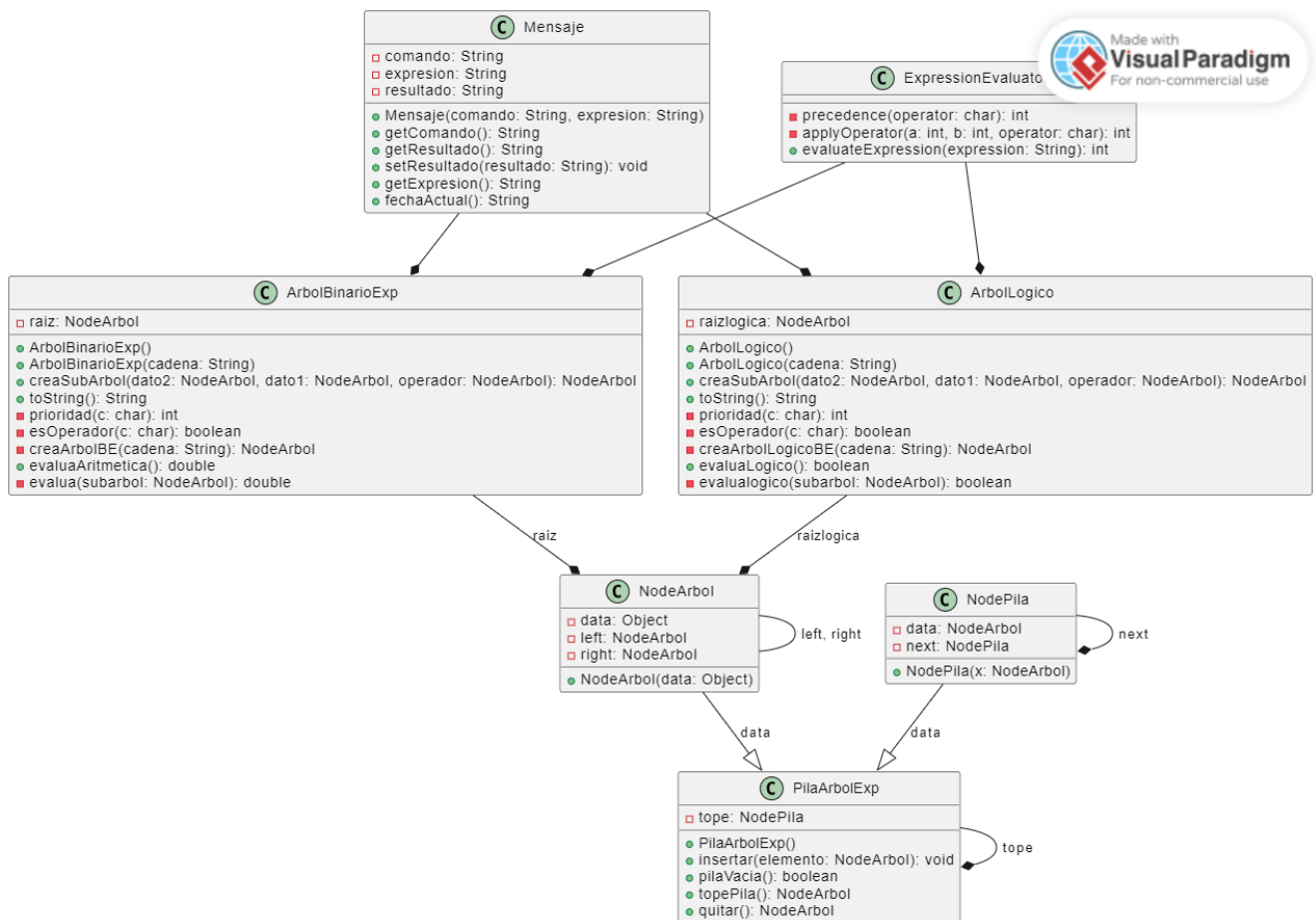


Documentación requerida

1. Descripción breve del problema

Este proyecto implica la creación de una aplicación cliente-servidor que permite a los usuarios ingresar y calcular expresiones matemáticas y lógicas de cualquier longitud. La aplicación utiliza árboles de expresión binaria para evaluar estas expresiones de manera estructurada. El cliente proporciona una interfaz web para ingresar expresiones y reconocer expresiones impresas con una cámara web, mientras que el servidor realiza los cálculos y mantiene un registro de las operaciones realizadas. Además, el sistema debe ser capaz de manejar múltiples usuarios de manera simultánea.

2. Diagrama de Clases



3. Descripción de las estructuras de datos desarrolladas.

Clase Mensaje

Descripción: La clase Mensaje representa un mensaje que contiene información sobre un comando y una expresión. Está diseñada para facilitar la comunicación y el transporte de información en el programa.

Atributos:

comando: Almacena el comando del mensaje, que podría ser una instrucción para realizar una operación específica.

expresión: Contiene la expresión relacionada con el comando, como una operación matemática o lógica.

resultado: Opcionalmente, puede almacenar el resultado de la operación realizada en base al comando.

