Proyecto 2. Algoritmos y Estructuras de Datos I

Profesor: Leonardo Araya Martínez

1st Duarte Astua Jose Julian
Área académica Ingeniería en Computadores
Instituto Tecnológico de Costa Rica
Costa Rica
josduarte@estudiantec.cr

2nd Vega Marín Giancarlo
Área académica Ingeniería en Computadores
Instituto Tecnológico de Costa Rica
Costa Rica
jungianca6@estudiantec.cr

***Abstract*—El proyecto se centra en implementar una calculadora que evalue expresiones matemáticas de cualquier longitud utilizando arboles de expresión binaria. Se busca incorporar funcionalidades como operaciones algebraicas y lógicas, así como el reconocimiento de expresiones impresas a través de una cámara web.**

***Index Terms*—component, formatting, style, styling, insert**

I. Descripción del problema

Este proyecto consiste en construir una calculadora que evalúa expresiones de longitud arbitraria. Con ese fin se utilizará un árbol de expresión binaria. La calculadora realizará operaciones algebraicas simples (+, -, *, /, **), así como operaciones lógicas (and, or, not, xor) de cualquier longitud, colocando la expresión en un árbol de expresiones binarias y luego evaluando el árbol de expresiones.

Un árbol de expresión binaria es un árbol binario, que tiene como máximo dos hijos. Recuerde que existen dos tipos de nodos en un árbol binario, los nodos hoja que no tienen hijos y los nodos internos que tienen uno o más hijos (y forman el cuerpo de el árbol). En un árbol de expresión binaria, los nodos internos contendrán los operadores de la expresión (+, -, *, /,

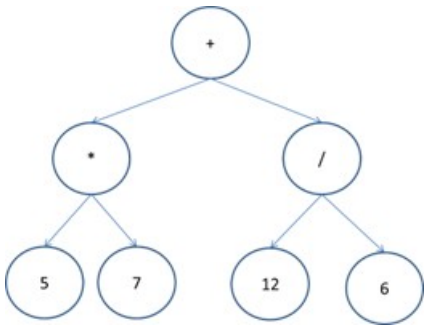

Fig. 1. Figura 1. Árbol de expresión binaria

El árbol en la Figura 1 representa la expresión (5 * 7) + (12/6) y el resultado de su evaluación es 37. Mientras que la expresión representada por el árbol de la Figura 2 es (5 * (10 - 15)) + 7 y su resultado es -18.

El proceso para evaluar árboles de expresión es recursivo. Primero evalúa el subárbol izquierdo, luego se evalúa el subárbol derecho y finalmente combina las dos soluciones usando el operador en el nodo.

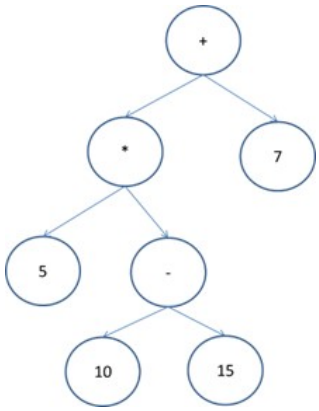

Fig. 2. Figura 2. Árbol de la expresión (5 * (10 - 15)) + 7.

El algoritmo para construir un árbol de expresión requiere utilizar la notación postfija (también conocida como notación polaca inversa, una versión modificada de una notación matemática inventada por matemáticos polacos a principios del siglo XX). La notación de sufijo se usa ampliamente en los círculos de computación porque las expresiones anotadas en notación de sufijo son completamente inequívocas sin tener que recurrir a paréntesis.

II. Diagrama de clases

El diagrama de clases refleja la organizacion estructurada del sistema. Las clases clave incluyen Cliente, Servidor, Arbol de Expresiones,etc.

