

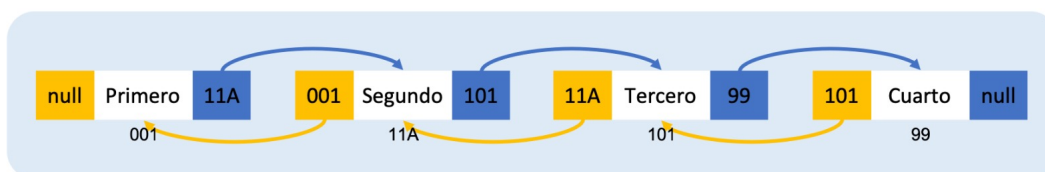
Listas doblemente enlazadas

Algoritmos y Estructuras de Datos

Dra. Violeta Ocegueda

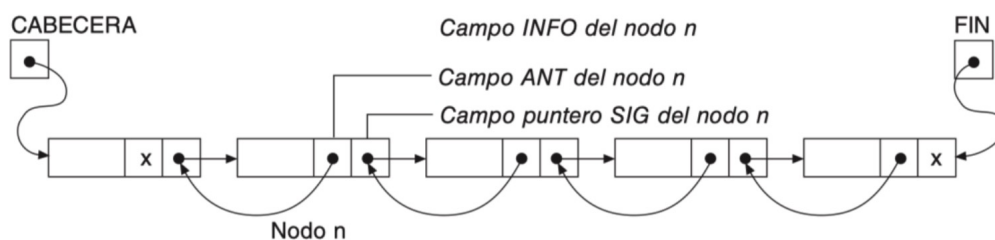
1

- Cada nodo contiene dos apuntadores, uno apunta al nodo que le precede y otro apunta al nodo que le sigue.
- Los nodos cabecera y final tienen referencias a null.
- Al estar doblemente enlazada facilita el recorrido en ambos sentidos.



2

- La variable CABECERA y el puntero SIG permiten recorrer la lista de izquierda a derecha.
- La variable FIN y el puntero ANT permiten recorrerla de derecha a izquierda.



3

Implementación

```
#include <stdio.h>
#include <stdlib.h>
```

```
struct nodo
{ int dato;
  struct nodo * ant;
  struct nodo * sig;
};
```

```
struct nodo * crearnodo(int dato)
{ struct nodo * p;
  p = (struct nodo*) malloc(sizeof (struct nodo));
  p->dato = dato;
  p->ant = NULL;
  p->sig = NULL;
  return(p);
}
```

```
struct nodo* insertar(struct nodo *inicio , int dato)
{ struct nodo *p, *q = NULL, *nuevo;
  p = inicio;
  nuevo = crearnodo (dato);
  while(p!=NULL && dato >= p->dato)
  {
    q = p;
    p = p->sig;
  }
  if (q==NULL) // insertar en la cabecera
  {
    nuevo->sig = p;
    p->ant = nuevo;
    inicio = nuevo;
  }
  else // insertar en medio o al final
  {
    q->sig = nuevo;
    nuevo->ant = q;
    nuevo->sig = p;
    if (p != NULL)
      p->ant = nuevo;
  }
  return(inicio);
}
```

4

Implementación

```
void mostrar(struct nodo *inicio)
{ struct nodo *aux;
  if(!inicio)
    printf("Esta vacia\n");
  else
  { aux=inicio;
    do{
      printf("\n %d",aux->dato);
      aux = aux->sig;
    }while(aux != NULL);
  }
}
```

```
struct nodo* eliminar(struct nodo *inicio, int dato)
{ struct nodo *p, *q = NULL;
  p = inicio;
  while(p!=NULL && p->dato!=dato)
  {
    q = p;
    p = p->sig;
  }
  if(q==NULL)
  { inicio = p->sig;
    if(inicio != NULL)
      inicio->ant = NULL;
    free(p);
  }
  else{
    if(p!=NULL)
    { q->sig = p->sig;
      if(p->sig != NULL)
        p->sig->ant = q;
      free(p);
    }
    else
      printf("No encontrado\n");
  }
  return inicio;
}
```

5

Implementación

```
int main () {
  struct nodo *inicio = NULL;
  int dato, op;
  do {
    printf("\n 1. Insertar en la lista");
    printf("\n 2. Eliminar un elemento ");
    printf("\n 3. Imprimir la lista");
    printf("\n 4. Salir");
    printf("\n Opcion: [ ]\b\b");
    scanf(" %d", &op);
  }
```

```
switch (op) {
  case 1:
    printf("\nDato:");
    scanf(" %d", &dato);
    if (inicio == NULL)
      inicio = crearnodo(dato);
    else
      inicio = insertar(inicio, dato);
    break;
  case 2:
    if(inicio != NULL){
      printf("\nEscribe dato a borrar: ");
      scanf(" %d", &dato);
      inicio = eliminar(inicio, dato);
    }
    else
      printf("Lista vacia \n");
    break;
  case 3:
    mostrar(inicio);
    break;
}
}while (op != 4);
}
```

6