

# Balanceo de signos de agrupación

## Sobre el método

En este proyecto se muestra la implementación de un método para rectificar el balanceo de [paréntesis](#), [corchetes](#) y [llaves](#) en una cadena de caracteres. Dicha cadena de caracteres puede ser tanto una oración como una operación aritmética. Lo que este método hace es verificar si los signos de agrupación están puestos de manera correcta en la cadena de caracteres. En la Tabla 1, se muestran diferentes expresiones aritméticas y el resultado que se obtendría al evaluarlas con el método.

Expresión aritmética	Resultado
$[(5-6)*7] / 4 + \{5*3-10\}$	FALSE
$((12-3+6)*8$	FALSE
$\{6*9*7\} / (5+3)$	FALSE
$[ \{40-5\} * 3 ]$	TRUE

Tabla 1: Ejemplo de input y output del método

## Pilas

La mejor manera de implementar el método de balanceo de signos de agrupación es con pilas (*stacks* en inglés). La particularidad de tales estructuras de datos es que solo tienen un extremo, llamado tope, por el que se insertan y retiran los datos. Las pilas, vistas como un tipo de dato abstracto, tienen cuatro operaciones básicas:

- *isEmpty(): boolean*      regresa si la pila está vacía
- *peek(): T*                      regresa el dato que está en el tope de la pila sin modificarla
- *pop(): T*                        saca y regresa el dato en el tope de la pila
- *push(T): void*                ingresa un dato al tope de la pila

## Configuración

Sigue los siguientes pasos para poder usar el código.



Diagrama 1: Pasos a seguir

A continuación, están los cinco pasos más detallados:

1. Descarga algún IDE que pueda leer y ejecutar Java como NetBeans o Eclipse.
2. Crea un nuevo proyecto. En este puedes poner las clases de pilas y también la clase en la que desees usar el método.
3. Implementa la pila al generar una clase por cada archivo dentro de la carpeta *pila*.
4. Copia el código del método de balanceo de signos de agrupación dentro de la clase en el que lo desees usar. Este método se encuentra dentro de la carpeta *balancear*.
5. Llama al método con la cadena de caracteres que desees evaluar como atributo de entrada.

## Acerca del algoritmo

El algoritmo para resolver el problema del balanceo de signos de agrupación se basa en ir carácter por carácter verificando si la cadena es correcta.

- Si el carácter a revisar no es un signo de agrupación seguimos adelante.
- Si el carácter es un signo de agrupación de apertura `[, {, (`, entonces guardamos este carácter dentro de la pila.
- Si el carácter es un signo de agrupación de cierre `], }, )`, entonces sacamos lo que haya en el tope de la pila y verificamos que sea el mismo tipo de signo de agrupación.

El algoritmo seguirá verificando cada carácter mientras no hayamos encontrado alguna inconsistencia. Finalmente, ya habiendo revisado toda la cadena, podremos saber que esta es correcta si es que la pila está vacía.

## Funcionalidad

La mayor funcionalidad que tiene este método es para programar una [calculadora](#). Las verificaciones de paréntesis, llaves y corchetes se deben de realizar antes de intentar resolver una expresión aritmética, pues si el balanceo no se checa, es muy probable que el código vaya a generar excepciones al momento de uso.

## Estado

El método está completo y sirve correctamente.

## Contacto

Para cualquier duda o sugerencia no dudes en contactarme [@nataliahc](#).