



**SOFT-10 Estructuras de datos**

**Primer estudio de caso**

# Aplicación de las pilas en el análisis de expresiones aritméticas

**Sección**

SCV2

**Estudiante**

[Andrés Vargas Guevara](#)

**Profesor**

[Romario Salas Cerdas](#)

**Fecha**

19 de Octubre, 2025

## **Tabla de contenidos**

|                                   |           |
|-----------------------------------|-----------|
| <b>Introducción</b>               | <b>3</b>  |
| <b>Desarrollo</b>                 | <b>4</b>  |
| <b>Implementación</b>             | <b>8</b>  |
| <b>Casos de prueba</b>            | <b>14</b> |
| <b>Referencias bibliográficas</b> | <b>19</b> |

## **Introducción**

Este documento presenta el primer caso de estudio, el cual consiste en el desarrollo de una investigación en torno a la aplicación de las pilas en la programación de una funcionalidad importante del software de sistema, acompañada por la correspondiente implementación de una pila programada en Java que tiene los métodos necesarios para permitir la ejecución del comportamiento investigado.

Se elige, de la lista de aplicaciones clásicas de las pilas, el análisis de expresiones aritméticas.

Según el artículo “Java Program to Evaluate an Expression using Stacks” (2024), la evaluación de expresiones aritméticas es un problema fundamental en informática. Este problema se puede resolver eficientemente mediante las estructuras de datos de pila. La pila puede operar según el principio LIFO (último en entrar, primero en salir), lo cual resulta especialmente útil para analizar expresiones y garantizar la precedencia y asociatividad adecuadas de los operadores.

El concepto principal detrás del uso de las pilas para evaluar las expresiones aritméticas implica convertir la expresión a un formato que pueda procesarse fácilmente.

## Desarrollo

La pila es una estructura de datos adecuada para la implementación de análisis de expresiones aritméticas debido a que permite manipular expresiones que normalmente se representan en notación infija (el operador se coloca entre los operandos) mediante expresiones representadas en notación postfija (el operador se coloca después de los operandos).

Las expresiones aritméticas representadas en notación infija se analizan de izquierda a derecha, elemento por elemento. Esta tarea, sencilla para los humanos, tiene su equivalente computacional si se maneja cada elemento como un token de una pila. Con esta representación las computadoras pueden beneficiarse de la capacidad de analizar una expresión elemento por elemento.

| Notación infija | Pila equivalente |
|-----------------|------------------|
| 1 + 2 * 3       | 3                |
|                 | *                |
|                 | 2                |
|                 | +                |
|                 | 1                |

Para que una expresión aritmética en notación infijo se considere válida, debe cumplir con las siguientes reglas:

1. No debe haber un operador matemático al principio de la expresión.

| Expresión no válida |
|---------------------|
| / 1 + 2 * 3         |

2. No debe haber un operador matemático al final de la expresión.

| Expresión no válida |
|---------------------|
| 1 + 2 * 3 /         |

3. No debe haber dos operadores matemáticos consecutivos en la expresión.

| Expresión no válida |
|---------------------|
| 1 + + 2 * 3         |

4. No debe haber dos operandos matemáticos consecutivos en la expresión.

| Expresión no válida |
|---------------------|
| 1 2 * 3 + 2         |

Como se mencionó anteriormente, las expresiones aritméticas representadas en notación infija son sencillas de leer para los humanos, mientras que las computadoras utilizan expresiones aritméticas representadas en notación postfija debido a que facilita su análisis, es más eficiente de evaluar, elimina la necesidad de paréntesis y suprime la necesidad de las reglas de precedencia.

A continuación se muestra un ejemplo de una expresión aritmética en notación infija y su expresión equivalente en notación postfija:

| Notación infija | Notación postfija |
|-----------------|-------------------|
| 1 + 2 * 3       | 1 2 3 * +         |

Para efectos de este documento, se asume que el programa recibe una expresión aritmética con constantes numéricas positivas y negativas, con operadores matemáticos (suma, resta, multiplicación y división), con espacio en medio de los operadores y operandos, y sin paréntesis.

Evaluar una expresión aritmética en notación postfijo mediante una pila es computacionalmente simple y directo, ya que requiere solamente pasar una vez por toda la expresión. En contraposición a otros métodos más complejos que involucran parseo recursivo o árboles de sintaxis abstracta.