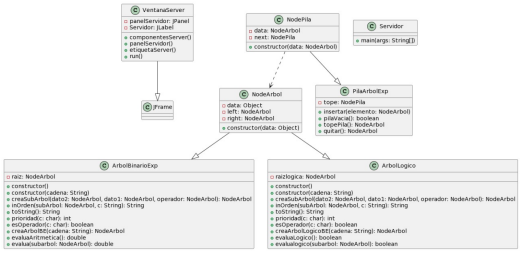

Fig. 3. Diagrama de servidor

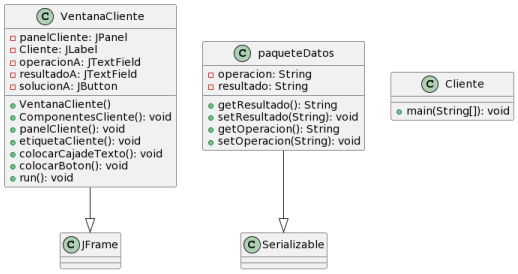

Fig. 4. Diagrama de Cliente

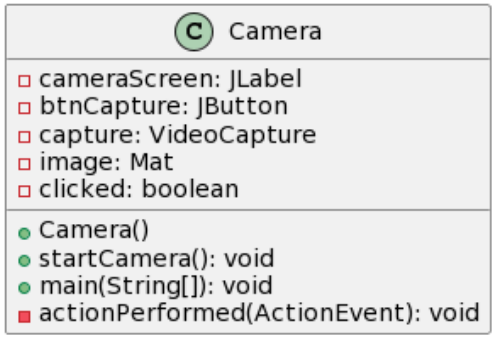

Fig. 5. Diagrama de Cliente

III. Descripcion de las Estructuras de Datos Desarrolladas

Árbol de Expresión Binaria:

Descripción: El árbol de expresión binaria es una estructura jerárquica que representa la estructura de una expresión matemática. Cada nodo del árbol contiene un operador o un operando, y los nodos están organizados de manera que reflejan la jerarquía de la expresión. Implementación: La clase ArbolBinarioExp implementa esta estructura. Los nodos internos contienen operadores, mientras que los nodos hoja contienen operandos.

Pila para Evaluación de Expresiones:

Descripción: Se utiliza una pila para evaluar expresiones de manera eficiente mediante un enfoque de evaluación de expresiones en notación postfija. Los operandos y operadores se apilan y desapilan según el orden de evaluación.

Implementación: La clase PilaArbolExp representa la pila, y la estructura NodePila define los nodos de la pila.

Nodo de Árbol y Nodo de Pila:

Descripción: Estos nodos son las unidades básicas que forman los árboles y las pilas. Los nodos de árbol contienen datos (operandos u operadores) y enlaces a los nodos hijos izquierdo y derecho en el caso del árbol de expresión binaria.

Implementacion: Las clases NodeArbol y NodePila definen la estructura de estos nodos, y se utilizan en la construcción del árbol de expresión y la pila, respectivamente.

Registro de Operaciones en Archivo CSV:

Descripción: Para mantener un registro de todas las operaciones realizadas, se utiliza un archivo CSV. Cada operación se registra con detalles como la expresión, el resultado y la fecha.

Implementación: El registro de operaciones se gestiona mediante la escritura y lectura de datos en un archivo CSV. Se utiliza la biblioteca opencsv para interactuar con archivos CSV, y la lógica de registro se encuentra en el método run de la clase VentanaServer.

Estas estructuras de datos proporcionan una base sólida para la manipulación y evaluación de expresiones matemáticas, garantizando un flujo eficiente de datos y operaciones dentro de la aplicación. La implementación adecuada de estas estructuras es crucial para el funcionamiento correcto y eficiente de la calculadora y su capacidad para gestionar múltiples operaciones simultáneamente.

IV. Problemas encontrados en forma de bugs de GitHub

A la hora de querer implementar una solución al código que permitiera trabajar con números negativos, causó que los números negativos siempre deben ir encerrados en un paréntesis. Por lo tanto, una resta como ”6-3” no se resuelve bien; sin embargo, si se encierra en un paréntesis y se cambia el operador por ”6+(-3)”, la operación si funciona, y se obtiene el resultado correcto.