

Universidad Nacional Autónoma de México



Facultad de Estudios Superiores Aragón

Ingeniería en Computación

Estructura de Datos

Jesús Hernández Cabrera

Axel Yahir Moreno Rodríguez



Turno Vespertino Grupo 1360

```
File Edit Selection View Go Run ··
                                                                                                                                                              Q Untitled (Workspace)
                                                                                                                                                                                                                                                                                      O
          V UNTITLED (WORKSPACE)
 Q

✓ Tarea 2

                                                                                 3 public class ListaADTMain {
 လူ
              J ConjuntoADT.java
                                                                                                        ListaLigada<SmartPhone> smartphones = new ListaLigada<>();
              (i) README.md
                                                                                                       SmartPhone smartphone1 = new SmartPhone(marca:"Huawei",ano:2018,modelo:"A2");
SmartPhone smartphone2 = new SmartPhone(marca:"Motorola",ano:2016,modelo:"Moto 6");
SmartPhone smartphone3 = new SmartPhone(marca:"Smasung",ano:2018,modelo:"A52"]);
SmartPhone smartphone4 = new SmartPhone(marca:"Phone",ano:2018,modelo:"15");
              Tarea 2.docx
                 Tarea 2.pdf
 €€
                                                                                       . .
             Estructura de DatosMaterial
                                                                                                        SmartPhone smartphone5 = new SmartPhone(marca: "Honor", ano: 2018, modelo: "X3");
                                                                                                        smartphones.agregarAlFinal(smartphone1);
smartphones.agregarAlFinal(smartphone2);
å
                                                                                                       smartphones.agregarAlFinal(smartphone3);
smartphones.agregarAlFinal(smartphone4);
smartphones.agregarAlFinal(smartphone5);
                 > Tarea 2
                  J Lista ADTMain.java
                                                                                                       smartphones.transversal();
System.out.println("");
                                                                                                       smartphones.eliminar(pos:2);
smartphones.transversal();
System.out.println("");
                                                                                                        SmartPhone smartphonenvo = new SmartPhone(marca:"Alcatel",ano:2008, modelo:"M2");
         ∨ ß ListaADTMain
                                                                                                       smartphones.actualizar(pos:2, smartphonenvo);
smartphones.agregarAlInicio(new SmartPhone(marca:"Realme",ano:2020,modelo:"C2"));
smartphones.agregarAlFinal(new SmartPhone(marca:"Poco", ano:2023, modelo:"X6 Pro"
                 main(Stringff) : void
                                                                                                        smartphones.transversal():
         > TIMELINE
                                                                                                        smartphones.eliminarElPrimero();
                                                                                                        smartphones.transversal();
System.out.println("");
        > JAVA PROJECTS
          > MAVEN
У ⊗ 3 🛆 9 🕍 0 🕁 📛 Java: Ready
                                                                                                                                                                                                                                                                   Ln 10. Col 49 Spaces: 4 UTF-8 CRLF Java 😅
★ File Edit Selection View Go Run
                                                                                                                                                                                                                                                                                      Q Untitled (Workspace)
0
          V UNTITLED (WORKSPACE)
                                                                              public class ListaLigada<T> {
    public void agregarAlInicio(T dato){
 Q
 လွ
               J Main.java
                                                                                                public void agregarDespuesDe(int ref,T dato){
                                                                                                      Nodo(T) aux = head;
while (aux != null && !aux.getDato().equals(ref)) {
   aux = aux.getSiguiente();
             (i) README.md
 €
              ■ Tarea 2.zip
                                                                                                       if (aux != null) {
   Nodo<T> nuevo = new Nodo<>(dato);
   nuevo.setSiguiente(aux.getSiguiente());
                                                                                                                aux.setSiguiente(nuevo);
 å
                                                                                                       if (pos == 0) {
    eliminarElPrimero();
                  J ListaLigada.java
J Nodo.java
                                                                                                       } else {
  Nodo<T> aux = head;
  for (int i = 0; i < pos - 1; i++) {</pre>
          V OUTLINE

✓ Mathematical Value of the Value of th
                                                                                                                aux.setSiguiente(aux.getSiguiente().getSiguiente());
                  ul tamanio
                                                                                                               tamanio--
                 C ListaLigada()
                  estaVacia() : boolean
         > JAVA PROJECTS
                                                                                                        if (!estaVacia())
                                                                                                                head = head.getSiguiente();
