elemento de la cola y otro que apunta al último elemento de la cola.

**5. ¿Cómo se insertan elementos en una cola?**

Para poder insertar elementos en la cola, primero es necesario crear el nodo requerido y llenarlo con la información deseada para propósitos del programa. Luego, ya que en las colas solo se pueden ingresar elementos por un extremo de la misma, el cual vendría siendo el final, se utiliza un método que realizará esta acción.

Primero que nada, el método necesita saber si la cola está vacía, esto es debido a que la acción a realizar cambia dependiendo de ello. En caso de estar vacía, se crea el nuevo nodo y los punteros que apuntan hacia el primer y último elemento de la cola serán asignados a este nuevo nodo, siendo que es el único en la cola ahora. Si no está vacía, se le dice al último nodo de la cola que su siguiente va a ser el nuevo nodo creado y luego se asigna este como último nodo de la cola. En caso de ser necesario, se puede aumentar la variable que tome la totalidad de elementos en la cola.

**6. ¿Cómo se eliminan elementos en una cola?**

Los elementos en una cola se eliminan por el extremo contrario al que se insertan, esto es siguiendo el principio FIFO, de esta manera el primer elemento que haya entrado a la cola, será también el primero en salir. De esta manera, se crea un método que realizará lo siguiente:

Primero va a revisar si la cola se encuentra vacía, ya que, en caso de estarlo, no se puede eliminar nada de la misma. Si no está vacía, entonces se utiliza un nodo auxiliar que va a guardar el primer nodo de la cola, siendo este el que se va a eliminar, luego se verifica si la cola tiene un solo elemento, si es el caso, se les asigna a los punteros de primero y último el valor de NULL; de otro modo, el primero de la lista ahora será el que tenía el anterior nodo como siguiente y se procede a liberar el espacio reservado con el puntero auxiliar. En caso de tener una variable que llegue el total de elementos de la cola, se disminuye la misma.

Al igual que la listas, en las colas también es de importancia saber que estas se pueden **recorrer** completamente sin afectar sus nodos, esto puede ser usado para aplicaciones que dependan del uso que queramos darle al programa, ya sea imprimir la información de los nodos, o más importante, vaciar la cola, que básicamente sería eliminar todos los elementos de la misma y luego liberar la estructura que la contiene.