



Universidad Nacional Autónoma de México

Facultad de Estudios Superiores Aragón

Estructura de Datos

Tarea 4

Laguna Velasco Elizabeth

Grupo: 1360

Fecha de entrega: 03 de septiembre de 2024

run:

Tarea 4

La lista de SmartPhones es:

```
[ marca: Apple, modelo: iPhone 15 Pro, sistemaOp: iOS 17, color: Titanio blanco] -->
[ marca: Samsung, modelo: Galaxy S23 Ultra, sistemaOp: Android 13, color: Menta] -->
[ marca: Google, modelo: Pixel 7 Pro, sistemaOp: Android 13, color: Obsidiana] -->
[ marca: OnePlus, modelo: OnePlus 11, sistemaOp: Android 13, color: Verde eterno] -->
[ marca: Xiaomi, modelo: Xiaomi 13 Pro, sistemaOp: Android 13, color: Negro ceramico] -->
```

Lista despues de eliminar el elemento en la posicion 2:

```
[ marca: Apple, modelo: iPhone 15 Pro, sistemaOp: iOS 17, color: Titanio blanco] -->
[ marca: Samsung, modelo: Galaxy S23 Ultra, sistemaOp: Android 13, color: Menta] -->
[ marca: OnePlus, modelo: OnePlus 11, sistemaOp: Android 13, color: Verde eterno] -->
[ marca: Xiaomi, modelo: Xiaomi 13 Pro, sistemaOp: Android 13, color: Negro ceramico] -->
```

Lista despues con un nuevo elemento al inicio y al final:

```
[ marca: Huawei, modelo: Mate 50 Pro, sistemaOp: HarmonyOS 3, color: Plata] -->
[ marca: Apple, modelo: iPhone 15 Pro, sistemaOp: iOS 17, color: Titanio blanco] -->
[ marca: Samsung, modelo: Galaxy S23 Ultra, sistemaOp: Android 13, color: Menta] -->
[ marca: OnePlus, modelo: OnePlus 11, sistemaOp: Android 13, color: Verde eterno] -->
[ marca: Xiaomi, modelo: Xiaomi 13 Pro, sistemaOp: Android 13, color: Negro ceramico] -->
[ marca: Sony, modelo: Xperia 1 IV, sistemaOp: Android 12, color: Negro] -->
```

Lista despues de eliminar el primer elemento:

```
[ marca: Apple, modelo: iPhone 15 Pro, sistemaOp: iOS 17, color: Titanio blanco] -->
[ marca: Samsung, modelo: Galaxy S23 Ultra, sistemaOp: Android 13, color: Menta] -->
[ marca: OnePlus, modelo: OnePlus 11, sistemaOp: Android 13, color: Verde eterno] -->
[ marca: Xiaomi, modelo: Xiaomi 13 Pro, sistemaOp: Android 13, color: Negro ceramico] -->
[ marca: Sony, modelo: Xperia 1 IV, sistemaOp: Android 12, color: Negro] -->
```

BUILD SUCCESSFUL (total time: 0 seconds)

```
package tarea4main;

public class Tarea4Main {
    public static void main(String[] args) {
        System.out.println("Tarea 4");
        System.out.println("-----");

        SmartPhone smartphone1 = new SmartPhone("Apple", "iPhone 15 Pro", "iOS 17", "Titanio blanco");
        ListaLigadaSimple<SmartPhone> smartphones = new ListaLigadaSimple<>();

        smartphones.agregarAlFinal(smartphone1);
        smartphones.agregarAlFinal(new SmartPhone("Samsung", "Galaxy S23 Ultra", "Android 13", "Menta"));
        smartphones.agregarAlFinal(new SmartPhone("Google", "Pixel 7 Pro", "Android 13", "Obsidiana"));
        smartphones.agregarAlFinal(new SmartPhone("OnePlus", "OnePlus 11", "Android 13", "Verde eterno"));
        smartphones.agregarAlFinal(new SmartPhone("Xiaomi", "Xiaomi 13 Pro", "Android 13", "Negro ceramico"));

        //Para imprimir la lista
        System.out.println("La lista de SmartPhones es: ");
        System.out.println();
        smartphones.transversal();

        //Para eliminar el de la posicion 2
        System.out.println("-----");
        System.out.println("Lista despues de eliminar el elemento en la posicion 2: ");
        System.out.println();
        smartphones.eliminarElemento(2);
        smartphones.transversal();
        System.out.println("-----");

        /*Para actualizar el segundo elemento
        SmartPhone smartphone2 = new SmartPhone("Oppo", "Reno7", "Android 12", "Azul Aurora");
        */
    }
}
```

```
//Para agregar un elemento al inicio y otro al final
smartphones.agregarAlInicio(new SmartPhone("Huawei", "Mate 50 Pro", "HarmonyOS 3", "Plata"));
smartphones.agregarAlFinal(new SmartPhone("Sony", "Xperia 1 IV", "Android 12", "Negro"));
System.out.println("Lista despues con un nuevo elemento al inicio y al final: ");
System.out.println();
smartphones.transversal();
System.out.println("-----");

//Para eliminar el primero
smartphones.eliminarPrimero();
System.out.println("Lista despues de eliminar el primer elemento: ");
smartphones.transversal();
System.out.println("-----");
```

```
}
```

```
package tarea4main;
```

```
public class ListaLigadaSimple<T> {
```

```
    private Nodo<T> head;
```

```
    private int tamaño;
```

```
    public ListaLigadaSimple() {
```

```
    }
```

```
    //Método para saber si la lista está vacía
```

```
    public boolean estaVacía() {
```

```
        boolean resultado = false;
```

```
        if(this.head == null) {
```

```
            resultado = true;
```

```
        }
```

```
        return resultado;
```

```
    }
```

```
    //Método para saber el número de elementos de la lista
```

```
    public int getTamaño() {
```

```
        Nodo<T> aux = this.head;
```

```
        int contador = 0;
```

```
        while(aux != null) {
```

```
            contador ++;
```

```
            aux = aux.getSiguiente();
```

```
        }
```

```
        return contador;
```

```
    }
```

```
    //Método para agregar datos al final
```

```
    public void agregarAlFinal(T dato) {
```

```
        Nodo<T> nuevo = new Nodo<>(dato);
```

```
        if(head == null) {
```

```
            head = nuevo;
```

```
        }else{
```

```
            Nodo<T> aux = this.head;
```

```
            while(aux.getSiguiente() != null) {
```

```

//Método para agregar datos al inicio
public void agregarAlInicio(T dato){
    this.head = new Nodo<>(dato, this.head);
}

//Método para agregar después de un nodo de referencia
public void agregarDespuesDe(T referencia, T dato){
    Nodo<T> auxiliar = this.head;

    while(auxiliar != null && !auxiliar.getDato().equals(referencia)){
        auxiliar = auxiliar.getSiguiente();
    }
    if(auxiliar != null){
        auxiliar.setSiguiente(new Nodo<T>(dato, auxiliar.getSiguiente()));
    }else{
        System.out.println("No se encontro el nodo de referencia");
    }
}

//Método para eliminar el elemento de cierta posición
public void eliminarElemento(Integer posicion){
    if(posicion < 0 || posicion >= getTamanio()){
        System.out.println("Posición inválida");
    }

    if(posicion == 0){
        this.head = this.head.getSiguiente();
    }else{
        Nodo<T> aux = this.head;
        for (int i = 0; i < posicion - 1; i++) {
            aux = aux.getSiguiente();
        }
        Nodo<T> nodoAEliminar = aux.getSiguiente();
        aux.setSiguiente(nodoAEliminar.getSiguiente());
    }
    tamanio--;
}

