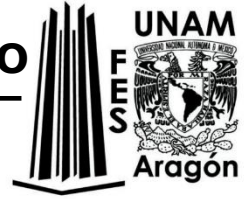




UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES  
ARAGON



## TAREA 9

**P R E S E N T A**

Alexis Hernández Zamudio

**A PROFESOR**

Jesús Hernández Cabrera

**Gpo:1158**

URL del repositorio:

<https://github.com/TyrBalder1439/Estructura-de-Datos->



Ciudad Nezahualcóyotl, EDOMEX. 24 de septiembre del 2024

```
1 public class Stack<T> { 2 usages  👤 felipe herman
2     private Nodo<T> tope; 8 usages
3     private int tamaño; 4 usages
4     private static class Nodo<T> { 4 usages  👤 felipe herman
5         T dato; 3 usages
6         Nodo<T> siguiente; 2 usages
7
8         Nodo(T dato) { 1 usage  👤 felipe herman
9             this.dato = dato;
10        }
11    }
12
13    public Stack() { 1 usage  👤 felipe herman
14        tope = null;
15        tamaño = 0;
16    }
17
18    public boolean isEmpty() { 4 usages  👤 felipe herman
19        return tope == null;
20    }
21
22    public int length() { no usages  👤 felipe herman
23        return tamaño;
24    }
```

```
24  ✓ public T pop() { 1 usage  👤 felipe herman
25  ✓     if (isEmpty()) {
26      |         throw new IllegalStateException("La pila está vacía");
27      |     }
28      |     T dato = tope.dato;
29      |     tope = tope.siguiiente;
30      |     tamano--;
31      |     return dato;
32  | }
33
34  ✓ public T peek() { 1 usage  👤 felipe herman
35  ✓     if (isEmpty()) {
36      |         throw new IllegalStateException("La pila está vacía");
37      |     }
38      |     return tope.dato;
39  | }
40
41  ✓ public void push(T value) { 1 usage  👤 felipe herman
42      |     Nodo<T> nuevoNodo = new Nodo<>(value);
43      |     nuevoNodo.siguiiente = tope;
44      |     tope = nuevoNodo;
45      |     tamano++;
46  | }
47  | }
48
```

```

Stack.java  BalanceoParentesisLlaves.java x
1 public class BalanceoParentesisLlaves {  felipe herman *
2
3 @  public static boolean estaBalanceado(String texto) {  2 usages  felipe herman *
4     Stack<Character> pila = new Stack<>();
5
6     for (int i = 0; i < texto.length(); i++) {
7         char c = texto.charAt(i);
8
9         if (c == '{' || c == '(') {
10             pila.push(c);
11         } else if (c == '}' || c == ')') {
12             if (pila.isEmpty()) {
13                 return false;
14             }
15             char tope = pila.peek();
16             if ((c == '}' && tope != '{') || (c == ')' && tope != '(')) {
17                 return false;
18             }
19             pila.pop();
20         }
21     }
22
23     return pila.isEmpty();
24 }

```

```

25 }
26 public static void main(String[] args) {  felipe herman
27     String textoDePrueba1 = "{(())}";
28     String textoDePrueba2 = "{()}";
29
30     System.out.println("Texto 1: {(())} " + (estaBalanceado(textoDePrueba1) ? " Balanceado" : "No balanceado"));
31     System.out.println("Texto 2: {()} " + (estaBalanceado(textoDePrueba2) ? " Balanceado" : "No balanceado"));
32 }
33 }
34

```

"C:\Program Files\Eclipse Adoptium\jdk-21.0

Texto 1: {(())} Balanceado

Texto 2: {(}) No balanceado

Process finished with exit code 0

|