



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

INGENIERÍA EN COMPUTACIÓN

Estructura de datos

TAREA 9

Prof. Jesús Hernandez Cabrera

Grupo: 1360

Arroyo Gutierrez Rodrigo Javier

```
Stack.java  BalanceDePP.java x
stack C:\U...
.idea
.out
src
  @ Balanc
  @ Stack
.gitignore
stack.iml
External Libr...
Scratches an...

1  import java.util.Scanner;
2
3  public class BalanceDePP {
4  public static void main(String[] args) {
5      Scanner scanner = new Scanner(System.in);
6
7      System.out.println("Introduce el texto para verificar el balanceo de paréntesis y llaves:");
8      String texto = scanner.nextLine();
9
10     Stack<Character> pilaBalance = new Stack<>();
11     pilaBalance.BalancePp(texto);
12     scanner.close();
13 }
14 }
```

BalanceDePP x

IDEA 2024.2.1\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath C:\Users\nodr1\IdeaProjects\stack\out\production\stack BalanceDePP

Introduce el texto para verificar el balanceo de paréntesis y llaves:
HalaAaAaA{(nada)(nada)}

Push {
Push (
Pop)
Push (
Pop)
Pop }

Esta balanceado

```
Stack.java  BalanceDePP.java x
stack C:\U...
.idea
.out
src
  @ Balanc
  @ Stack
.gitignore
stack.iml
External Libr...
Scratches an...

1  import java.util.Scanner;
2
3  public class BalanceDePP {
4  public static void main(String[] args) {
5      Scanner scanner = new Scanner(System.in);
6
7      System.out.println("Introduce el texto para verificar el balanceo de paréntesis y llaves:");
8      String texto = scanner.nextLine();
9
10     Stack<Character> pilaBalance = new Stack<>();
11     pilaBalance.BalancePp(texto);
12     scanner.close();
13 }
14 }
```

BalanceDePP x

<1 internal line>

Introduce el texto para verificar el balanceo de paréntesis y llaves:
Test de balance{de parenthesis() y llaves }

Push {
Push (
Pop)
Push {
Pop }

Desbalance, falta cerrar

Process finished with exit code 0

```
Stack.java  BalanceDePP.java x
stack C:\U...
.idea
out
src
BalanceDePP
Stack
.gitignore
stack.iml
External Libr...
Scratches an...

3 public class BalanceDePP {
4     public static void main(String[] args) {
5
6
7         System.out.println("Introduce el texto para verificar el balanceo de paréntesis y llaves:");
8         String texto = scanner.nextLine();
9
10        Stack<Character> pilaBalance = new Stack<>();
11        pilaBalance.BalancePp(texto);
12        scanner.close();
13    }
14 }
15
16

BalanceDePP x
<1 internal line>
Introduce el texto para verificar el balanceo de paréntesis y llaves:
Hola abra( y cierra) y cierra otra vez)
Push (
Pop )
Exception in thread "main" java.lang.RuntimeException Create breakpoint : Desbalance, no se puede hacer pop sin hacer un push primero
    at Stack.pop(Stack.java:22)
    at Stack.BalancePp(Stack.java:46)
    at BalanceDePP.main(BalanceDePP.java:11)
Process finished with exit code 1
```

```
.java  BalanceDePP.java x
import java.util.Scanner;

public class BalanceDePP {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Introduce el texto para verificar el balanceo de paréntesis y llaves:");
        String texto = scanner.nextLine();

        Stack<Character> pilaBalance = new Stack<>();
        pilaBalance.BalancePp(texto);
        scanner.close();
    }
}
```

```

Stack.java × BalanceDePP.java
1  import java.util.ArrayList;
2
3  public class Stack<T> { 4 usages
4      private ArrayList<T> data; 11 usages
5
6      public Stack() { 2 usages
7          this.data = new ArrayList<>();
8      }
9
10     public boolean isEmpty() { 2 usages
11         return data.isEmpty();
12     }
13
14     public int length() { no usages
15         return data.size();
16     }
17
18     public T pop() { 2 usages
19         if (!data.isEmpty()) {
20             return data.remove(data.size() - 1);
21         }
22         throw new RuntimeException("Desbalance, no se puede hacer pop sin hacer un push primero");
23     }
24
25     public T peek() { no usages
26         if (!isEmpty()) {
27             return data.get(data.size() - 1);
28         }
29         throw new RuntimeException("Stack is empty");
30     }
31 }

```

```

Stack.java × BalanceDePP.java
3  public class Stack<T> { 4 usages
32     public void push(T value) { 2 usages
33         data.add(value);
34     }
35
36     @Override
37     public void BalancePp(String texto) { 1 usage
38         Stack<Character> pilaBalance = new Stack<>();
39         for (int i = 0; i < texto.length(); i++) {
40             char caracterActual = texto.charAt(i);
41
42             if (caracterActual == '(') {
43                 pilaBalance.push(caracterActual);
44                 System.out.println("Push (");
45             }
46             else if (caracterActual == ')') {
47                 pilaBalance.pop();
48                 System.out.println("Pop )");
49             } else if (caracterActual == '{') {
50                 pilaBalance.push(caracterActual);
51                 System.out.println("Push {");
52             } else if (caracterActual == '}') {
53                 pilaBalance.pop();
54                 System.out.println("Pop }");
55             }
56         }
57     }

```

```

        if (!pilaBalance.isEmpty()) {
            System.out.println("Desbalance, falta cerrar");
        } else {
            System.out.println("Esta balanceado");
        }
    }

    @Override
    public String toString() {
        StringBuilder info = new StringBuilder("-----\n");
        for (int i = data.size() - 1; i >= 0; i--) {
            info.append(data.get(i)).append("\n|---|\n");
        }
        return info.toString();
    }
}

```