



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

ALUMNO:

PAZ MALDONADO CARLOS SAÚL

NOMBRE DEL PROFESOR:

HERNANDEZ CABRERA JESUS

NOMBRE DE LA MATERIA:

ESTRUCTURA DE DATOS

FECHA DE ENTREGA:

3 de septiembre del 2024

TAREA NO. 4

CAPTURAS DEL CÓDIGO

NOTAS:


Se habilitaron las características de vista previa para poder utilizar las “Template Strings” ya que es más fácil de concatenar las variables en un String.

Project
Default settings for all modules. Configure these parameters for each module on the module page as needed.

Name:

ListaLigada

SDK:

 22.0.2 Oracle OpenJDK 22.0.2

▼

Edit


Language level:

22 (Preview) - Statements before super(), string templates (2nd preview) etc.

▼

Compiler output:

; Aragón\Semestre 2025-I\Estructura de datos\Tareas programación\ListaLigada\out



Used for module subdirectories, Production and Test directories for the corresponding sources.

CLASE ListaLigada

```
package unam.fesaragon.estructuradatos;

public class ListaLigada<T> {
    private T dato;
    private int tamaño;
    private Nodo<T> Siguiente;
    private Nodo<T> head;

    // Constructores
    public ListaLigada() {
    }

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

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

    // Comprobar si está vacía
    public boolean estaVacía() {
        return this.tamaño == 0;
    }

    // Agregar al final de la lista
    public void agregarAlFinal(T dato) {
        if (!estaVacía()) {
            Nodo<T> aux = head;
            while (aux.getSiguiente() != null) {
                aux = aux.getSiguiente();
            }
            aux.setSiguiente(new Nodo<>(dato));
        } else {
```

```

        head = new Nodo<>(dato);
    }
    this.tamanio++;
}

// Agregar al inicio de la lista
public void agregarAlInicio(T dato) {
    if (!estaVacia()) {
        Nodo<T> nuevo = new Nodo<>(dato, this.head);
        this.head = nuevo;
    } else {
        head = new Nodo<>(dato);
    }
    this.tamanio++;
}

// Agregar después de
public void agregarDespuesDe(T referencia, T dato) {
    if (estaVacia()) {
        agregarAlInicio(dato);
    } else {
        Nodo<T> aux = head;
        while (aux != null &&
!aux.getDato().equals(referencia)) {
            aux = aux.getSiguiente();
        }
        if (aux != null) {
            aux.setSiguiente(new Nodo<>(dato,
aux.getSiguiente()));
        }
        this.tamanio++;
    }
}

// Metodo para eliminar por posición
public void eliminar(int posicion) {
    if (estaVacia()) {
        System.out.println("La lista esta vacia");
        return;
    }
    if (posicion > this.tamanio || posicion < 1) {
        System.out.println("La posición que quieres
eliminar está fuera de rango");
        return;
    }

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

```

```

aux.setSiguiente(aux.getSiguiente().getSiguiente());
    }
    this.tamanio--;
}

//Metodo para eliminar el primero
public void eliminarElPrimero() {
    if (estaVacia()) {
        System.out.println("La lista esta vacia");
    } else {
        head = head.getSiguiente();
        tamanio--;
    }
}

//Metodo para eliminar el ultimo elemento
public void eliminarElFinal() {
    if (estaVacia()) {
        System.out.println("La lista esta vacia");
    } else if (head.getSiguiente() == null) {
        head = null;
        tamanio--;
    } else {
        Nodo<T> aux = head;
        while (aux.getSiguiente().getSiguiente() !=
null) {
            aux = aux.getSiguiente();
        }
        aux.setSiguiente(null);
        tamanio--;
    }
}

public int buscar(T valor) {
    if (estaVacia()) {
        System.out.println(";La lista está vacía!");
        return 0;
    } else {
        Nodo<T> aux = head;
        int posicion = 1;
        while (aux != null) {
            if (aux.getDato().equals(valor)) {
                return posicion;
            }
            aux = aux.getSiguiente();
            posicion++;
        }
        return 0;
    }
}

public void actualizar(T a_buscar, T valor) {
    if (estaVacia()) {

```

```

        System.out.println("La lista está vacía");
    } else {
        Nodo<T> aux = head;
        while (aux != null) {
            if (aux.getDato().equals(a_buscar)) {
                aux.setDato(valor);
                return;
            }
            aux = aux.getSiguiente();
        }
        System.out.println("No se encuentra en la lista
el elemmento que intentas buscxar");
    }
}

// Metodo para imprimir la lista
public void transversal() {
    if (estaVacía()) {
        System.out.println("Esta vacía la lista");
    } else {
        Nodo<T> aux = head;
        while (aux != null) {
            System.out.print("|" + aux.getDato() + "|" -
-> ");
            aux = aux.getSiguiente();
        }
        System.out.println();
    }
}

// Getters y setters
public void setSiguiente(Nodo<T> siguiente) {
    Siguiente = siguiente;
}

public T getDato() {
    return dato;
}

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

public int getTamanio() {
    return tamanio;
}

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

