

RESOLUCIÓN
PRACTICA NRO. 9
LISTA ENLAZADA SIMPLE

Clase Nodo:

```
public class Nodo {  
    int info;  
    Nodo enlace;  
    public Nodo(int dato) {  
        info=dato;  
        enlace=null;  
    }  
}
```

1. Desarrolle una función que identifique y muestre todos los números pares contenidos en una lista enlazada simple.

Clase Lista

```
public class Lista {  
    Nodo Inicio;  
    Nodo ultimo;  
    int tam;  
    public Lista(){  
        tam=0;  
    }  
  
    public void insertar(int dato){  
        Nodo aux=new Nodo(dato);  
        if(tam==0){  
            Inicio=aux;  
        }else{  
            ultimo.enlace=aux;  
        }  
        ultimo=aux;  
        tam++;  
    }  
    public void mostrar(){  
        Nodo aux=Inicio;  
        while(aux!=null){  
            System.out.println(aux.info);  
            aux=aux.enlace;  
        }  
    }  
  
    public void mostrarPares(){  
        Nodo aux=Inicio;  
        while(aux!=null){  
            if(aux.info%2==0)  
                System.out.println(aux.info);  
            aux=aux.enlace;  
        }  
    }  
}
```

```
}
```

Clase Main

```
public static void main(String[] args) {  
    // TODO code application logic here  
    Lista nuevo=new Lista();  
    nuevo.insertar(5);  
    nuevo.insertar(3);  
    nuevo.insertar(8);  
    nuevo.insertar(1);  
    nuevo.insertar(4);  
  
    nuevo.mostrarPares();  
}
```

2. Escribe una función que devuelva la cantidad de elementos pares en una lista enlazada simple.

Clase Lista

```
public int ContPares(){  
    int n=0;  
    Nodo aux=Inicio;  
    while(aux!=null){  
        if(aux.info%2==0)  
            n++;  
        aux=aux.enlace;  
    }  
    return n;  
}
```

Clase Main

```
public static void main(String[] args) {  
    // TODO code application logic here  
    Lista nuevo=new Lista();  
    nuevo.insertar(5);  
    nuevo.insertar(3);  
    nuevo.insertar(8);  
    nuevo.insertar(1);  
    nuevo.insertar(4);  
  
    nuevo.mostrarPares();  
    System.out.println(nuevo.ContPares());  
}
```

3. Escribe una función que tome un número como parámetro y devuelva la posición de ese número en la lista enlazada. Si hay más de una ocurrencia, devuelve la posición de la primera. Si el número no está en la lista, la función devuelve -1.

Clase Lista

```
public class Lista {  
    Nodo Inicio;  
    Nodo ultimo;  
    int tam;  
    public Lista(){  
        tam=0;
```

```

}

public void insertar(int dato){
    Nodo aux=new Nodo(dato);
    if(tam==0){
        Inicio=aux;
    }else{
        ultimo.enlace=aux;
    }
    ultimo=aux;
    tam++;
}

public void mostrar(){
    Nodo aux=Inicio;
    while(aux!=null){
        System.out.println(aux.info);
        aux=aux.enlace;
    }
}

public void mostrarPares(){
    Nodo aux=Inicio;
    while(aux!=null){
        if(aux.info%2==0)
            System.out.println(aux.info);
        aux=aux.enlace;
    }
}

public int ContPares(){
    int n=0;
    Nodo aux=Inicio;
    while(aux!=null){
        if(aux.info%2==0)
            n++;
        aux=aux.enlace;
    }
    return n;
}

public int Buscar(int num){
    int pos=0;
    boolean en=false;
    Nodo aux=Inicio;
    while(aux!=null){
        if(aux.info==num){
            en=true;
            break;
        }
        aux=aux.enlace;
        pos++;
    }
    if(en==false)
        return -1;
}

```

```

        return pos;
    }
}

```

Clase Main

```

public static void main(String[] args) {
    // TODO code application logic here
    Lista nuevo=new Lista();
    nuevo.insertar(5);
    nuevo.insertar(3);
    nuevo.insertar(8);
    nuevo.insertar(1);
    nuevo.insertar(4);

    nuevo.mostrarPares();
    System.out.println(nuevo.ContPares());
    System.out.println(nuevo.Buscar(7));
}

```

4. Desarrolle una función que elimine un elemento específico, representado por 'N', de la lista.

Clase Lista

```

public class Lista {
    Nodo Inicio;
    Nodo ultimo;
    int tam;
    public Lista(){
        tam=0;
    }

    public void insertar(int dato){
        Nodo aux=new Nodo(dato);
        if(tam==0){
            Inicio=aux;
        }else{
            ultimo.enlace=aux;
        }
        ultimo=aux;
        tam++;
    }
    public void mostrar(){
        Nodo aux=Inicio;
        while(aux!=null){
            System.out.println(aux.info);
            aux=aux.enlace;
        }
    }

    public void mostrarPares(){
        Nodo aux=Inicio;
        while(aux!=null){
            if(aux.info%2==0)
                System.out.println(aux.info);
            aux=aux.enlace;
        }
    }
}

```

```

    }
}
public int ContPares(){
    int n=0;
    Nodo aux=Inicio;
    while(aux!=null){
        if(aux.info%2==0)
            n++;
        aux=aux.enlace;
    }
    return n;
}
public int Buscar(int num){
    int pos=0;
    boolean en=false;
    Nodo aux=Inicio;
    while(aux!=null){
        if(aux.info==num){
            en=true;
            break;
        }
        aux=aux.enlace;
        pos++;
    }
    if(en==false)
        return -1;
    return pos;
}
public void eliminar(int num){
    Nodo aux=Inicio;
    Nodo aux2=aux.enlace;
    if(aux.info==num){
        Inicio=aux2;
        return;
    }
    while(aux2!=null){
        if(aux2.info==num){
            aux.enlace=aux2.enlace;
            aux2.enlace=null;
            return;
        }
        aux=aux2;
        aux2=aux.enlace;
    }
}
}

```

Clase Main

```

public static void main(String[] args) {
    // TODO code application logic here
    Lista nuevo=new Lista();
    nuevo.insertar(5);
    nuevo.insertar(3);
}

```

```
nuevo.insertar(8);
nuevo.insertar(1);
nuevo.insertar(4);

nuevo.mostrar();

nuevo.eliminar(8);
System.out.println("Eliminando el 8");
nuevo.mostrar();
}
```