

# RELATÓRIO - TRABALHO 1

**NOME:** Natalia Oliveira Borges

**MATRICULA:** 16/0015863

**DISCIPLINA:** Estrutura de Dados

## I. Notação Polonesa Reversa:

A Notação Polonesa Reversa, ou notação posfixa, é uma forma de escrever uma expressão matemática em que o sinal da operação vem depois dos operandos, diferente da notação usual em que o operador vem entre os operandos.

Essa notação tem certas vantagens sobre a usual quando se analisa a forma de implementação nesse trabalho. Como as operações são efetuadas na ordem em que são colocadas, independente de prioridade de operadores, não há necessidade de inicializadores de escopos, reduzindo assim a complexidade de calcular a expressão.

Uma expressão na forma posfixa é resolvida percorrendo a expressão até encontrar um operador, quando o operador é encontrado, os dois operandos que o antecedem realizam a operação determinada e o resultado dessa operação substitui o conjunto operandos operador na expressão. Dessa forma a expressão é percorrida até que todos os operadores sejam analisados. O número que resta no final é o resultado da expressão.

Exemplo: A expressão  $2 * 4 - 5 * (10 + 40/4)$  poderia ser resolvida seguinte forma:

1. Passando para notação pós-fixa:  $2\ 4\ *\ 5\ 10\ 40\ 4\ /\ +\ *\ -$
2. Percorrendo a expressão até encontrar um operador: Operador encontrado (\*) operandos anteriores (2 e 4).
3. O resultado da operação  $2*4=8$  substitui (2 4 \*) na expressão:  $8\ 5\ 10\ 40\ 4\ /\ +\ *\ -$

Seguindo o mesmo processo:

4. Operador ( / ) operandos (40 4):  $40/4 = 10$ . Expressão:  $8\ 5\ 10\ 10\ +\ *\ -$
5. Operador ( + ) operandos (10 10):  $10+10 = 20$ . Expressão:  $8\ 5\ 20\ *\ -$
6. Operador ( \* ) operandos (5 20):  $5*20 = 100$ . Expressão  $8\ 100\ -$
7. Operador ( - ) operandos (8 100):  $8-100=-92$ . Expressão -92

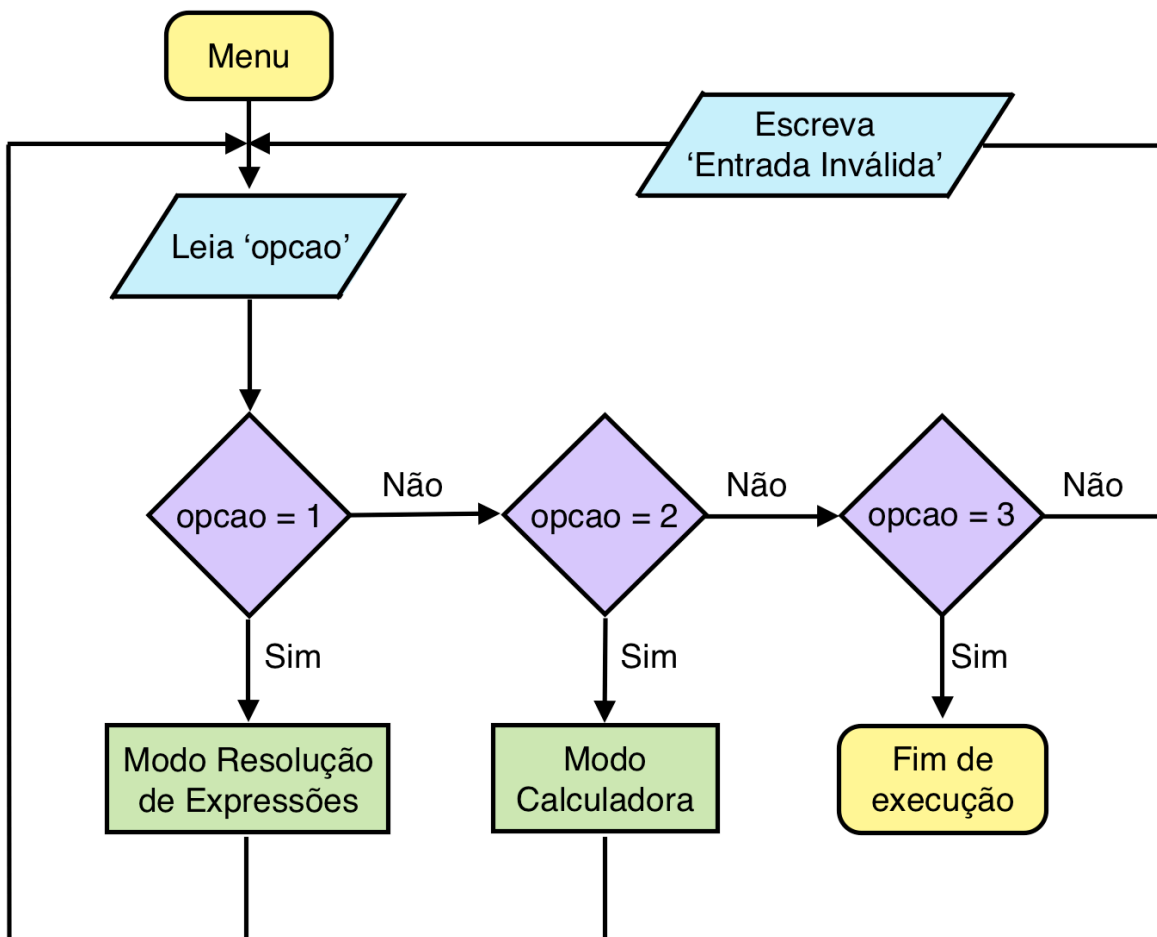
Não há mais operadores, logo, o resultado da expressão é -92.

