**Actividad Nº3.**

**Contenido.**

I . Listas Circulares 2

1. Introducción 2
2. Definición 2
3. Operaciones 2
4. Operación de insertar 3
5. Operación de Eliminar 3
6. Algoritmo del programa 4
7. Diagrama de flujo del programa 5
8. Conclusión 6

I.Listas Circulares.

**1-Introducción**

En esta actividad se presentan las listas circulares, este tipo de listas son lineales y dinámicas de datos. Lineal por que a cada elemento le puede seguir sólo otro elemento, dinámicas por que se puede manejar la memoria de manera flexible, sin necesidad de reservar espacio de antelación.

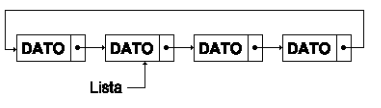
Principalmente se menciona su definición, las operaciones que se pueden realizar con ellas, los algoritmos para realizar la operación de insertar y de eliminar elementos de una lista circular, también se presenta el algoritmo y el diagrama de flujo del programa que implementamos para insertar y eliminar elementos de una lista circular.

**2-Definición.**

Una lista circular es una lista lineal en la que el último nodo a punta al primero (Fig-1).

No existen casos especiales, cada nodo siempre tiene uno anterior y uno siguiente.

En algunas listas circulares se añade un nodo especial de cabecera, de ese modo se evita la única excepción posible, la de que la lista esté vacía.

**Fig-1.**

A pesar de que las listas circulares simplifiquen las operaciones sobre ellas, también introducen algunas complicaciones. Por ejemplo, en un proceso de búsqueda, no es tan sencillo dar por terminada la búsqueda cuando el elemento buscado no existe.

Otra alternativa que se usa a menudo, y que simplifica en cierto modo el uso de listas circulares es crear un nodo especial de hará la función de nodo cabecera. De este modo, la lista nunca estará vacía, y se eliminan casi todos los casos especiales.

**3-Operaciones.**

* Añadir o insertar elementos.
* Buscar o localizar elementos.
* Borrar elementos.
* Moverse a través de la lista.

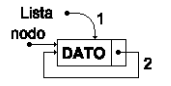
Existen diferentes formas de insertar y de eliminar elementos en un lista circular pero yo me enfoque en las formas más oportunas que podrían ayudar en un futuro en la elaboración de un programa. A continuación se presentaran los algoritmos de la función insertar y la función eliminar en una lista circular, posteriormente se presentará la codificación con su respectivo diagrama de flujo y su algoritmo.

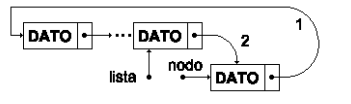
**4-Operación de insertar.**

**Algoritmo.**

1. Si lista está vacía hacemos que lista apunte a nodo.
2. Si lista no está vacía, hacemos que nodo->siguiente apunte a lista->siguiente.
3. Después que lista->siguiente apunte a nodo.

Ejemplos:





**Fig-2.**  **Fig-3.**

**5-Operación de Eliminar.**

**Algoritmo.**

1. Si la lista tiene un solo nodo:

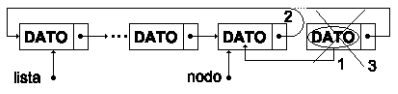
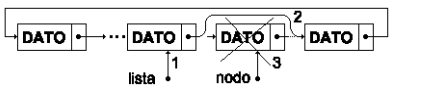
Borrar el nodo lista.

Hacer lista = NULL.

1. Si la lista no tiene un solo nodo entonces:

Hacemos lista -> siguiente = nodo -> siguiente.

Borramos nodo.

Ejemplos:

**Fig-4. Fig-5.**

A continuación se presentará el algoritmo y el diagrama de flujo del programa para insertar y eliminar en una lista circular.

**6-Algoritmo del programa.**

1. Inicio.
2. Se declara la estructura con las variables: elemento de tipo char, \*next struct Circular que nos servirá para enlazar con nodo siguiente.
3. Definimos la estructura como tipo dato y el tipo de dato que generan las listas Circulares.
4. Declaramos las funciones insertar, eliminar, mostrar, menú, introducirDato.
5. Función main, Se declara lista igual a NULL tipo Lista, también declaramos tres variables a, b, c tipo char.
6. Hacer

menú (), función que contiene el menú principal del programa.

imprimir “Acción a realizar: ”

Leer a

selecciona numero switch (a)

Case 1:

imprimir “Introducir elemento:”

Leer b

insertar (&lista, b), función para insertar elementos en la lista.

