



**Universidad Nacional Autónoma de México**



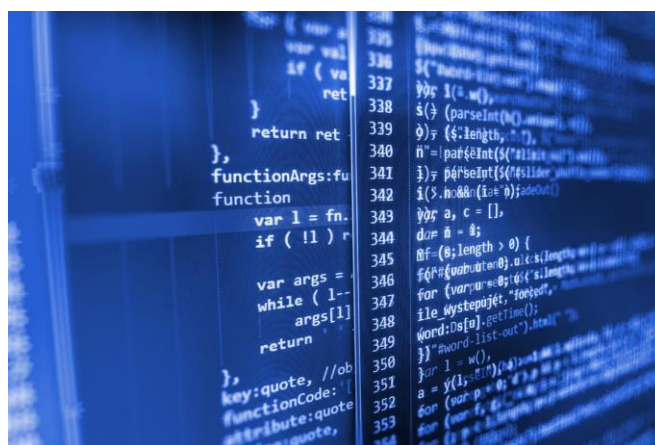
**Facultad de Estudios Superiores Aragón**

## **Ingeniería en Computación**

### **Estructura de Datos**

**Jesús Hernández Cabrera**

**Axel Yahir Moreno Rodríguez**



**Turno Vespertino**

**Grupo 1360**

```
J Main.java ...sistexp J SistemaExperto.java J Main.java ...pilas X J Llaves.java J Stack.java
Estructura de Datos > Tareas > Tarea 9 > tarea9 > src > main > java > unam > mx > pilas > J Main.java > Main > main(String[])
1 package unam.mx.pilas;
2 public class Main{
3     Run | Debug
4     public static void main(String[] args) {
5         Llaves llaves = new Llaves();
6
7         String test1 = "{(a + b) * (c + d)}";
8         String test2 = "{(a + b) * c + d}";
9         String test3 = "{((()))}";
10        String test4 = "{((()))}";
11        String test5 = "package unam.mx.pilas; public class main{public static void main(String[] args){Llaves llaves = new Llaves();";
12        String test6 = "package unam.mx.pilas; public class main{public static void main(String[] args){Llaves llaves = new Llaves();";
13
14        // condicion ? valor_si_true : valor_si_false;
15        // true = Balanceado // false = No Balanceado
16        System.out.println("Prueba 1: " + (llaves.checarBalanceo(test1) ? "Balanceado" : "No balanceado"));
17        System.out.println("Prueba 2: " + (llaves.checarBalanceo(test2) ? "Balanceado" : "No balanceado"));
18        System.out.println("Prueba 3: " + (llaves.checarBalanceo(test3) ? "Balanceado" : "No balanceado"));
19        System.out.println("Prueba 4: " + (llaves.checarBalanceo(test4) ? "Balanceado" : "No balanceado"));
20        System.out.println("Prueba 5: " + (llaves.checarBalanceo(test5) ? "Balanceado" : "No balanceado"));
21        System.out.println("Prueba 6: " + (llaves.checarBalanceo(test6) ? "Balanceado" : "No balanceado"));
22    }
}
```

```
Estructura de Datos > Tareas > Tarea 9 > tarea9 > src > main > java > unam > mx > pilas > J Llaves.java > Llaves > checarBalanceo(String)
1 package unam.mx.pilas;
2 import java.util.Stack;
3
4 public class Llaves {
5     public boolean checarBalanceo(String texto) {
6         Stack<Character> stack = new Stack<>();
7
8         for (int i = 0; i < texto.length(); i++) {
9             char currentChar = texto.charAt(i);
10            if (currentChar == '{' || currentChar == '(') {
11                stack.push(currentChar);
12            }
13            else if (currentChar == '}' || currentChar == ')') {
14                if (stack.isEmpty()) {
15                    return false;
16                }
17                char topChar = stack.pop();
18                if ((currentChar == '}' && topChar != '{') || (currentChar == ')' && topChar != '(')) {
19                    return false;
20                }
21            }
22        }
23        return stack.isEmpty();
24    }
25 }
```

Estructura de Datos > Tareas > Tarea 9 > tarea9 > src > main > java > unam > mx > pilas > Stack.java > Stack > toString()

```
1 package unam.mx.pilas;
2 import java.util.Arrays;
3
4 public class Stack {
5     private int top;
6     private int[] stackArray;
7     private int maxSize;
8
9     public Stack(int size) {
10         maxSize = size;
11         stackArray = new int[maxSize];
12         top = -1;
13     }
14
15     public boolean isEmpty() {
16         boolean r = false;
17         if(maxSize == 0){
18             r = true;
19         }
20         return r;
21     }
22
23     public int length() {
24         return top + 1;
25     }
26 }
```