## II. Arquitetura do Sistema

Para exemplificar a arquitetura do sistema serão apresentados fluxogramas de todo o processo.

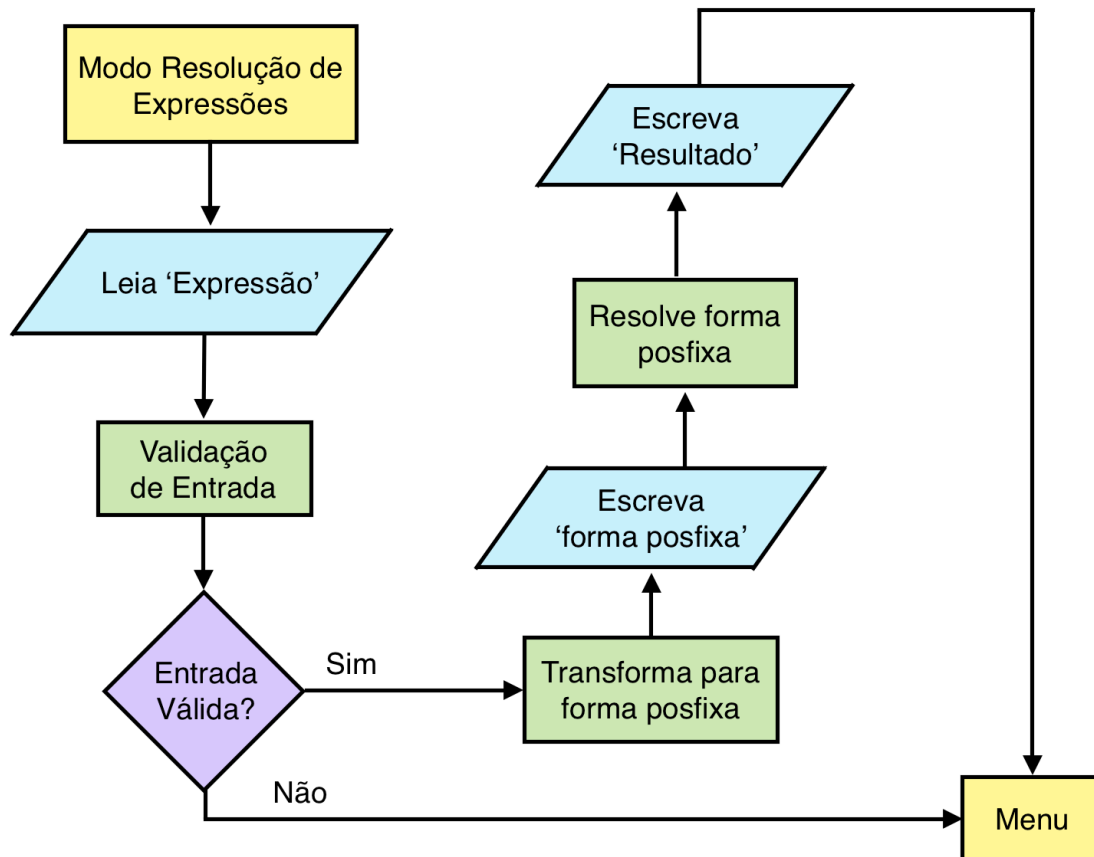
## 1. Funcionamento do menu:

Ao começar o programa, o menu é exibido. Em seguida, o usuário deve fornecer uma entrada selecionando a opção desejada. Se selecionar a opção '1', entra no modo resolução de expressões. Se selecionar a opção '2', entra no modo calculadora. Ao terminar a execução dos modos, o menu é reapresentado e é possível navegar pelas opções até que seja escolhida a opção de saída. Se selecionar a opção '3', termina a execução do programa. Se outra entrada for colocada, uma mensagem de erro aparece e novamente uma entrada é requisitada.



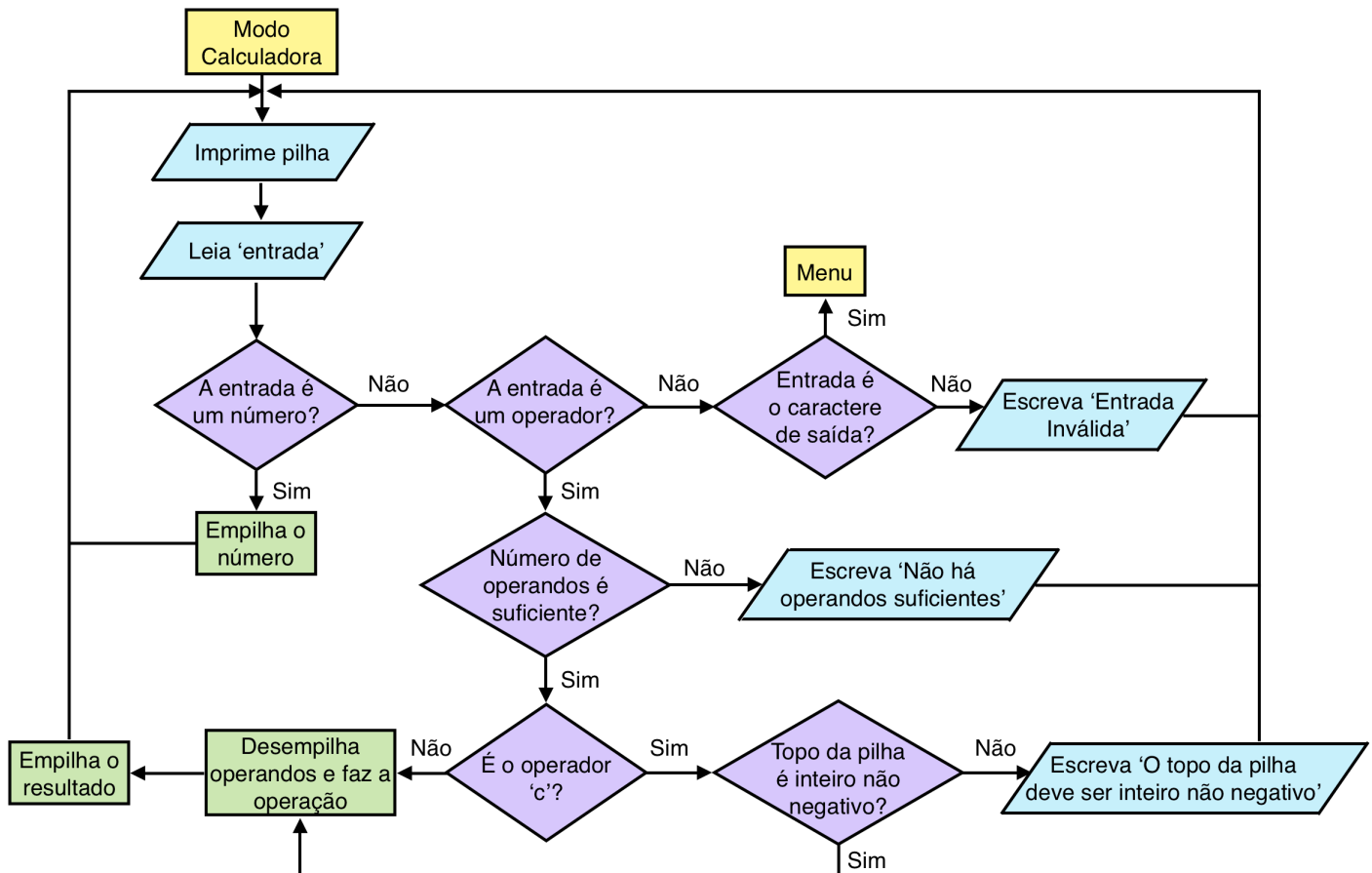
## 2. Funcionamento do modo resolução de expressão:

Ao entrar no modo resolução de expressão, uma entrada é requisitada, essa entrada deve ser a expressão que se deseja resolver. Após inserir a expressão, ela é validada. A validação leva em conta se a expressão está sintaticamente correta e confere se todos os inicializadores de escopo foram devidamente fechados na ordem correta. Se a expressão dada não for válida, volta-se ao menu do programa. Se a expressão for válida, transforma a expressão infixa para posfixa e imprime na tela a expressão posfixa. Finalmente, a expressão posfixa é solucionada e o resultado também é impresso na tela.



### 3. Funcionamento do modo calculadora:

Ao entrar no modo calculadora, a pilha é impressa, inicialmente vazia, deve-se então receber uma entrada. Se a entrada for um número, o número é empilhado. Se a entrada for um operador, verifica-se se o número de operandos na pilha é suficiente para realizar a operação. Se não for suficiente, uma mensagem de erro aparece. Se o número de operandos for suficiente, verifica se o operador é o operador 'c', se for é necessário checar se o topo da pilha é inteiro não negativo. Se o topo não for válido, outra mensagem de erro aparece. Caso o operador não seja o 'c' ou os requisitos do operador 'c' sejam atendidos, os operandos são desempilhados e a operação é realizada, em seguida o resultado é empilhado na pilha. Se a entrada for o caractere de saída, a execução volta para o menu do programa. Caso a entrada não seja número, operador ou caractere de saída, ela é inválida e uma mensagem de erro aparece. Enquanto o caractere de saída não for digitado, a execução sempre volta a imprimir a pilha e esperar uma nova entrada.



### III. Funcionalidades

#### 1. Resolução de expressões:

No modo resolução de expressões deve-se fornecer um expressão em notação infixa, com o objetivo de chegar ao seu resultado.

#### MODO RESOLUÇÃO DE EXPRESSÕES

Digite a expressão:

Após fornecer a expressão, ela é validada. A validação leva em conta se a expressão está sintaticamente correta e se todos os inicializadores de escopos que foram abertos também foram fechados.

Se a expressão fornecida for válida, o programa transforma essa expressão infixa em posfixa.

#### MODO RESOLUÇÃO DE EXPRESSÕES

Digite a expressão:

15+{[4,5/3]-(1,5-5)}

A expressão 15+{[4,5/3]-(1,5-5)} é válida.

Transformando essa expressão para posfixa:

15 4,5 3 / 1,5 5 - - +

Por fim, usando as regras de operação da notação polonesa reversa, a expressão posfixa é resolvida e o resultado é mostrado.

```
0 resultado dessa expressão é:
20
Pressione 'Enter' para voltar ao menu.
```

Se a expressão fornecida não for válida, uma mensagem de erro aparece e o programa espera o usuário digitar a tecla 'Enter' para voltar ao menu.

```
MODULO RESOLUÇÃO DE EXPRESSÕES

Digite a expressão:
1+3*(6+7
A expressão 1+3*(6+7 não é válida.

Pressione 'Enter' para voltar ao menu.
```

## 2. Calculadora

O modo calculadora também simula o comportamento da notação polonesa reversa.

