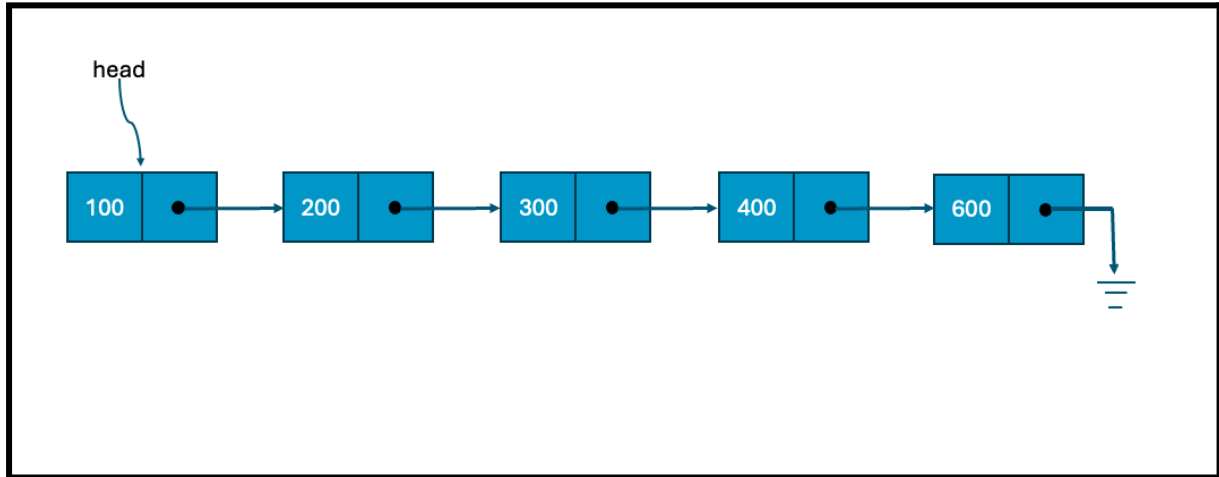


### Tarea 3- Prácticas con la clase nodo

- Crear la estructura de la imagen anexa.



- Imprimir todo usando un **while**

```
1 package listasEnlazadas;
2
3 public class ListasEnlazadas {
4
5     public static void main(String[] args) {
6
7         Nodo<Integer> head = new Nodo<>(100, new Nodo<>(200, new Nodo<>(300, new Nodo<>(400, new Nodo<>(600))));
8
9         Nodo<Integer> aux = head;
10        System.out.print("|");
11        while (aux != null) {
12            System.out.print(aux.getDato() + " | -> |");
13            aux = aux.getSiguiete();
14        }
15        System.out.print("null|");
16    }
17 }
18
19
20
```

Output - listasEnlazadas (run)

```
run:
|100| -> |200| -> |300| -> |400| -> |600| -> |null|BUILD SUCCESSFUL (total time: 0 seconds)
```

- Cambiar el valor del 3º nodo de 300 a 333
- Imprimir todo nuevamente

```

1 package listasenzadas;
2
3 public class ListasEnlazadas {
4
5     public static void main(String[] args) {
6
7         Nodo<Integer> head = new Nodo<>(100, new Nodo<>(200, new Nodo<>(300, new Nodo<>(400, new Nodo<>(600))));
9         head.getSiguiete().getSiguiete().setDato(333);
10
11         Nodo<Integer> aux = head;
12         System.out.print("|");
13         while (aux != null){
14             System.out.print(aux.getDato() + "| -> |");
15             aux = aux.getSiguiete();
16         }
17         System.out.print("null|");
18     }
19 }
20
21

```

Output - listasEnlazadas (run)

```

run:
|100| -> |200| -> |333| -> |400| -> |600| -> |null|BUILD SUCCESSFUL (total time: 0 seconds)

```

- Insertar un nodo 700 después del nodo 600 (al final)
- Imprimir de nuevo todo con un while.

```

1 package listasenzadas;
2
3 public class ListasEnlazadas {
4
5     public static void main(String[] args) {
6
7         Nodo<Integer> head = new Nodo<>(100, new Nodo<>(200, new Nodo<>(300, new Nodo<>(400, new Nodo<>(600))));
9         head.getSiguiete().getSiguiete().setDato(333);
10         head.getSiguiete().getSiguiete().getSiguiete().setSiguiete(new Nodo<>(700,null));
11
12         Nodo<Integer> aux = head;
13         System.out.print("|");
14         while (aux != null){
15             System.out.print(aux.getDato() + "| -> |");
16             aux = aux.getSiguiete();
17         }
18         System.out.print("null|");
19     }
20 }
21
22

```

Output - listasEnlazadas (run)

```

run:
|100| -> |200| -> |333| -> |400| -> |600| -> |700| -> |null|BUILD SUCCESSFUL (total time: 0 seconds)

```

- Insertar un nodo con valor 50 al principio (antes del nodo 100)
- Imprimir todo de nuevo.

```

1 package listasenlazadas;
2
3
4 public class ListasEnlazadas {
5
6     public static void main(String[] args) {
7
8         Nodo<Integer> head = new Nodo<>(100, new Nodo<>(200, new Nodo<>(300, new Nodo<>(400, new Nodo<>(600))));
9         head.getSiguiente().getSiguiente().setDato(333);
10        head.getSiguiente().getSiguiente().getSiguiente().setSiguiente(new Nodo<>(700,null));
11
12        //Agregar un nodo al inicio de la lista
13        head = new Nodo<>(50,head);
14
15        Nodo<Integer> aux = head;
16        System.out.print("|");
17        while (aux != null){
18            System.out.print(aux.getDato() + "| -> |");
19            aux = aux.getSiguiente();
20        }
21        System.out.print("null|");
22    }
23 }

```

Output - listasenlazadas (run)

```

run:
|50| -> |100| -> |200| -> |333| -> |400| -> |600| -> |700| -> |null|BUILD SUCCESSFUL (total time: 0 seconds)

```

## Clase Nodo

```

1
2 package listasenlazadas;
3
4 public class Nodo<T> {
5     private T dato;
6     private Nodo<T> siguiente;
7
8     public Nodo() {
9     }
10
11    public Nodo(T dato) {
12        this.dato = dato;
13    }
14
15    public Nodo(T dato, Nodo<T> siguiente) {
16        this.dato = dato;
17        this.siguiente = siguiente;
18    }
19
20    public T getDato() {
21        return dato;
22    }
23
24    public void setDato(T dato) {
25        this.dato = dato;
26    }
27
28    public Nodo<T> getSiguiente() {
29        return siguiente;
30    }
31 }

```

```
32  [-] public void setSiguiente(Nodo<T> siguiente) {  
33      |     this.siguiente = siguiente;  
34      | }  
35  
36      @Override  
37  [●] [-] public String toString() {  
38      |     return "Nodo{" +  
39      |         |         "dato=" + dato +  
40      |         |         ", siguiente=" + siguiente +  
41      |         |         "'}';  
42      |     }  
43  }  
44
```