Estructura de Datos > Tareas > Tarea 9 > tarea9 > src > main > java > unam > mx > pilas > Stack.java > Stack > toString()

```
4 public class Stack {
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27     public int pop() {
28         if (!isEmpty()) {
29             return stackArray[top--];
30         } else {
31             System.out.println(x:"La pila esta vacia");
32             return -1;
33         }
34     }
35
36     public int peek() {
37         if (!isEmpty()) {
38             return stackArray[top];
39         } else {
40             System.out.println(x:"La pila esta vacia");
41             return -1;
42         }
43     }
44
45     public void push(int value) {
46         if (!isFull()) {
47             stackArray[++top] = value;
48         } else {
49             System.out.println(x:"La pila esta llena");
50         }
51     }
52 }
```

```
6 String test1 = "{(a + b) * (c + d)}";
7 String test2 = "{(a + b) * c + d}";
8 String test3 = "{((()))}";
9 String test4 = "{((()))}";
10 String test5 = "package unam.mx.pilas; public class main{public static void main(String[] args){LLaves llaves = new LLaves();";
11 String test6 = "package unam.mx.pilas; public class main{public static void main(String[] args){LLaves llaves = new LLaves();";
12
13 // condicion 1: valor de top > valor de maxSize
```

PROBLEMS 21 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Run: Main + v [ ] [ ] ... ^ x

Prueba 1: Balanceado  
Prueba 2: No balanceado  
Prueba 3: Balanceado  
Prueba 4: No balanceado  
Prueba 5: Balanceado  
Prueba 6: No balanceado

PS C:\Users\axelx\Documents\FES\3er semestre\Estructura de Datos>

```

package unam.mx.pilas;
public class Main{
public static void main(String[] args) {
    Llaves llaves = new Llaves();

    String test1 = "{(a + b) * (c + d)}";
    String test2 = "{(a + b) * c + d}";
    String test3 = "{{(())}}";
    String test4 = "{{(())}}";
    String test5 = "package unam.mx.pilas; public class main{public static
void main(String[] args){Llaves llaves = new Llaves();}}";
    String test6 = "package unam.mx.pilas; public class main{public static
void main(String[] args){Llaves llaves = new Llaves();}}";

    // condicion ? valor_si_true : valor_si_false;
    // true = Balanceado // false = No Balanceado
    System.out.println("Prueba 1: " + (llaves.checarBalanceo(test1) ?
"Balanceado" : "No balanceado"));
    System.out.println("Prueba 2: " + (llaves.checarBalanceo(test2) ?
"Balanceado" : "No balanceado"));
    System.out.println("Prueba 3: " + (llaves.checarBalanceo(test3) ?
"Balanceado" : "No balanceado"));
    System.out.println("Prueba 4: " + (llaves.checarBalanceo(test4) ?
"Balanceado" : "No balanceado"));
    System.out.println("Prueba 5: " + (llaves.checarBalanceo(test5) ?
"Balanceado" : "No balanceado"));
    System.out.println("Prueba 6: " + (llaves.checarBalanceo(test6) ?
"Balanceado" : "No balanceado"));
    }
}

```

```

package unam.mx.pilas;
import java.util.Stack;

public class Llaves {
    public boolean checarBalanceo(String texto) {
        Stack<Character> stack = new Stack<>();

        for (int i = 0; i < texto.length(); i++) {
            char currentChar = texto.charAt(i);
            if (currentChar == '{' || currentChar == '(') {
                stack.push(currentChar);
            }
            else if (currentChar == '}' || currentChar == ')') {

```

```

        if (stack.isEmpty()) {
            return false;
        }
        char topChar = stack.pop();
        if ((currentChar == '}' && topChar != '{') || (currentChar
== ')') && topChar != '(')) {
            return false;
        }
    }
}
return stack.isEmpty();
}
}

```

```

package unam.mx.pilas;
import java.util.Arrays;

public class Stack {
    private int top;
    private int[] stackArray;
    private int maxSize;

    public Stack(int size) {
        maxSize = size;
        stackArray = new int[maxSize];
        top = -1;
    }

    public boolean isEmpty() {
        boolean r = false;
        if(maxSize == 0){
            r = true;
        }
        return r;
    }

    public int length() {
        return top + 1;
    }

    public int pop() {
        if (!isEmpty()) {
            return stackArray[top--];
        } else {

```

```

        System.out.println("La pila esta vacia");
        return -1;
    }
}

public int peek() {
    if (!isEmpty()) {
        return stackArray[top];
    } else {
        System.out.println("La pila esta vacia");
        return -1;
    }
}

public void push(int value) {
    if (!isFull()) {
        stackArray[++top] = value;
    } else {
        System.out.println("La pila esta llena");
    }
}

public boolean isFull() {
    return top == maxSize - 1;
}

@Override
public String toString() {
    if (isEmpty()) {
        return "La pila esta vacia.";
    }
    return "Estado de la pila: " +
Arrays.toString(Arrays.copyOfRange(stackArray, 0, top + 1));
}
}

```