O programa inicia com uma pilha vazia e deve-se preenche-la com os operandos desejados. A pilha aceita números positivos, negativos, inteiros ou pontos flutuantes.

```
Para sair, digite: 'X'.

MODULO CALCULADORA

Pilha Vazia.
->
```

Uma vez que a pilha tenha o numero necessário de operandos é possível fazer algumas operações. Vale ressaltar que a pilha aceita números positivos e negativos, inteiros ou pontos flutuantes.

Usando a seguinte pilha como exemplo.

```
MODULO CALCULADORA

3. 2
2. 4
1. 3
->
```

Funcionamento das operações:

Operações	
+	Adiciona os dois últimos números. (4+3)
-	Faz a diferença entre os dois últimos números. (4-3)
*	Multiplica os dois últimos números. (4*3)
/	Divide o penúltimo número pelo último (4/3)

Operações	
+	Repete a operação de soma para todos os elementos da pilha (2+(4+3))
-	Repete a operação de diferença para todos os elementos da pilha (2-(4-3))
*	Repete a operação de multiplicação para todos os elementos da pilha. (2*(4*3))
/	Repete a operação de divisão para todos os elementos da pilha. (2/(4/3))
c	Copia o penúltimo número quantas vezes for informado no último número. (4 4 4)

### Exemplos:

Operador (-)

```

MOD0 CALCULADORA

3. 1
2. 12
1. 5
->-|

```

->

```

MOD0 CALCULADORA

2. 1
1. 7
->|

```

Operador (\*!)

```

MOD0 CALCULADORA

3. 2
2. 3,500000
1. 7
->*!|

```

->

```

MOD0 CALCULADORA

1. 49
->|

```

Operador (c)

```

MOD0 CALCULADORA

3. 2
2. 5
1. 3
->c|

```

->

```

MOD0 CALCULADORA

4. 2
3. 5
2. 5
1. 5
->|

```

### Entradas inválidas:

- Número de operandos insuficientes.

```

MOD0 CALCULADORA

1. 4
->*|

```

Mensagem:

```

Não há elementos suficientes na pilha.

```

- Combinação de caracteres inválida.

MOD0 CALCULADORA

```
2. 2
1. 1,250000
->bed
```

Mensagem:

Entrada Inválida.

- Tentar realizar a operação 'c' com o topo da pilha sendo negativo ou não inteiro.

MOD0 CALCULADORA

```
3. 2
2. 4
1. 5,530000
->c
```

Mensagem:

Para usar o comando 'c' o topo da pilha precisa ser inteiro não negativo.

#### Observação:

Para sair do modo calculadora, usa-se os caracteres 'x' ou 'X'.

## IV. Smoke Test

### 1. Suite de teste 1 (Testes Menu)

- Apresentação do menu  
Arquivo - main.c

Ação	Expectativa	Resultado	Obteve sucesso
Abrir o programa e aguardar exibição do menu.	Menu deve exibir corretamente todas as opções.	Menu exibe todas as opções corretamente.	Sim

- Navegação  
Arquivo - main.c

Ação	Expectativa	Resultado	Obteve sucesso
Digitar '1'.	Entrar no modo resolução de expressões.	Entra no modo resolução de expressões.	Sim
Digitar '2'.	Entrar no modo calculadora.	Entra no modo calculadora.	Sim
Digitar '3'.	Sair do programa	Sai do programa	Sim

## 2. Suite de teste 2 (Teste do Modo Resolução de Expressões)

- Validação de entrada  
Arquivo - calculadora.c

Ação	Expectativa	Resultado	Obteve sucesso
$2 + 3$	Deve validar expressão com números inteiros.	Expressão válida	Sim
$2,7 * 4,5 - 2$	Deve validar expressão com números de ponto flutuante.	Expressão válida	Sim
$-4 - 2$	Deve validar expressão com números negativos.	Expressão válida	Sim
$-0,25 / 0,125$	Deve validar expressão com números de ponto flutuante negativos.	Expressão válida	Sim
$(34 - 2,3) * [5 / 2]$	Deve validar expressão com uso correto de inicializadores de escopo.	Expressão válida	Sim
$\{2 * 5 / (4)\}$	Deve invalidar expressão com uso incorreto de inicializadores de escopo.	Expressão inválida	Sim
$-4 * -6$	Deve invalidar expressão com operador seguido de operador. Para colocar números negativos no meio da expressão é necessário o uso de parênteses.	Expressão inválida	Sim
$4,5 8 + 2$	Deve invalidar expressão com número seguido de número sem um operador entre eles.	Expressão inválida	Sim
$2 + \text{bed}$	Deve invalidar expressão com caracteres que não sejam numéricos, operadores ou inicializadores de escopo	Expressão inválida	Sim

