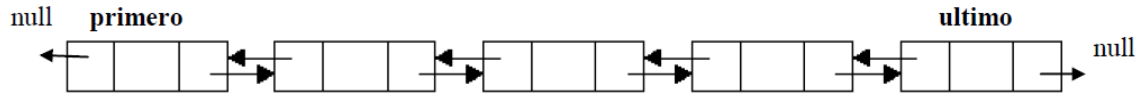


# Listas de enlace doble

estructura de una lista de enlace doble;



## CLASE NODO

```
public class Nodo {
    int info;
    Nodo siguiente;
    Nodo anterior;
    public Nodo (int a) {
        info=a;
        siguiente=null;
        anterior=null;
    }
}
```

## CLASE NODO

```
public class LDoble {
    Nodo primero, ultimo;
    void insertar (int a) {
        if (primero==null) {
            primero=new Nodo(a);
            ultimo=primero;
        }
        else {
            Nodo temp=new Nodo(a);
            ultimo.siguiente=temp;
            temp.anterior=ultimo;
            ultimo=temp;
        }
    }
    void mostrar() {
        Nodo aux=primero;
        while (aux!=null) {
            System.out.print(aux.info+" ");
            aux=aux.siguiente;
        }
    }
    void MostrarAtras() {
        Nodo aux=ultimo;
        while (aux!=null) {
            System.out.print(aux.info+" ");
            aux=aux.anterior;
        }
    }
}
```

```

void MostrarDatosNoRepetidos() {
    if(primerono==null) {
        Nodo aux1=ultimo, aux2=ultimo;
        while(aux1!=null) {
            aux2=ultimo;
            while(aux2!=null) {
                if(aux1.info==aux2.info&&aux1!=aux2)
                    break;
                aux2=aux2.anterior;
            }
            if(aux2==null)
                System.out.print(aux1.info+" ");
            aux1=aux1.anterior;
        }
    }
}

void eliminar() {
    if(primerono==null) {
        Nodo aux1=ultimo, aux2=ultimo;
        while(aux1!=null) {
            aux2=ultimo;
            while(aux2!=null) {
                if(aux1.info==aux2.info&&aux1!=aux2)
                    break;
                aux2=aux2.anterior;
            }
            if(aux2==null) {
                if(aux1==primero) {
                    primero=primero.siguiente;
                    aux1.anterior=null;
                }
                else {
                    if(aux1==ultimo) {
                        ultimo=ultimo.anterior;
                        ultimo.siguiente=null;
                    }
                    else {
                        Nodo tem=aux1.anterior;
                        tem.siguiente=aux1.siguiente;
                        aux1=aux1.siguiente;
                        aux1.anterior=tem;
                    }
                }
            }
            aux1=aux1.anterior;
        }
    }
}

int suma() {
    int sum=0;
    if(primerono==null) {
        Nodo tem=primero;
        while(tem!=null) {
            sum+=tem.info;
            tem=tem.siguiente;
        }
    }
    return sum;
}

```

```
}
```

## CLASE DE PRUEBA

```
import java.util.Scanner;

public class Prueba {
    public static void main(String[] args) {
        LDoble l=new LDoble();
        Scanner cs=new Scanner(System.in);
        int n=cs.nextInt();
        while(n!=0){
            n--;
            //insercion de numeros aleatorios del 1 al 30
            l.insertar(1+(int) (Math.random()*30));
        }
        l.mostrar();
        System.out.println();
        l.MostrarAtras();
        System.out.println();
        l.MostrarDatosNoRepetidos();
        l.eliminar();
        System.out.println();
        l.mostrar();
        System.out.println();
        System.out.print("la sumatoria es: "+l.suma());
    }
}
```

El tratamiento de esta clase de listas de doble enlace es el mismo que con los de simple enlace, simplemente no hay que olvidar que el manejo de los 2 enlaces en momentos de eliminar e insertar. Además de hacer el trabajo más sencillo en otras operaciones (en el sentido de recorrer una lista).