

Universidad Mariano Gálvez de Guatemala, Huehuetenango

Facultad de Ingeniería en Sistemas

Curso: Programación III

Docente: Gustavo Adolfo Maldonado Argueta



Estudiante: Madelyn Arleth Torres Gómez

Carné: 4490-22-7308

Huehuetenango 12 de abril de 2024

Listas enlazadas

```
package pckListaEnlazada;
public class ListaEnlazada {
    private Nodo primero; //señala el primer nodo, crea puntero primero
    private Nodo actual;

    public ListaEnlazada(){
        primero = null; //se declara primero = null
        actual = primero;
    }

    public void agregarNodo(int dat){
        Nodo nuevo; //crear el apuntador a nado, el puntador se llama nuevo
        nuevo = new Nodo( dato: dat, siguiente: primero); //crear un nodo con sus campos, el dato se guarda al nuevo
        primero = nuevo; //primero apunta al nuevo nodo
    }

    public void MostrarDatos(){
        if(primero == null) //no hay nodos
            System.out.println( x: "La lista está vacía!!");
        else{ //si hay nodos
            Nodo actual = primero;

            while(actual != null){
                System.out.print( actual.getDato() + " ");
                actual = actual.getSiguiente();
            }
            System.out.print( s: "\n");
        }
    }
}
```

```
public void borrarNodo(int dato){ // Método para eliminar un nodo de la lista enlazada
    if(primero == null) // Verifica si la lista está vacía
        System.out.println( x: "La lista está vacía!!"); // Imprime un mensaje indicando que la lista está vacía
    else{ // Si la lista no está vacía
        if(primero.getDato() == dato){ // Verifica si el primer nodo contiene el dato a eliminar
            primero = primero.getSiguiente(); // Actualiza el primer nodo para que apunte al siguiente nodo
        }
        else{ // Si el dato a eliminar no está en el primer nodo
            Nodo actual = primero; // Declaración de una variable local actual que apunta al primer nodo

            while(actual.getSiguiente() != null &&
                actual.getSiguiente().getDato() != dato) // Itera sobre la lista hasta encontrar el nodo a eliminar o llegar al final de la
                actual = actual.getSiguiente(); // Avanza al siguiente nodo de la lista

            if(actual.getSiguiente() == null) // Verifica si se llegó al final de la lista sin encontrar el dato a eliminar
                System.out.println( x: "El dato no existe en la lista!!"); // Imprime un mensaje indicando que el dato no existe en la lista
            else{ // Si se encontró el dato a eliminar
                actual.setSiguiente( siguiente: actual.getSiguiente().getSiguiente()); // Elimina el nodo de la lista enlazando el nodo anterior con
            }
        }
    }
}
```

```
package pckListaEnlazada;
```

```
public class Nodo {  
    private int dato;  
    private Nodo siguiente;  
  
    public Nodo(int dato, Nodo siguiente) {  
        this.dato = dato;  
        this.siguiente = siguiente;  
    }  
  
    public int getDato() {  
        return dato;  
    }  
  
    public void setDato(int dato) {  
        this.dato = dato;  
    }  
  
    public Nodo getSiguiente() {  
        return siguiente;  
    }  
  
    public void setSiguiente(Nodo siguiente) {  
        this.siguiente = siguiente;  
    }  
}
```

```
package pckListaEnlazada;
```

```
public class PruebaListaEnlazada {  
  
    public static void main(String[] args) {  
        ListaEnlazada lista1;  
        lista1 = new ListaEnlazada();  
  
        lista1.agregarNodo(12);  
        lista1.agregarNodo(14);  
        lista1.agregarNodo(8);  
        lista1.agregarNodo(4);  
  
        lista1.MostrarDatos();  
  
        lista1.borrarNodo(4);  
        lista1.MostrarDatos();  
    }  
}
```

Output - ListaEnlazada (run) ×

run:

4 8 14 12

8 14 12 BUILD SUCCESSFUL (total time: 0s)