



Universidad Nacional Autónoma de  
México

Facultad de Estudios Superiores  
Aragón

Alumno: Leal Sanchez Bryan Uriel

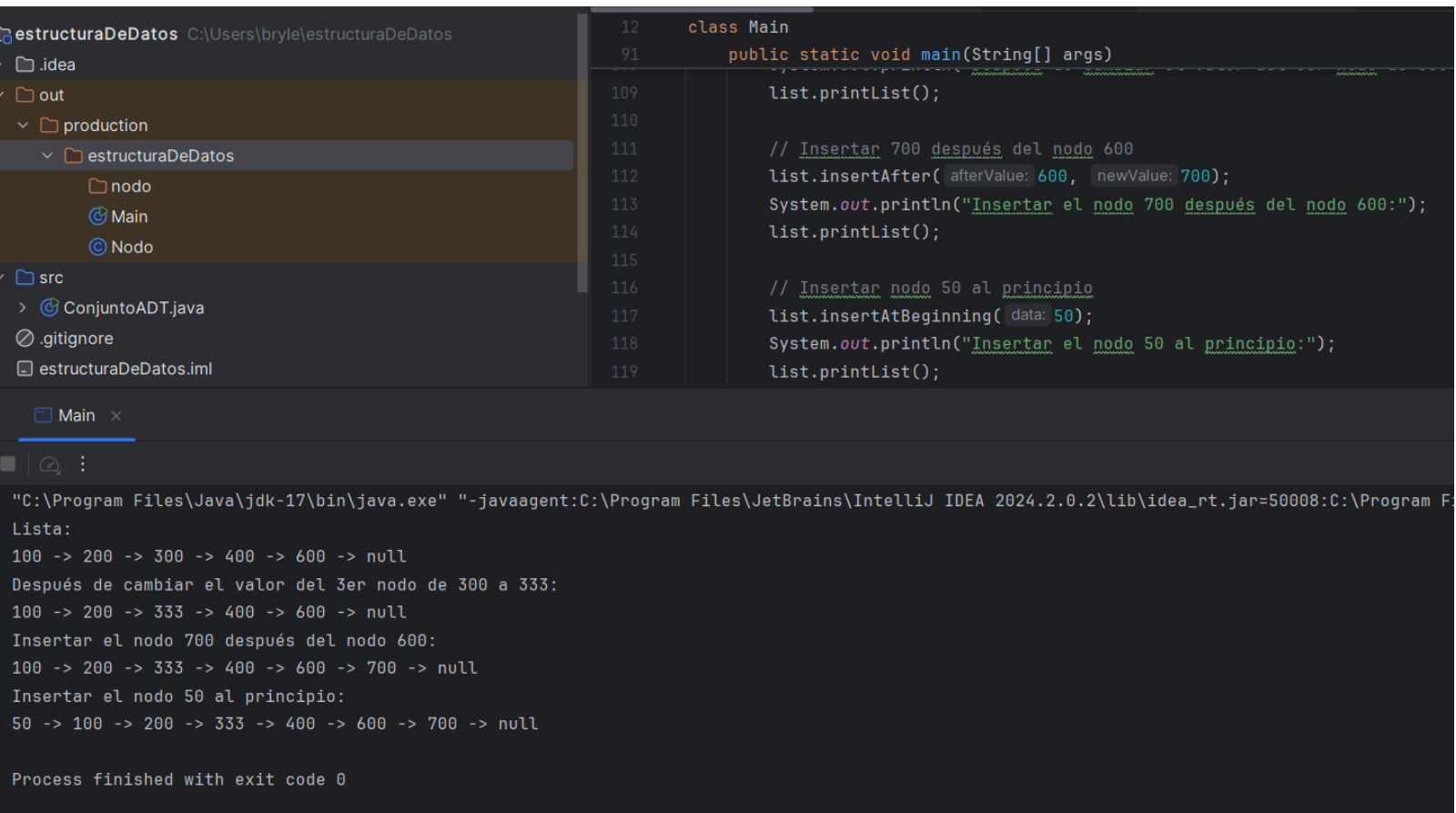
Materia: Estructura de Datos

Grupo: 1360

Profesor: Hernández Cabrera Jesús

Nodos

# Ejecución del Programa



```
12 class Main
91 public static void main(String[] args)
109 list.printList();
110
111 // Insertar 700 después del nodo 600
112 list.insertAfter( afterValue: 600, newValue: 700);
113 System.out.println("Insertar el nodo 700 después del nodo 600:");
114 list.printList();
115
116 // Insertar nodo 50 al principio
117 list.insertAtBeginning( data: 50);
118 System.out.println("Insertar el nodo 50 al principio:");
119 list.printList();
```

