

1. Problema

Hacer una calculadora que procese operaciones simples utilizando la estructura de datos Pila.

2. Código

2.1. Pila.h

```
#ifndef PILA.H
#define PILA.H

template<typename T>
class Pila
{
public:
    Pila();
    class Nodo{
    public:
        Nodo();
        Nodo(T dato);prosece
        T dato;
        Nodo *siguiente;
    private:
    };
    void agregar(T dato);
    bool esVacia();
    T desapilar();
protected:
private:
    Nodo * cabeza;
};

template<typename T>
Pila<T>::Nodo::Nodo(){
    siguiente = nullptr;
}
template<typename T>
Pila<T>::Nodo::Nodo(T dato){
    this->dato = dato;
}
template<typename T>
Pila<T>::Pila(){
    cabeza = nullptr;
}
template<typename T>
void Pila<T>::agregar(T dato){
    Pila<T>::Nodo * nuevo = new Nodo(dato);
    if(cabeza == nullptr){
        cabeza = nuevo;
    }
    else{
```

```

        nuevo->siguiente = cabeza;
        cabeza = nuevo;
    }
}
template<typename T>
bool Pila<T>::esVacia(){
    if(cabeza == nullptr)
        return true;
    return false;
}
template<typename T>
T Pila<T>::desapilar(){
    if(esVacia()){
        return 0;
    }
    else{
        T result = cabeza->dato;
        cabeza = cabeza->siguiente;
        return result;
    }
}
}

#endif // PILA_H

```

2.2. main.cpp

```

#include <iostream>
#include <Pila.h>
#include <math.h>

using namespace std;

float convertirNumero(string numero){
    double resultado = 0;
    auto iter = numero.end();
    iter--;
    double contador = 0;
    for(iter; iter!= numero.begin(); iter--){
        if((*iter) == '.'){
            resultado /= pow(10,contador);
            contador = -1;
        }
        else{
            resultado += pow(10,contador) * ((*iter) - 48);
        }
        contador++;
    }
    resultado += pow(10,contador) * ((*iter) - 48);
    return resultado;
}

int main()
{
    Pila<float> memoria;
    string expresion;
    string temp;

```

```

cin>>expression;
for(int i = 0; i < expression.size(); i++){
    if(expression[i] == '|'){
        memoria.agregar(convertirNumero(temp));
        temp.clear();
    }
    else if((expression[i] >= 48 and expression[i] <= 57)
        or expression[i] == '.'){
        temp.insert(temp.end(), expression[i]);
    }
    else{
        switch(expression[i]){
            case '+':
                memoria.agregar(memoria.desapilar()
                                + convertirNumero(temp));
                temp.clear();
                break;
            case '-':
                memoria.agregar(memoria.desapilar()
                                - convertirNumero(temp));
                temp.clear();
                break;
            case '*':
                memoria.agregar(memoria.desapilar()
                                * convertirNumero(temp));
                temp.clear();
                break;
            case '/':
                memoria.agregar(memoria.desapilar()
                                / convertirNumero(temp));
                temp.clear();
                break;
            default:
                cout<<"El caracter "<<expression[i]
                    <<" no se puede reconocer";
                return 0;
        }
    }
}
cout<<memoria.desapilar();
}

```

3. Ejemplo

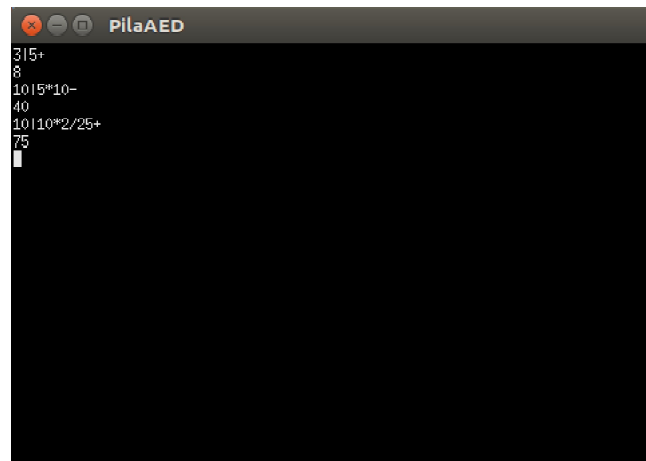


Figura 1: EJemplo