- Resolução da expressão:

Ação	Expectativa	Resultado	Obteve sucesso
$2 + 3$	Exibir a expressão na notação posfixa e o resultado	$2 3 + = 5$	Sim
$2,7 * 4,5 - 2$	Exibir a expressão na notação posfixa e o resultado	$2.7 4.5 * 2 - = 10,150000$	Sim



Ação	Expectativa	Resultado	Obteve sucesso
-4 - 2	Exibir a expressão na notação posfixa e o resultado	-4 2 - = -6	Sim
-0,25 / 0,125	Exibir a expressão na notação posfixa e o resultado	-1,25 0,125 / = -2	Sim
(34 - 2,3) * [5 / 2]	Exibir a expressão na notação posfixa e o resultado	34 2,3 - 5 2 / * =79,250000	Sim

Arquivo - calculadora.c

### 3. Suite de teste 3 (Teste do Modo Calculadora)

Arquivo - calculadora.c

Ação (->)	Pilha Antes	Expectativa	Resultado	Pilha Depois	Obteve sucesso
2	Pilha Vazia	Adicionar número inteiro na pilha.	Adicionou o número.	1. 2	Sim
-2	1. 2	Adicionar número inteiro negativo na pilha.	Adicionou o número.	2. 2 1. -2	Sim
1,35	2. 2 1. -2	Adicionar número ponto flutuante na pilha.	Adicionou o número.	3. 2 2. -2 1. 1,350000	Sim
-4,5	3. 2 2. -2 1. 1,350000	Adicionar número ponto flutuante negativo na pilha.	Adicionou o número.	4. 2 3. -2 2. 1,35000 1. -4,500000	Sim
+	2. 4 1. 5	Adicionar os dois últimos operandos da pilha.	Realizou operação.	1. 9	Sim
-	2. 4,500000 1. 0,500000	Fazer a diferença entre os dois últimos operandos da pilha.	Realizou operação.	1. 4	Sim
*	2. 4 1. 7	Multiplicar os dois últimos operandos.	Realizou operação.	1. 28	Sim
/	2. 4 1. 3	Dividir os dois últimos operandos.	Realizou operação.	1. 1,333333	Sim
+!	3. 1 2. 5 1. -2	Repetir operação '+' até sobrar um elemento na pilha.	Realizou operação.	1. 4	Sim

Ação (->)	Pilha Antes	Expectativa	Resultado	Pilha Depois	Obteve sucesso
-!	4. 2 3. 12 2. 3,500000 1. 1,500000	Repetir operação '-' até sobrar um elemento na pilha.	Realizou operação.	1. -8	Sim
*!	3. -8 2. 3 1. 0,125000	Repetir operação '*' até sobrar um elemento na pilha.	Realizou operação.	1. -3	Sim
/!	3. 1 2. 2 1. 4	Repetir operação '/' até sobrar um elemento na pilha.	Realizou operação.	1. 2	Sim
c	3. 2 2. 1 1. 4	Repetir o penúltimo número a quantidade informada no topo da pilha.	Realizou operação.	5. 2 4. 1 3. 1 2. 1 1. 1	Sim
+	1. 2	Informar ao usuário que não há números suficientes na pilha.	"Não há elementos suficientes na pilha".	1. 2	Sim
c	2. 6 1. 0,400000	Informar ao usuário que o topo da pilha precisa ser inteiro não negativo.	"Para usar o comando 'c' o topo da pilha precisa ser inteiro não negativo".	2. 6 1. 0,400000	Sim
oi	2. 5,500000 1. 3	Informar ao usuário que a entrada é inválida.	"Entrada Inválida".	2. 5,500000 1. 3	Sim
x	Pilha Vazia	Sair do Modo Calculadora e voltar ao menu.	Saiu do modo calculadora e voltou ao menu.	-	Sim