



Universidad Nacional Autónoma de México



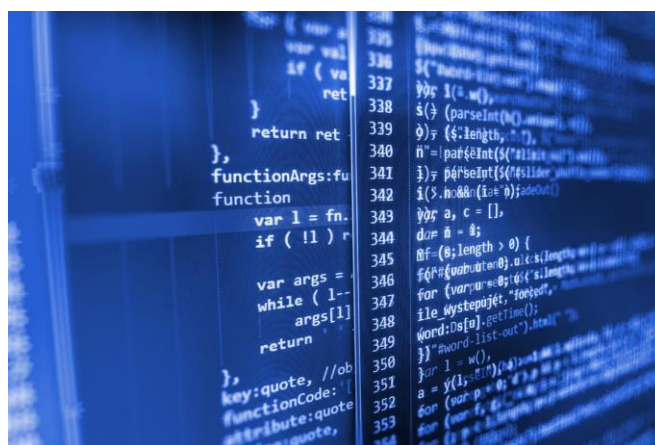
Facultad de Estudios Superiores Aragón

Ingeniería en Computación

Estructura de Datos

Jesús Hernández Cabrera

Axel Yahir Moreno Rodríguez



Turno Vespertino

Grupo 1360

```
1 public class NodosMain {
2     public static void main(String[] args) {
3
4         Nodo<Integer> head = new Nodo<>{dato:100,new Nodo<>{dato:200,new Nodo<>{dato:300, new Nodo<>{dato:400, new Nodo<>{dato:600, new Nodo<>{dato:700}}}}}};
5
6         Nodo<Integer> aux = head;
7         System.out.print("|");
8         while (aux != null){
9             System.out.print(aux.getData() + "| -> |");
10            aux = aux.getSiguiente();
11        }
12        System.out.println("null|");
13
14        head.getSiguiente().getSiguiente().setDato(dato:333);
15        Nodo<Integer> aux2 = head;
16        System.out.print("|");
17        while (aux2 != null){
18            System.out.print(aux2.getData() + "| -> |");
19            aux2 = aux2.getSiguiente();
20        }
21        System.out.println("null|");
22
23        head.getSiguiente().getSiguiente().getSiguiente().setSiguiente(new Nodo<>{dato:700});
24
25        Nodo<Integer> aux3 = head;
26        System.out.print("|");
27        while (aux3 != null){
28            System.out.print(aux3.getData() + "| -> |");
29            aux3 = aux3.getSiguiente();
30        }
31    }
32 }
```

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[50] -> [100] -> [200] -> [333] -> [400] -> [600] -> [700] -> [null]
PS C:\Users\axelx\Documents\FES\3er semestre\Estructura de Datos>
PS C:\Users\axelx\Documents\FES\3er semestre\Estructura de Datos>
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' 'NodosMain'
[100] -> [200] -> [300] -> [400] -> [600] -> [null]
[100] -> [200] -> [333] -> [400] -> [600] -> [null]
[100] -> [200] -> [333] -> [400] -> [600] -> [700] -> [null]
[50] -> [100] -> [200] -> [333] -> [400] -> [600] -> [700] -> [null]
PS C:\Users\axelx\Documents\FES\3er semestre\Estructura de Datos>
```

```

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

        Nodo<Integer> head = new Nodo<>(100,new Nodo<>(200,new Nodo<>(300,
new Nodo<>(400, new Nodo<>(600))));

        Nodo<Integer> aux = head;
        System.out.print("|");
        while (aux != null){
            System.out.print(aux.getDato() + "| -> |");
            aux = aux.getSiguiente();
        }
        System.out.println("null|");

        head.getSiguiente().getSiguiente().setDato(333);
        Nodo<Integer> aux2 = head;
        System.out.print("|");
        while (aux2 != null){
            System.out.print(aux2.getDato() + "| -> |");
            aux2 = aux2.getSiguiente();
        }
        System.out.println("null|");

        head.getSiguiente().getSiguiente().getSiguiente().getSiguiente().set
Siguiete(new Nodo<>(700));

        Nodo<Integer> aux3 = head;
        System.out.print("|");
        while (aux3 != null){
            System.out.print(aux3.getDato() + "| -> |");
            aux3 = aux3.getSiguiente();
        }
        System.out.println("null|");

        head = new Nodo<>(50,head);
        Nodo<Integer> aux4 = head;
        System.out.print("|");
        while (aux4 != null){
            System.out.print(aux4.getDato() + "| -> |");
            aux4 = aux4.getSiguiente();
        }
        System.out.println("null|");
    }
}

```

```

    }
}
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 +
            '}';
    }
}

```

}