

Miguel Lozano Arvizu

Implementación de una Lista Simplemente Enlazada
con Nodos en Java.

Clase Nodo
public class Nodo {
private int dato;
private Nodo siguiente;

public Nodo (int dato) {
this.dato = dato;
this.siguiente = null;

public int getDato () {
return dato;

}

public void setDato () {
this.dato = dato;

}

public Nodo getSiguiente () {
return siguiente;

}

public void setSiguiente () {
this.siguiente = siguiente;

}

}

falla primero

Clase ListaEnlazada

```
public class ListaEnlazada {  
    Nodo cabeza; // Primer nodo - la cabeza  
    // Constructor
```

```
public ListaEnlazada() {  
    this.cabeza = null;  
}  
// Metodos de las operaciones.
```

- Corregido - Metodos -

```
public class ListaEnlazada <T> {  
    private Nodo <T> cabeza; // Encapsulamiento
```

```
public ListaEnlazada() { // Constructor  
    this.cabeza = null;  
}
```

```
public Nodo <T> getCabeza() {  
    return cabeza;  
}
```

```
// Metodos de las operaciones.  
// Agregar
```

```
public void agregar (T dato) {  
    Nodo <T> nuevo = new Nodo <> (dato);  
    nuevo.siguiente = cabeza;  
    cabeza = nuevo;  
}
```


// Eliminar

```
public void eliminar (T dato) {  
    if (cabeza == null) {  
        return;  
    }  
    if (cabeza.data.equals(dato)) {  
        cabeza = cabeza.siguiiente;  
        return;  
    }  
}
```

```
    Nodo <T> actual = cabeza;  
    while (actual.siguiiente != null) {  
        if (actual.siguiiente == actual.siguiiente.siguiiente;  
        return;  
    }  
    actual = actual.siguiiente;  
}  
// Imprimir
```

```
public void imprimir () {  
    Nodo <T> actual = cabeza;  
    while (actual != null) {  
        System.out.println (actual.data);  
        actual = actual.siguiiente;  
    }  
}
```

1/ Buscar dato

```
public boolean buscar (T dato) {  
    Nodo<T> actual = cabeza;  
    while (actual != null) {  
        if (actual.dato.equals(dato)) {  
            return true;  
        }  
        actual = actual.siguiente;  
    }  
    return false;  
}
```

3 - Dibujo

3.1 - Agregar al inicio (agregarAlInicio (int dato)).

```
public void agregarAlInicio (int dato) {  
    Nodo nuevo = new Nodo (dato);  
    nuevo.setsiguiente (cabeza);  
    cabeza = nuevo;  
}
```

3.2 - Eliminar al Inicio (eliminarAlInicio()).

```
public void eliminarInicio () {  
    if (cabeza != null) {  
        cabeza = cabeza.getsiguiente ();  
    }  
}
```

3.3 Recorrer / Imprimir (imprimirLista ())

```
public void imprimirLista () {  
    Nodo actual = cabeza;  
    while (actual != null) {  
        System.out.println (actual.getData () + " ");  
        actual = actual.getsiguiente ();  
    }  
    System.out.println (" null");  
}
```


4.- Caso Practico: Gestion de un Historial de Tracciones

```
public class Transacciones {  
    public static void main (String[] args) {  
        // Inicializar lista  
        ListaEnlazada lista = new ListaEnlazada();  
  
        lista.agregarInicio (150);  
        lista.agregarInicio (45);  
        lista.agregarInicio (200);  
        lista.agregarInicio (75);  
  
        System.out.println ("Lista Actual");  
        lista.imprimir ();  
  
        lista.eliminarInicio ();  
  
        System.out.println ("\nDespues de la cancelacion");  
        lista.imprimir ();  
    }  
}
```

- Desafio

```
public void agregar final (int dato) {  
    Nodo nuevo = new Nodo (dato);  
    if  
        (cabeza == null) {  
            cabeza = nuevo;  
            return;  
        }  
}
```

Higuel Lozano Arvizu GTID 141

Scribe

51

```
Nodo actual = cabeza;  
while (actual.getsiguiente() != null) {  
    actual.setsiguiente(nuevo);  
}
```