         > MAVEN
 PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL PORTS
                                                                                                                                                                                                                                                    数 Run: ListaADTMain 十~ 口 歯 ··· へ
                                                                                                                                                                                                                                                                                                                                         Z
  PS C:\Users\axelx\Documents\FES\3er semestre\Estructura de Datos>
                                                                                                                                                                                                                                                                                                                                        嵏
  PS C:\Users\axelx\Documents\FES\3er semestre\Estructura de Datos> c:; cd 'c:\Users\axelx\Documents\FES\3er semestre\Estructura de Datos'; &
    'C:\Program Files\Java\jre-1.8\bin\java.exe' '-cp' 'C:\Users\axelx\AppData\Roaming\Code\User\workspaceStorage\837d8bb75a804ad25f5399becbd65caa\redhat.java\jdt_ws\jdt.ls-java-project\bin' 'ListaADTMain'
  [Huawei, 2018, A2] --> [Motorola, 2016, Moto G] --> [Samsung, 2018, A52] --> [Phone, 2018, 15] --> [Honor, 2018, X3] --> null
  [Huawei, 2018, A2] --> [Motorola, 2016, Moto G] --> [Phone, 2018, 15] --> [Honor, 2018, X3] --> null
  [Realme, 2020, C2] --> [Huawei, 2018, A2] --> [Motorola, 2016, Moto G] --> [Alcatel, 2008, M2] --> [Honor, 2018, X3] --> [Poco, 2023, X6 Pro
    ] --> null
  [Huawei, 2018, A2] --> [Motorola, 2016, Moto G] --> [Alcatel, 2008, M2] --> [Honor, 2018, X3] --> [Poco, 2023, X6 Pro] --> null
  PS C:\Users\axelx\Documents\FES\3er semestre\Estructura de Datos> [
```

```
public class ListaLigada<T> {
    private Nodo<T> head;
    private int tamanio;
    public ListaLigada() {
    public boolean estaVacia(){
        boolean conj=false;
        if (this.head == null) {
            System.out.println("esta vacia");
            conj=true;
        return conj;
    public int getTamanio(){
        Nodo aux = this.head;
        int contador = 0;
        while (aux!=null){
            contador++;
            aux = aux.getSiguiente();
        return contador;
    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){
                aux = aux.getSiguiente();
            aux.setSiguiente(nuevo);
    public void agregarAlInicio(T dato){
        this.head = new Nodo<>(dato, this.head);
```

```
public void agregarDespuesDe(int ref,T dato){
   Nodo<T> aux = head;
    while (aux != null && !aux.getDato().equals(ref)) {
        aux = aux.getSiguiente();
    if (aux != null) {
        Nodo<T> nuevo = new Nodo<>(dato);
        nuevo.setSiguiente(aux.getSiguiente());
        aux.setSiguiente(nuevo);
        tamanio++;
public void eliminar(int pos){
   if (pos == 0) {
        eliminarElPrimero();
    } else {
        Nodo<T> aux = head;
        for (int i = 0; i < pos - 1; i++) {
            aux = aux.getSiguiente();
        aux.setSiguiente(aux.getSiguiente().getSiguiente());
        tamanio--;
public void eliminarElPrimero(){
    if (!estaVacia()) {
        head = head.getSiguiente();
       tamanio--;
public void eliminarElFinal(){
   if (!estaVacia()) {
        if (tamanio == 1) {
            head = null;
            Nodo<T> aux = head;
            while (aux.getSiguiente().getSiguiente() != null) {
                aux = aux.getSiguiente();
            aux.setSiguiente(null);
```

```
tamanio--;
public int buscar(T dato){
   Nodo<T> aux = head;
   int posicion = 0;
   while (aux != null) {
        if (aux.getDato().equals(dato)) {
            return posicion;
        aux = aux.getSiguiente();
        posicion++;
   return -1;
public void actualizar(int pos,T dato) {
    if (pos < 0 || pos>= getTamanio()) {
        System.out.println("Posición fuera de rango");
       return;
   Nodo<T> aux = head;
   for (int i = 0; i < pos; i++) {
        aux = aux.getSiguiente();
    aux.setDato(dato);
public void transversal(){
   Nodo<T> aux = this.head;
   while (aux != null){
        System.out.print(aux.getDato() + " --> ");
       aux = aux.getSiguiente();
   System.out.println("null");
```

```
public class SmartPhone {
   private String marca;
   private int ano;
```

```
private String modelo;
public SmartPhone() {
public SmartPhone(String marca, int ano, String modelo) {
    this.marca = marca;
    this.ano = ano;
    this.modelo = modelo;
public String getMarca(String marca) {
   return marca;
public void setMarca(String marca) {
   this.marca = marca;
public int getAno() {
   return ano;
public void setano(int ano) {
   this.ano = ano;
public String getModelo() {
   return modelo;
public void setModelo(String modelo) {
   this.modelo = modelo;
@Override
public String toString() {
    return "["+marca + ", " + ano +", " + modelo +"]";
```

```
public class ListaADTMain {
   public static void main(String[] args) {
```

```
ListaLigada<SmartPhone> smartphones = new ListaLigada<>();
SmartPhone smartphone1 = new SmartPhone("Huawei", 2018, "A2");
SmartPhone smartphone2 = new SmartPhone("Motorola",2016,"Moto G");
SmartPhone smartphone3 = new SmartPhone("Samsung",2018,"A52");
SmartPhone smartphone4 = new SmartPhone("Phone",2018,"15");
SmartPhone smartphone5 = new SmartPhone("Honor",2018,"X3");
smartphones.agregarAlFinal(smartphone1);
smartphones.agregarAlFinal(smartphone2);
smartphones.agregarAlFinal(smartphone3);
smartphones.agregarAlFinal(smartphone4);
smartphones.agregarAlFinal(smartphone5);
smartphones.transversal();
System.out.println("");
smartphones.eliminar(2);
smartphones.transversal();
System.out.println("");
SmartPhone smartphonenvo = new SmartPhone("Alcatel",2008, "M2");
smartphones.actualizar(2, smartphonenvo);
smartphones.agregarAlInicio(new SmartPhone("Realme",2020,"C2"));
smartphones.agregarAlFinal(new SmartPhone("Poco", 2023, "X6 Pro"));
smartphones.transversal();
System.out.println("");
smartphones.eliminarElPrimero();
smartphones.transversal();
System.out.println("");
```

```
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 +
            ", siguiente=" + siguiente +
```