Case 2:

Si lista != NULL

entonces

mostrar (lista), función muestra los elementos que se encuentran en la lista.

Imprimir “Elemento a Eliminar: ”

Leer c

eliminar (&lista, c), función que elimina el elemento c de la lista si existe.

mostrar (lista), función que muestra la lista de elemento de la lista.

Case 3:

mostrar (lista), función que muestra la lista de elemento de la lista.

Case 4:

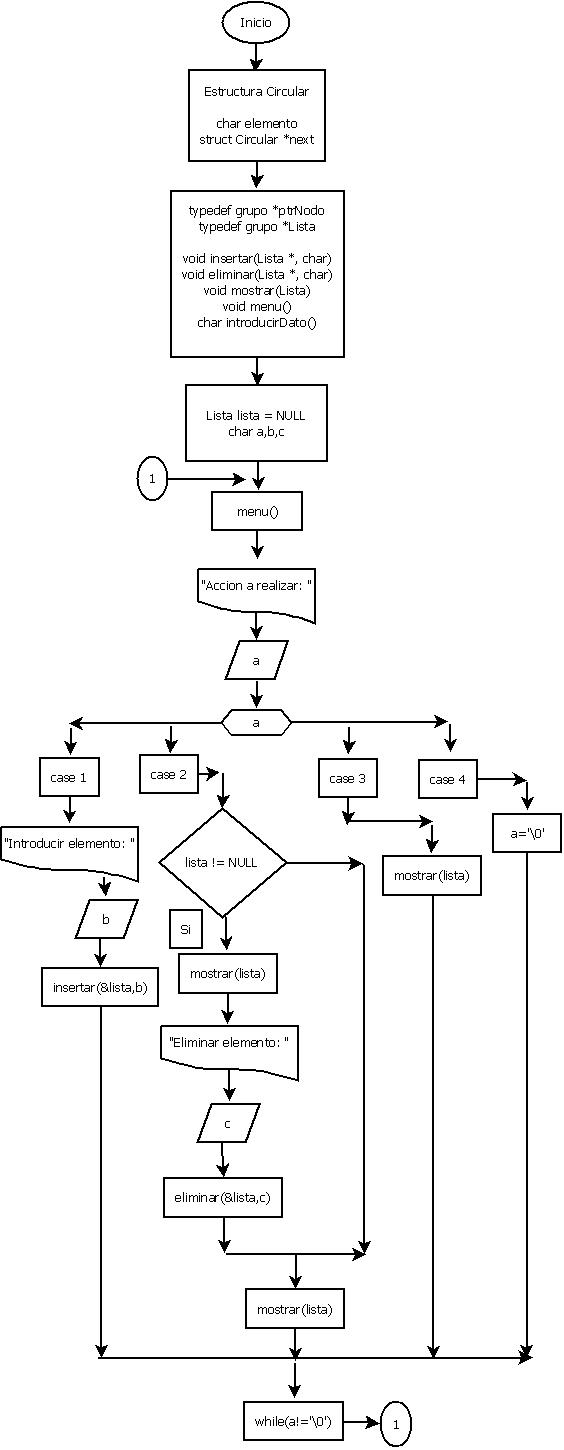
a= ‘\0’, Se inicializa la variable a en ‘\0’ para que salga del ciclo mientras y termine el programa.

Fin selección

Mientras (a!=‘\0’), Cuando la variable sal contenga el valor de ‘0’ se termina el ciclo mientras.

1. Fin.

**7-Diagrama de flujo del programa.**



**8-Conclusión.**

Por medio de las listas circulares tenemos más facilidad de simplificar las operaciones sobre ellas, cosa que en las listas enlazadas no teníamos, pero cabe mencionar que un proceso de búsqueda no estan sencillo dar por terminada la búsqueda cuando el elemento buscado no existe. Más sin embargo tenemos otras alternativas como crear un nodo especial que hará la función de cabecera.

En pocas palabras las listas circulares son muy importantes en aplicaciones que podamos desarrollar pero tiene sus ventajas y desventajas como se mención anteriormente.