Función: El propósito principal de esta clase es encapsular información relacionada con comandos y expresiones para su posterior procesamiento en el programa. Puede ser utilizada para llevar un registro de las operaciones realizadas y sus resultados.

Clase ExpressionEvaluator

Descripción: La clase **ExpressionEvaluator** se encarga de evaluar expresiones matemáticas. Utiliza un algoritmo que convierte expresiones en notación infija en notación posfija y, posteriormente, calcula su resultado.

Métodos:

evaluateExpression (expression: String): Evalúa una expresión matemática en notación infija y devuelve su resultado como un entero.

Función: Esta clase proporciona una funcionalidad esencial para calcular el resultado de expresiones matemáticas ingresadas por los usuarios. Transforma la expresión en un formato adecuado para su evaluación y luego realiza los cálculos necesarios.

Clase NodeArbol

Descripción: La clase **NodeArbol** define los nodos que conforman un árbol de expresiones. Cada nodo puede contener un operador o un valor, y tiene dos hijos (izquierdo y derecho) que representan

Atributos:

data: Almacena un objeto (puede ser un operador o un valor).

left: Representa el hijo izquierdo del nodo.

right: Representa el hijo derecho del nodo. la estructura de una expresión.

Función: Estos nodos se utilizan para construir árboles de expresiones, lo que facilita la evaluación de expresiones matemáticas y lógicas. Cada nodo almacena información sobre la operación u operando en ese punto del árbol.

Clase NodePila

Descripción: La clase **NodePila** define los nodos utilizados en una pila de nodos de árbol. Cada nodo de esta pila almacena un nodo del árbol de expresiones.

Atributos:

data: Almacena un objeto de tipo NodeArbol.

next: Representa el siguiente nodo en la pila.

Función: Estos nodos se utilizan en la construcción y evaluación de árboles de expresiones. Ayudan a mantener un seguimiento de los nodos del árbol a medida que se construye o evalúa.

Clase PilaArbolExp

Descripción: La clase **PilaArbolExp** implementa una pila de nodos de árbol. Se utiliza para gestionar la construcción y evaluación de árboles de expresiones.

Métodos:

insertar (elemento: NodeArbol): Agrega elementos a la parte superior de la pila.

pilaVacía(): Verifica si la pila está vacía.

topePila(): Obtiene el primer elemento de la pila.

quitar(): Elimina el elemento superior de la pila y lo devuelve.

Función: La pila de nodos de árbol permite un acceso eficiente a los nodos del árbol durante la construcción y evaluación de expresiones. Facilita la organización de los nodos y la gestión de su orden.

Clase ArbolBinarioExp

Descripción: La clase **ArbolBinarioExp** se encarga de la construcción y evaluación de árboles de expresiones matemáticas. Puede convertir una expresión matemática en notación infija en un árbol y calcular su resultado.

Atributos:

raiz: Representa la raíz del árbol.

Métodos:

ArbolBinarioExp(): Constructor para un árbol vacío.

ArbolBinarioExp (cadena: String): Constructor que crea un árbol a partir de una expresión en notación infija.

creaSubArbol(dato2: NodeArbol, dato1: NodeArbol, operador: NodeArbol): Crea un subárbol con dos nodos hijos y un operador.

evaluaAritmetica(): Evalúa el árbol y devuelve el resultado de la expresión matemática.

Función: Esta clase es fundamental para evaluar expresiones matemáticas complejas. Construye una estructura de árbol que refleja la jerarquía de operaciones en la expresión, lo que facilita su evaluación.

Clase ArbolLogico

Descripción: La clase **ArbolLogico** se utiliza para construir y evaluar árboles de expresiones lógicas. Puede convertir expresiones lógicas en notación infija en un árbol y calcular su resultado.

Función: Esta clase es esencial para la evaluación de expresiones lógicas, como expresiones booleanas. Al igual que **ArbolBinarioExp**, permite la construcción de una estructura de árbol que refleja la estructura de la expresión.

Atributos:

raizlogica: Representa la raíz del árbol.

Métodos:

ArbolLogico(): Constructor para un árbol lógico vacío.

ArbolLogico (cadena: String): Constructor que crea un árbol a partir de una expresión lógica en notación infija.

creaSubArbol (dato2: NodeArbol, dato1: NodeArbol, operador: NodeArbol): Crea un subárbol lógico con dos nodos hijos y un operador.

evaluaLogico (): Evalúa el árbol y devuelve el resultado de la expresión lógica.

Problemas encontrados en forma de bugs de github: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo

Como primer punto no se pudo hacer uso de la funcionalidad cuando la cámara toma una fotografía ya que la cámara de la computadora no cuenta con la suficiente definición para captar una imagen con buena resolución y así el resto del programa pueda resolver la operación a la cual se le toma foto.

4. Planificación y administración del proyecto: se utilizará Azure DevOps para la administración del proyecto.

Este apartado se responde por medio del link donde se administro las tareas de cada desarrollador del proyecto, así como las correspondientes asignación y características de las tareas a realizar por el equipo.

<https://dev.azure.com/JosephProjects01/CalculadorTree>