Process finished with exit code 0

Output:

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.0.2\lib\idea_rt.jar=50008:C:\Program F:
Lista:
100 -> 200 -> 300 -> 400 -> 600 -> null
Después de cambiar el valor del 3er nodo de 300 a 333:
100 -> 200 -> 333 -> 400 -> 600 -> null
Insertar el nodo 700 después del nodo 600:
100 -> 200 -> 333 -> 400 -> 600 -> 700 -> null
Insertar el nodo 50 al principio:
50 -> 100 -> 200 -> 333 -> 400 -> 600 -> 700 -> null
```

# Capturas del Código

```
1  class Nodo { 12 usages
2      int data; 4 usages
3      Nodo next; 11 usages
4
5      public Nodo(int data) 3 usages
6      {
7          this.data = data;
8          this.next = null;
9      }
10 }
11     //inicio de la lista
12 class Main
13 {
14     private Nodo head; 10 usages
15     > public Main() { this.head = null; }
16
17
18
19
20     public void add(int data)
21     {
22         Nodo newNode = new Nodo(data);
23         if (head == null) {
24             head = newNode;
25         }
26         else
27         {
28             Nodo current = head;
29             while (current.next != null)
30             {
31                 current = current.next;
32             }
33             current.next = newNode;
```

```
class Main
    public void add(int data)
    {
    }
}

public void printList() 8 usages
{
    Nodo current = head;
    while (current != null)
    {
        System.out.print(current.data + " -> ");
        current = current.next;
    }
    System.out.println("null");
}

public void updateThirdNode(int newData) 2 usages
{
    if (head == null) return;

    Nodo current = head;
    int count = 1;

    while (current != null && count < 3)
    {
        current = current.next;
        count++;
    }

    if (current != null)
    {
```

```
12     class Main
48         public void updateThirdNode(int newData) 2 usages
64     }
65 }
66
67     public void insertAfter(int afterValue, int newValue) 2 usages
68     {
69         Nodo current = head;
70
71         while (current != null)
72         {
73             if (current.data == afterValue)
74             {
75                 Nodo newNode = new Nodo(newValue);
76                 newNode.next = current.next;
77                 current.next = newNode;
78                 break;
79             }
80             current = current.next;
81         }
82     }
83
84     public void insertAtBeginning(int data) 2 usages
85     {
86         Nodo newNode = new Nodo(data);
87         newNode.next = head;
88         head = newNode;
89     }
90
91     public static void main(String[] args)
```

```
12     class Main
84     public void insertAtBeginning(int data) 2 usages
86         Nodo newNode = new Nodo(data);
87         newNode.next = head;
88         head = newNode;
89     }
90
91     public static void main(String[] args)
92     {
93         Main list = new Main();
94
95         // lista original
96         list.add(100);
97         list.add(200);
98         list.add(300);
99         list.add(400);
100        list.add(600);
101
102        // Imprimir la lista original
103        System.out.println("Lista:");
104        list.printList();
105
106        // Cambiar valor del 3er nodo de 300 a 333
107        list.updateThirdNode( newData: 333);
108        System.out.println("Después de cambiar el valor del 3er nodo de 300 a 333:");
109        list.printList();
110
111        // Insertar 700 después del nodo 600
112        list.insertAfter( afterValue: 600, newValue: 700);
113        System.out.println("Insertar el nodo 700 después del nodo 600:");
114        list.printList();
```

```
12     class Main
13
14     {
15         public static void main(String[] args)
16         {
17             list.add(300);
18             list.add(400);
19             list.add(600);
20
21             // Imprimir la lista original
22             System.out.println("Lista:");
23             list.printList();
24
25             // Cambiar valor del 3er nodo de 300 a 333
26             list.updateThirdNode( newData: 333);
27             System.out.println("Después de cambiar el valor del 3er nodo de 300 a 333:");
28             list.printList();
29
30             // Insertar 700 después del nodo 600
31             list.insertAfter( afterValue: 600, newValue: 700);
32             System.out.println("Insertar el nodo 700 después del nodo 600:");
33             list.printList();
34
35             // Insertar nodo 50 al principio
36             list.insertAtBeginning( data: 50);
37             System.out.println("Insertar el nodo 50 al principio:");
38             list.printList();
39         }
40     }
41 }
```