

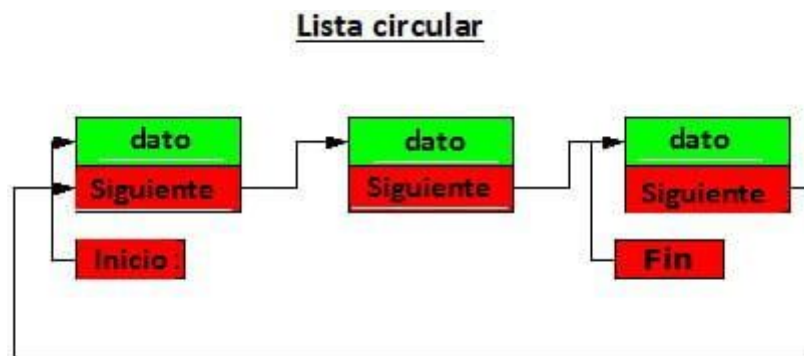
Listas Enlazadas Circulares

La lista circular es una especie de lista enlazada simple o doblemente enlazada, pero que posee una característica adicional para el desplazamiento dentro de la lista: esta no tiene fin. Para que la lista sea sin fin, el puntero siguiente del último elemento apuntará hacia el primer elemento de la lista en lugar de apuntar al valor NULL, como hemos visto en el caso de listas enlazadas simples o doblemente enlazadas.

En las listas circulares, nunca se llega a una posición en la que ya no sea posible desplazarse. Cuando se llegue al último elemento, el desplazamiento volverá a comenzar desde el primer elemento.

Requisitos

Los requisitos son los siguientes: los tipos de datos, las estructuras, el uso de typedef, los punteros, la función usuario, las listas enlazadas simples y las listas doblemente enlazadas.



Construir el modelo de un elemento de la lista

Para definir un elemento de la lista, el tipo struct será utilizado. El elemento de la lista va a tener un campo dato y un puntero siguiente. Este debe ser del mismo tipo que el elemento, en caso contrario no podrá apuntar hacia el elemento. El puntero siguiente permitirá el acceso al próximo elemento.

```
typedef struct ElementoLista {
    char *dato;
    struct ElementoLista *siguiente;
}Elemento;
```

Para tener el control de la lista es mejor guardar ciertos elementos: el primer elemento, el último elemento y el número de elementos. Para ello, otra estructura será utilizada (no es obligatorio, pueden ser utilizadas variables).

```
typedef struct ListaIdentificar {
    Elemento *inicio;
    Elemento *fin;
    int tamaño;
}Lista;
```