```

Clase Nodo

```
package unam.fesaragon.estructuradatos;

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;
    }

    public String toString() {
        return STR."Nodo{dato=\{dato},
siguiente=\{siguiente}\{'\}'"};
    }
}
```

Clase SmartPhone

```
package unam.fesaragon.estructuradatos;

public class SmartPhone {
    private int modelo;
    private String marca;
    private int versionDeModelo;
    private String sistemaOperativo;
```

```

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

    @Override
    public String toString() {
        return STR."\{this.marca} \{this.versionDeModelo}
\{this.modelo}";
    }

    public int getModelo() {
        return modelo;
    }

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

    public String getMarca() {
        return marca;
    }

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

    public int getVersionDeModelo() {
        return versionDeModelo;
    }

    public void setVersionDeModelo(int versionDeModelo) {
        this.versionDeModelo = versionDeModelo;
    }

    public String getSistemaOperativo() {
        return sistemaOperativo;
    }

    public void setSistemaOperativo(String
sistemaOperativo) {
        this.sistemaOperativo = sistemaOperativo;
    }
}

```

Clase MainListaLigada

```

package unam.fesaragon.estructuradatos;

public class MainListaLigada {
    public static void main(String[] args) {
        ListaLigada<SmartPhone> celulares = new
ListaLigada<>();
        SmartPhone cel1 = new SmartPhone(2021, "Apple", 13,
"iOS");
        SmartPhone cel2 = new SmartPhone(2024, "Samsung",
21, "Android");
        SmartPhone cel3 = new SmartPhone(2017, "Google", 6,
"Android");
        SmartPhone cel4 = new SmartPhone(2020, "OnePlus",
9, "Android");
        SmartPhone cel5 = new SmartPhone(2019, "Sony", 3,
"Android");
        // Agregar 5 SmartPhones
        celulares.agregarAlFinal(cel1);
        celulares.agregarAlFinal(cel2);
        celulares.agregarAlFinal(cel3);
        celulares.agregarAlFinal(cel4);
        celulares.agregarAlFinal(cel5);

        // Imprimir el contenido
        System.out.println("Contenido de la lista:");
        celulares.transversal();

        // Eliminar el de la posición 2
        celulares.eliminar(2);

        // Volver a imprimir el contenido
        System.out.println();
        System.out.println("Lista despues de eliminar el
elemento 2:");
        celulares.transversal();

        // Actualizar el segundo elemento
        celulares.actualizar(cel3, new SmartPhone(2022,
"Google", 7, "Android"));

        // Volver a imprimir el contenido
        System.out.println();
        System.out.println("Lista despues de actualizar el
elemento 2:");
        celulares.transversal();

        // Agregar un elemento al inicio y otro diferente
al final
        celulares.agregarAlInicio(new SmartPhone(2024,
"Huawei", 50, "HarmonyOS"));
        celulares.agregarAlFinal(new SmartPhone(2018,
"Xiaomi", 11, "Android"));
    }
}

```



```
        // Volver a imprimir el contenido
        System.out.println();
        System.out.println("Agregando al principio y al
final:");
        celulares.transversal();

        // Eliminar el primero
        celulares.eliminarElPrimero();

        // Volver a imprimir el contenido
        System.out.println();
        System.out.println("Eliminando el primer
elemento:");
        celulares.transversal();
    }
}
```

CAPTURAS DE EJECUCIÓN

```
C:\Program Files\Java\jdk-22.0.2\bin\java.exe" --enable-preview "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2\lib\idea_rt.jar=61481:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2\b
Contenido de la lista:
|Apple 13 2021| --> |Samsung 21 2024| --> |Google 6 2017| --> |OnePlus 9 2020| --> |Sony 3 2019| -->

Lista despues de eliminar el elemento 2:
|Apple 13 2021| --> |Google 6 2017| --> |OnePlus 9 2020| --> |Sony 3 2019| -->

Lista despues de actualizar el elemento 2:
|Apple 13 2021| --> |Google 7 2022| --> |OnePlus 9 2020| --> |Sony 3 2019| -->

Agregando al principio y al final:
|Huawei 50 2024| --> |Apple 13 2021| --> |Google 7 2022| --> |OnePlus 9 2020| --> |Sony 3 2019| --> |Xiaomi 11 2018| -->

Eliminando el primer elemento:
|Apple 13 2021| --> |Google 7 2022| --> |OnePlus 9 2020| --> |Sony 3 2019| --> |Xiaomi 11 2018| -->

Process finished with exit code 0
```