La evaluación de la expresión aritmética mediante el uso de pilas implica los siguientes pasos generales:

1. Convertir la expresión aritmética en notación infija a notación postfija

El programa recibe una expresión aritmética en notación infija almacenada en una variable tipo String y retorna una expresión aritmética en notación postfija almacenada en una variable tipo Pila.

| Entrada [String] | Salida [Pila] |
|------------------|---------------|
| 1 + 2 * 3        |               |
|                  | +             |
|                  | *             |
|                  | 3             |
|                  | 2             |
|                  | 1             |

2. Evaluar la expresión aritmética en notación postfija

El programa recibe una expresión aritmética en notación postfija almacenada en una variable tipo Pila y retorna el resultado almacenado en una variable tipo Pila.

| Entrada [Pila] | Salida [Pila] |
|----------------|---------------|
|                |               |
| +              | 7             |
| *              |               |
| 3              |               |

|   |  |
|---|--|
| 2 |  |
| 1 |  |
|   |  |

## Implementación

Ubicación del repositorio

<https://github.com/avargasgu-ucenfotec/analisis-expresiones-aritmeticas>

El programa cuenta con 3 implementaciones principales:

1. Convierte una expresión aritmética en notación infija almacenada en una variable tipo String en una expresión aritmética en notación infija almacenada en una variable tipo Pila. Para esto, se recorre la expresión de izquierda a derecha, se hace una copia de cada token y se inserta en la pila mediante el método "push".

| Paso | String      | Pila  |
|------|-------------|---|
| 1    | "1 + 2 * 3" | <div><div></div><div></div><div></div><div></div><div></div></div>  |
| 2    | "1 + 2 * 3" | <div><div></div><div></div><div></div><div></div><div>1</div></div> |
| 3    | "1 + 2 * 3" | <div><div></div><div></div><div></div></div>                        |



|   |             |  |
|---|-------------|--|
|   |             | <div>+</div> <div>1</div>  |
| 4 | "1 + 2 * 3" | <div></div> <div></div> <div>2</div> <div>+</div> <div>1</div>   |
| 5 | "1 + 2 * 3" | <div></div> <div>*</div> <div>2</div> <div>+</div> <div>1</div>  |
| 6 | "1 + 2 * 3" | <div>3</div> <div>*</div> <div>2</div> <div>+</div> <div>1</div> |

2. Valida ciertas condiciones que debe tener una expresión aritmética. Para esto, se recorre la pila evaluando si el elemento de arriba es un operador aritmético, si hay dos operadores aritméticos consecutivos en la pila, si hay dos operandos consecutivos en la pila, y si el último elemento es un operador

aritmético. Si se cumple alguna de las condiciones anteriores, se dice que la expresión de entrada no es válida.

| Paso   | Pila   |
|--|--|
| ¿Está la pila vacía?<br><b>No</b><br>¿El elemento en la cima es un operador?<br><b>No</b><br>¿Hay dos números consecutivos?<br><b>No</b> | <div>3</div> <div>*</div> <div>2</div> <div>+</div> <div>1</div> |
| ¿Está la pila vacía?<br><b>No</b><br>¿El elemento en la cima es un operador?<br><b>No</b><br>¿Hay dos números consecutivos?<br><b>No</b> | <div></div> <div>*</div> <div>2</div> <div>+</div> <div>1</div>  |
| ¿Está la pila vacía?<br><b>No</b><br>¿El elemento en la cima es un operador?<br><b>No</b><br>¿Hay dos números consecutivos?<br><b>No</b> | <div></div> <div></div> <div>2</div> <div>+</div> <div>1</div>   |
| ¿Está la pila vacía?<br><b>No</b><br>¿El elemento en la cima es un operador?<br><b>No</b><br>¿Hay dos números consecutivos?<br><b>No</b> | <div></div> <div></div> <div></div> <div>+</div> <div>1</div>    |

|  |   |
|--|---|
|  | 1 |
| ¿Está la pila vacía?<br><b>No</b><br>¿El elemento en la cima es un operador?<br><b>No</b><br>¿Hay dos números consecutivos?<br><b>No</b> | 1 |
| ¿Está la pila vacía?<br><b>Sí</b><br>Se concluye que la expresión ingresada es una expresión aritmética válida                           |   |

3. Evalúa una expresión aritmética en notación postfija almacenada en una variable tipo Pila. Para esto, se recorre la pila evaluando si el elemento es un número o un operador matemático. Dependiendo de lo anterior, se toman dos elementos de la pila y se ejecuta la aritmética, o se sigue recorriendo la pila.

| Paso  | Pila | Resultado |
|---|------|-----------|
| ¿Está la pila vacía?<br><b>No</b><br>¿El elemento en la cima es un operador?<br><b>No</b> | 1    |           |
|   | 2    |           |
|   | 3    |           |
|   | *    |           |
|   | +    | 1         |

|  |   |  |
|--|---|--|
|  |   |  |
| ¿Está la pila vacía?<br><b>No</b><br>¿El elemento en la cima<br>es un operador?<br><b>No</b> | <div></div> <div>2</div> <div>3</div> <div>*</div> <div>+</div> | <div></div> <div></div> <div></div> <div>2</div> <div>1</div>                      |
| ¿Está la pila vacía?<br><b>No</b><br>¿El elemento en la cima<br>es un operador?<br><b>No</b> | <div></div> <div></div> <div>3</div> <div>*</div> <div>+</div>  | <div></div> <div></div> <div>3</div> <div>2</div> <div>1</div>                     |
| ¿Está la pila vacía?<br><b>No</b><br>¿El elemento en la cima<br>es un operador?<br><b>Sí</b> | <div></div> <div></div> <div></div> <div>*</div> <div>+</div>   | <div></div> <div></div> <div></div> <div><math>3 * 2 = 6</math></div> <div>1</div> |
| ¿Está la pila vacía?<br><b>No</b><br>¿El elemento en la cima<br>es un operador?<br><b>Sí</b> | <div></div> <div></div> <div></div> <div></div> <div>+</div>    | <div></div> <div></div> <div></div> <div></div> <div><math>6 + 1 = 7</math></div>  |

|                            |  |   |
|----------------------------|--|---|
|                            |  |   |
| ¿Está la pila vacía?<br>Sí |  | 7 |

