

Evaluación de balanceo de paréntesis y llaves

- Programar el ADT pila(stack)

```
import java.util.EmptyStackException;
public class StackADT<T> {

    private Node<T> top;
    private int Tamanio;

    //Clase interna para un nodo
    private static class Node<T> {
        private T data;
        private Node<T> next;

        public Node(T data) {
            this.data = data;
            this.next = null;
        }
    }

    public StackADT() {
        top = null;
        Tamanio = 0;
    }

    //Metodo para agregar un elemento a la pila
    public void push(T data) {
        Node<T> newNode = new Node<>(data);
        newNode.next = top;
        top = newNode;
        Tamanio++;
    }

    //Metodo para sacar un elemento de la pila
    public T pop() {
        if (isEmpty()) {
            throw new EmptyStackException();
        }
        T data = top.data;
        top = top.next;
        Tamanio--;
        return data;
    }
}
```

```

        //Metodo para ver el elemento de mas arriba sin
sacarlo
        public T peek() {
            if (isEmpty()) {
                throw new EmptyStackException();
            }
            return top.data;
        }

        //Verifica si la pila esta vacia
        public boolean isEmpty() {
            return top == null;
        }

        //Obtener el tamaño de la pila
        public int size() {
            return Tamano;
        }
    }
}

```

- Implementar el programa que dada una entrada de texto, verifique si este contiene llaves balanceadas es decir que cada llave "{" tiene su respectiva llave de cierre "}". De igual forma con los paréntesis "(" y ")".

```

public class Balanceo {

    //Verifica si los caracteres de apertura y cierre son
del mismo tipo
    private static boolean esPareja(char apertura, char
cierre) {
        return (apertura == '(' && cierre == ')') ||
            (apertura == '{' && cierre == '}') ||
            (apertura == '[' && cierre == ']');
    }

    //Verifica si los parentesis y llaves estan
balanceados
    public static boolean estanBalanceados(String
expresion) {
        StackADT<Character> pila = new StackADT<>();

        //Recorremos cada caracter de la expresion
        for (char c : expresion.toCharArray()) {

```

```

        //Si encontramos un caracter de apertura, lo
agregamos a la pila
        if (c == '(' || c == '{' || c == '[') {
            pila.push(c);
        }
        //Si encontramos un caracter de cierre
        else if (c == ')' || c == '}' || c == ']') {
            // Si la pila esta vacia, significa que no
hay un caracter de apertura correspondiente
            if (pila.isEmpty()) {
                return false;
            }
            //Sacamos el elemento superior de la pila
y verificamos si es del mismo tipo
            char apertura = pila.pop();
            if (!esPareja(apertura, c)) {
                return false;
            }
        }
    }

    //Al final, la pila debe estar vacia si los
parentesis y llaves estan balanceados
    return pila.isEmpty();
}
}

```

- Programar una función o método main con un texto de prueba.

