Universidad Nacional Autónoma de México

Facultad de Estudios Superiores Aragón

Estructura de Datos

Tarea 5

Laguna Velasco Elizabeth

Grupo: 1360

Fecha de entrega: 08 de septiembre de 2024

```
run:
Tarea 5
----------------------------------------
<--| 50|-->
<--| 60|-->
<--| 65|-->
<--| 70|-->
<--| 80|-->
<--| 90|-->

<--| 50|-->
<--| 65|-->
<--| 70|-->
<--| 80|-->
<--| 90|-->

<--| 50|-->
<--| 65|-->
<--| 70|-->
<--| 88|-->
<--| 90|-->

No se encontro el valor
BUILD SUCCESSFUL (total time: 0 seconds)
```

```java
package tarea5main;

public class Tarea5Main {

    public static void main(String[] args) {
        System.out.println("Tarea 5");
        System.out.println("----------------------------------------");
        DoubleLinkedList<Integer> numeros = new DoubleLinkedList<>();
        numeros.agregarAlInicio(50);
        numeros.agregarAlFinal(60);
        numeros.agregarAlFinal(65);
        numeros.agregarAlFinal(70);
        numeros.agregarAlFinal(80);
        numeros.agregarAlFinal(90);
        numeros.transversal(0);
        numeros.eliminar(2);
        numeros.transversal(0);
        numeros.actualizarValorDePosicion(4, 88);
        numeros.transversal(0);
        numeros.buscarElemento(80);
    }

}
```

```java
package tarea5main;

public class DoubleLinkedList<T> {
    private NodoDoble<T> head;
    private NodoDoble<T> tail;
    private int tamanio;

    public DoubleLinkedList(){
    }

    //Metodo para saber si la lista esta vacia
    public boolean estaVacia(){
        boolean res = false;
        if(this.head == null && this.tail == null){
            res = true;
        }
        return res;
    }

    //Metodo para conocer el tamanio de la lista
    public int getTamanio(){
        return tamanio;
    }

    //Metodo para agregar un elemento al inicio de la lista
    public void agregarAlInicio(T valor){
        NodoDoble<T> nuevo = new NodoDoble<>(valor);
        if(this.estaVacia()){
            this.head = nuevo;
            this.tail = nuevo;
        } else{
            this.head.setAnterior(nuevo);
            nuevo.setSiguiente(this.head);
            this.head = nuevo;
```

```java
//Metodo para agregar un elemento al final de la lista
public void agregarAlFinal(T valor){
    NodoDoble<T> nuevo = new NodoDoble<>(valor);
    if(this.estaVacia()){
        this.head = nuevo;
        this.tail = nuevo;
    } else {
        this.tail.setSiguiente(nuevo);
        nuevo.setAnterior(this.tail);
        this.tail = nuevo;
    }
    this.tamanio++;
}

//Para mostrar la lista 0--> de izq a der , 1 --> de der a izq
public void transversal(int direccion){
    if(direccion == 1){
        NodoDoble<T> aux = this.tail;
        while(aux != null){
            System.out.println(aux);
            aux = aux.getAnterior();
        }
    } else {
        NodoDoble<T> aux = this.head;
        while(aux != null){
            System.out.println(aux);
            aux = aux.getSiguiente();
        }
    }
    System.out.println("");
}
```

```java
//Para agregar un elemento despues de otro de referencia
public void agregarDespuesDe(T referencia, T valor){
    NodoDoble<T> aux = this.head;

    while(aux.getData() != referencia){
        aux = aux.getSiguiente();
    }
    if(aux == null){
        System.out.println("No existe la referencia");
    }else{
        NodoDoble<T> nuevo = new NodoDoble<>(valor);
        nuevo.setSiguiente(aux.getSiguiente());
        nuevo.setAnterior(aux);
        if(aux.getSiguiente() != null){
            aux.getSiguiente().setAnterior(nuevo);
        }else{
            this.tail = nuevo;
        }
        aux.setSiguiente(nuevo);
        this.tamanio++;
    }
}

//Para obtener un elemento en la posicion solicitada
public T obtenerPosicion(int posicion){
    if(posicion < 0 || posicion >= this.tamanio){
        System.out.println("Posicion no disponible");
        return null;
    }
    NodoDoble<T> aux = this.head;
    for (int i = 0; i < posicion; i++) {
        aux = aux.getSiguiente();
    }
    return aux.getData();
}
```

```java
//Para eliminar el primer elemento de la lista
public void eliminarElPrimero(){
    if(this.estaVacia()){
        System.out.println("La lista esta vacia, no se puede eliminar elemento");
    }
    if(this.head == this.tail){
        this.head = null;
        this.tail = null;
    } else{
        this.head = this.head.getSiguiente();
        this.head.setAnterior(null);
    }
    this.tamanio--;
}

//Para eliminar el ultimo elemento de la lista
public void eliminarElFinal(){
    if(this.estaVacia()){
        System.out.println("La lista esta vacia, no se puede eliminar elemento");
    }
    if(this.head == this.tail){
        this.head = null;
        this.tail = null;
    } else {
        this.tail = this.tail.getAnterior();
        this.tail.setSiguiente(null);
    }
    this.tamanio--;
}
```

```java
//Para eliminar un elemento en cierta posicion
public void eliminar(int posicion){
    int pos = posicion - 1;
    if(pos < 0 || pos >= this.tamanio){
        System.out.println("Posicion no disponible");
    }
    if(pos == 0){
        eliminarElPrimero();
    }
    if(pos == this.tamanio -1){
        eliminarElFinal();
    }
    NodoDoble<T> aux = this.head;
    for (int i = 0; i < pos; i++) {
        aux = aux.getSiguiente();
    }
    aux.getAnterior().setSiguiente(aux.getSiguiente());
    aux.getSiguiente().setAnterior(aux.getAnterior());

    this.tamanio--;
}

//Para buscar un elemento en la lista y regresar la posicion en donde se encuentra
public int buscarElemento(T valor){
    NodoDoble<T> aux = this.head;
    int posicion = 0;

    while(aux != null){
        if(aux.getData() == valor){
            return posicion;
        }
        aux = aux.getSiguiente();
        posicion++;
    }
    System.out.println("No se encontro el valor");
```

```java
//Para actualizar un valor
public void actualizarValorDePosicion(int posicion, T valor){
    int pos = posicion -1;
    if(pos < 0 || pos >= this.tamanio){
        System.out.println("Posicion no disponible");
    }

    NodoDoble<T> aux = this.head;
    for (int i = 0; i < pos; i++) {
        aux = aux.getSiguiente();
    }
    aux.setData(valor);
}
```

```java
package tarea5main;

public class NodoDoble<T> {
    private T data;
    private NodoDoble<T> siguiente;
    private NodoDoble<T> anterior;

    public NodoDoble(){
    }

    public NodoDoble(T data){
        this.data = data;
    }

    public NodoDoble(T data, NodoDoble<T> siguiente, NodoDoble<T> anterior){
        this.data = data;
        this.siguiente = siguiente;
        this.anterior = anterior;
    }

    public T getData(){
        return data;
    }

    public void setData(T data){
        this.data = data;
    }

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

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