```

```
//Método para eliminar el último de la lista
```

```
public void eliminarUltimo() {  
    Nodo<T> auxiliar = this.head;  
    if (getTamano() < 1) {  
        while (auxiliar.getSiguiente().getSiguiente() != null) {  
            auxiliar = auxiliar.getSiguiente();  
        }  
        auxiliar.setSiguiente(null);  
    } else {  
        this.head = null;  
    }  
}
```

```
//Método para buscar un elemento en la lista y regresar la posición en la que se encuentra
```

```
public int buscar(T valor) {  
    if (estaVacia() == true) {  
        return -1;  
    }  
    Nodo<T> auxiliar = this.head;  
    int contador = 0;  
    while (auxiliar.getDato() != valor) {  
        auxiliar = auxiliar.getSiguiente();  
        contador++;  
    }  
    return contador;  
}
```

```
//Método para actualizar un valor
```

```
public void actualizar(T aBuscar, T dato) {  
    Nodo<T> auxiliar = this.head;  
    while (auxiliar.getDato() != aBuscar) {  
        auxiliar = auxiliar.getSiguiente();  
    }  
    auxiliar.setDato(dato);  
}
```

```
//Método para hacer un recorrido transversal y mostrar todos los elementos
public void transversal(){
    Nodo<T> aux = this.head;
    while(aux != null){
        System.out.println(aux.getDato() + " --> ");
        aux = aux.getSiguiente();
    }
    System.out.println("");
}
```



```

package tarea4main;

public class Nodo<T>{
    private T dato;
    private Nodo<T> siguiente;

    public Nodo() {
    }

    public Nodo(T dato){
        this.dato = dato;
    }

    public Nodo(T dato, Nodo<T> siguiente){
        this.dato = dato;
        this.siguiente = siguiente;
    }

    public T getDato() {
        return dato;
    }

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

    public Nodo<T> getSiguiente() {
        return siguiente;
    }

    public void setSiguiente(Nodo<T> siguiente){
        this.siguiente = siguiente;
    }

    @Override
    public String toString() {
        return "Nodo{" +
            "dato=" + dato +

```

```
package tarea4main;

public class SmartPhone {
    private String marca;
    private String modelo;
    private String sistemaOperativo;
    private String color;

    public SmartPhone() {
    }

    public SmartPhone(String marca, String modelo, String sistemaOperativo, String color){
        this.marca = marca;
        this.modelo = modelo;
        this.sistemaOperativo = sistemaOperativo;
        this.color = color;
    }

    public String getMarca(){
        return marca;
    }

    public void setMarca(String marca){
        this.marca = marca;
    }

    public String getModelo(){
        return modelo;
    }

    public void setModelo(String modelo){
        this.modelo = modelo;
    }

    public String getSistemaOperativo(){
        return sistemaOperativo;
    }
}
```

```
public String getColor() {  
    return color;  
}
```

```
public void setColor(String color) {  
    this.color = color;  
}
```

@Override

```
public String toString() {  
    return "[ "+  
        "marca: " + marca +  
        ", modelo: " + modelo +  
        ", sistemaOp: " + sistemaOperativo +  
        ", color: " + color +  
        ']' ;  
}
```