



Universidad CENFOTEC
Ingeniería en Software

Curso: Estructuras de Datos

Profesor: Romario Salas Cerdas

Estudio de Caso #1

Estudiante: Abigail Mora Torres

III Cuatrimestre, 2025.

Las pilas son estructuras de datos caracterizadas por su modo de acceso tipo LIFO (last in, first out), o UEPS (último en entrar, primero en salir) justo como una pila de libros; para tener acceso al primer libro que fue ingresado a la pila, se necesita extraer todos los que están encima de él.

Este comportamiento resulta especialmente útil cuando se requiere un control jerárquico y reversible de las operaciones, como sucede en la evaluación de expresiones aritméticas o en la ejecución de rutinas anidadas dentro de un programa. Además, la pila facilita la organización temporal de datos que necesitan ser procesados en un orden inverso al de su llegada, proporcionando eficiencia y claridad en la gestión de operaciones complejas.

En el análisis de expresiones aritméticas, la pila desempeña un papel esencial al permitir procesar las operaciones respetando la precedencia y la agrupación definidas por los paréntesis. Una expresión aritmética combina operandos y operadores que deben evaluarse en un orden específico; pero este orden no siempre es lineal y puede variar según la jerarquía de los operadores.

Por ejemplo, en la expresión $(A+B)/4$, el operador de suma debe ejecutarse antes de la división, y la pila permite almacenar temporalmente los operadores y paréntesis hasta que puedan resolverse en el orden correcto asegurando que la evaluación se realice de manera coherente y confiable. Esta característica es crucial, ya que cualquier error en el manejo del orden de operaciones puede generar resultados incorrectos y afectar el funcionamiento global del software.

Según Aho, Lam, Sethi y Ullman (2008), las pilas son estructuras fundamentales en el análisis sintáctico de los compiladores, ya que permiten manejar el anidamiento de símbolos como paréntesis o delimitadores en las expresiones. En este sentido, la pila no solo ofrece orden y coherencia, sino que también proporciona seguridad en la ejecución: cada operación se realiza en el momento exacto y bajo las reglas de prioridad establecidas, lo que garantiza la estabilidad y consistencia de los procesos de evaluación.

Además, su eficiencia radica en que permite insertar y extraer elementos en tiempo constante, evitando la necesidad de recorrer toda la estructura para acceder a un dato específico.

La pila es la estructura ideal para el análisis de expresiones aritméticas porque ofrece orden, coherencia y reversibilidad, esto garantiza que cada operación se realice en el momento exacto y bajo las reglas de prioridad establecidas, manteniendo la estabilidad del proceso de evaluación.

En este proyecto, la funcionalidad de análisis de expresiones aritméticas se implementó mediante un programa en que utiliza una pila para verificar que los paréntesis de la expresión ingresada estén correctamente balanceados.

La pila actúa como estructura "último en entrar, primero en salir", permitiendo almacenar cada paréntesis de apertura a medida que se recorre la expresión. Cuando se encuentra un paréntesis de cierre, el programa extrae el último paréntesis apilado y verifica que coincida, garantizando que la expresión mantenga un orden correcto de apertura y cierre de símbolos.

Este enfoque permite brindar mensajes precisos al usuario: si los paréntesis están balanceados, el programa confirma la validez de la expresión; en caso contrario, muestra el mensaje `"Expresión inválida (paréntesis desbalanceados)."`.

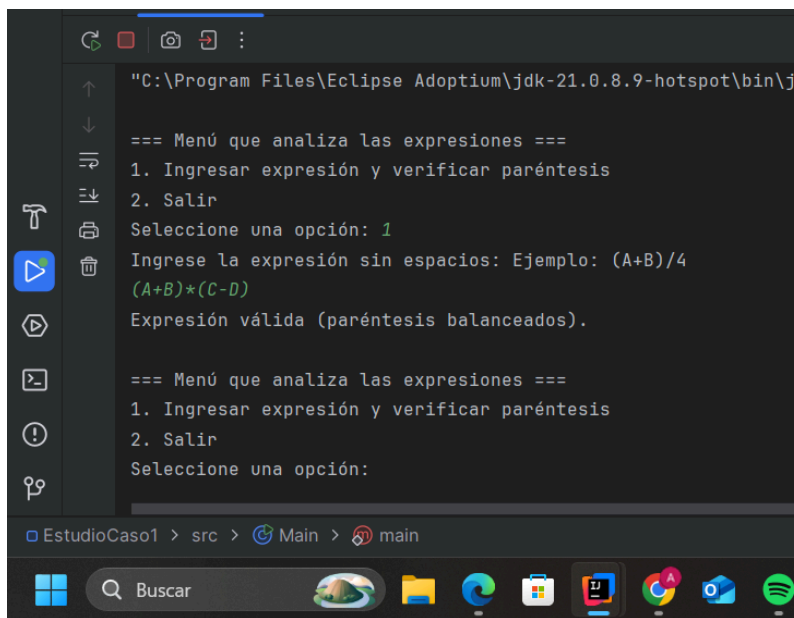
El software está diseñado para que el usuario pueda ingresar distintas expresiones aritméticas a través de un menú interactivo, mostrando de manera inmediata si la expresión es válida o no según el balanceo de paréntesis.

Este ejemplo demuestra cómo las pilas se aplican en sistemas reales de software, especialmente en compiladores o interpretadores, donde es fundamental controlar la correcta estructura de las expresiones y garantizar que no existan errores de sintaxis relacionados con símbolos anidados.

Finalmente, la implementación de pilas en el análisis de expresiones aritméticas demuestra cómo una estructura de datos aparentemente sencilla puede ser crucial para garantizar orden, seguridad y eficiencia en sistemas de software.

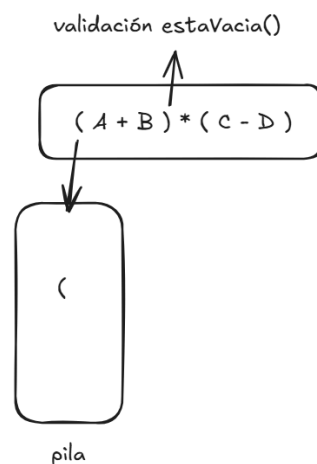
Su uso no solo facilita la evaluación correcta de expresiones, sino que también contribuye a la robustez general del programa, minimizando errores y optimizando la ejecución de operaciones complejas.

La Figura 1 muestra un ejemplo de ejecución del programa, donde la expresión $(A+B)*(C-D)$ es analizada y verificada como balanceada.



```
"C:\Program Files\Eclipse Adoptium\jdk-21.0.8-hotspot\bin\java.exe"  
  
=== Menú que analiza las expresiones ===  
1. Ingresar expresión y verificar paréntesis  
2. Salir  
Seleccione una opción: 1  
Ingrese la expresión sin espacios: Ejemplo: (A+B)/4  
(A+B)*(C-D)  
Expresión válida (paréntesis balanceados).  
  
=== Menú que analiza las expresiones ===  
1. Ingresar expresión y verificar paréntesis  
2. Salir  
Seleccione una opción:
```

La figura 2 muestra la representación gráfica de la pila



Bibliografía

Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2008). *Compiladores: principios, técnicas y herramientas* (2.^a ed.). Pearson Educación.

Recuperado de:

<https://books.google.com.gt/books?id=yG6qJBAnE9UC&printsec=frontcover#v=onepage&q&f=false>