## Casos de prueba

1. El usuario ingresa una expresión aritmética válida

Análisis de expresiones aritméticas

Ingrese una expresión aritmética:  $1 + 2 * 3$

Menú

- [1] Mostrar la expresión como una pila de tokens
- [2] Analizar la validez de la expresión aritmética
- [3] Salir

Seleccione una opción del menú:  $1$

Pila almacenada

|     3   |

-----

|     \*   |

-----

|     2   |

-----

|     +   |

-----

|     1   |

-----

Menú

- [1] Mostrar la expresión como una pila de tokens
- [2] Analizar la validez de la expresión aritmética
- [3] Salir

Seleccione una opción del menú:  $2$

Expresión aritmética es válida.

El resultado de evaluar la expresión aritmética es:  $7$

2. El usuario ingresa una expresión aritmética no válida (tiene un operador matemático al inicio de la expresión)

Análisis de expresiones aritméticas

Ingresa una expresión aritmética:  $+ 1 * 2$

Menú

- [1] Mostrar la expresión como una pila de tokens
- [2] Analizar la validez de la expresión aritmética
- [3] Salir

Seleccione una opción del menú:  $1$

Pila almacenada

```
|      2 |  
-----  
|      * |  
-----  
|      1 |  
-----  
|      + |  
-----
```

Menú

- [1] Mostrar la expresión como una pila de tokens
- [2] Analizar la validez de la expresión aritmética
- [3] Salir

Seleccione una opción del menú:  $2$

Expresión aritmética no válida.

Existe un operador aritmético al inicio de la expresión ingresada.

3. El usuario ingresa una expresión aritmética no válida (tiene un operador matemático al final de la expresión)

Análisis de expresiones aritméticas

Ingresa una expresión aritmética:  $3 * 5 /$

Menú

- [1] Mostrar la expresión como una pila de tokens
- [2] Analizar la validez de la expresión aritmética
- [3] Salir

Seleccione una opción del menú:  $1$

Pila almacenada

```
|      / |  
-----  
|      5 |  
-----  
|      * |  
-----  
|      3 |  
-----
```

Menú

- [1] Mostrar la expresión como una pila de tokens
- [2] Analizar la validez de la expresión aritmética
- [3] Salir

Seleccione una opción del menú:  $2$

Expresión aritmética no válida.

Existe un operador aritmético al final de la expresión ingresada.



4. El usuario ingresa una expresión aritmética no válida (tiene dos operadores matemáticos de forma consecutiva)

```
Análisis de expresiones aritméticas
Ingrese una expresión aritmética: -2 * * 3

Menú
[1] Mostrar la expresión como una pila de tokens
[2] Analizar la validez de la expresión aritmética
[3] Salir
Seleccione una opción del menú: 1

Pila almacenada

|      3 |
|-----|
|      * |
|-----|
|      * |
|-----|
|     -2 |
|-----|

Menú
[1] Mostrar la expresión como una pila de tokens
[2] Analizar la validez de la expresión aritmética
[3] Salir
Seleccione una opción del menú: 2

Expresión aritmética no válida.
Existen dos operadores aritméticos consecutivos en la expresión ingresada.
```

5. El usuario ingresa una expresión aritmética no válida (tiene dos operandos de forma consecutiva)

Análisis de expresiones aritméticas

Ingrese una expresión aritmética:  $2\ 2\ +\ 3$

Menú

- [1] Mostrar la expresión como una pila de tokens
- [2] Analizar la validez de la expresión aritmética
- [3] Salir

Seleccione una opción del menú:  $1$

Pila almacenada

|      $3$    |

-----

|      $+$    |

-----

|      $2$    |

-----

|      $2$    |

-----

Menú

- [1] Mostrar la expresión como una pila de tokens
- [2] Analizar la validez de la expresión aritmética
- [3] Salir

Seleccione una opción del menú:  $2$

Expresión aritmética no válida.

Existen dos operandos consecutivos en la expresión ingresada.

### Referencias bibliográficas

*Java Program to Evaluate an Expression using Stacks*. (2024, Junio 27). GeeksforGeeks.

<https://www.geeksforgeeks.org/java/java-program-to-evaluate-an-expression-using-stacks/>

Blancarte, O. (2017, Junio 21). *Evaluador de expresiones postfijas (código fuente)*.

Oscar Blancarte Blog.

<https://www.oscarblancarteblog.com/2017/06/21/evaluador-expresiones-postfijas-codigo-fuente/>