

# ED: Trabalho 1

Avaliação de Expressões Aritméticas (forma pós-fixa)

Gabriel Correia de Vasconcelos - 16/0120781

Pedro Augusto Coelho Nunes - 16/0141044

# Introdução e Objetivo

- Linguagem de programação C
- Algoritmo para conversão da notação algébrica infixa para pós-fixa e resolução da expressão pós conversão
- Estrutura de dados pilha

# Métodos de desenvolvimento

- Uso do git
- Pair programming



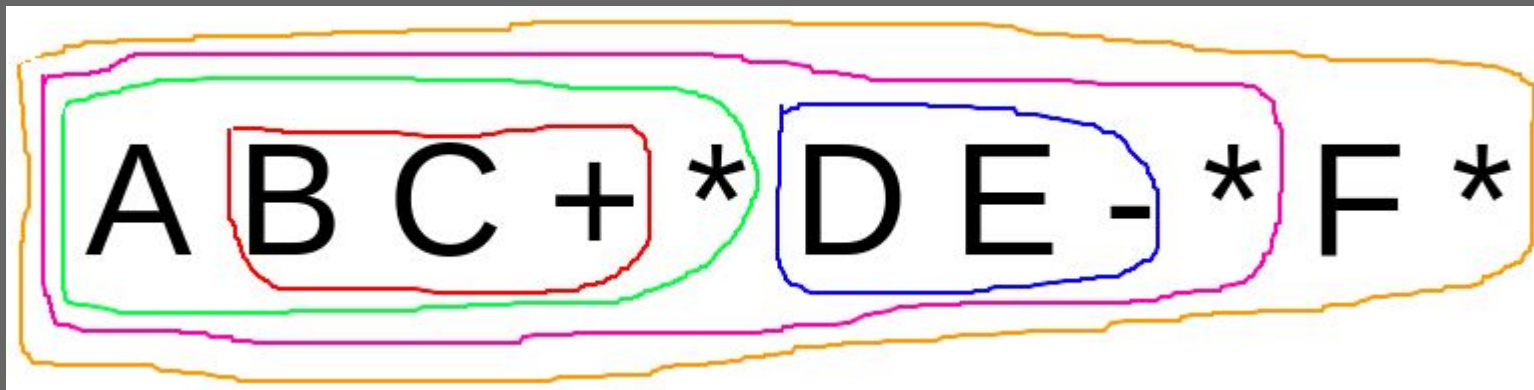
**git**

# Notação infixa

$$A * (B + C) * (D - E) * F$$

Obs: Necessita de parênteses para indicar a ordem das operações

# Notação pós-fixa



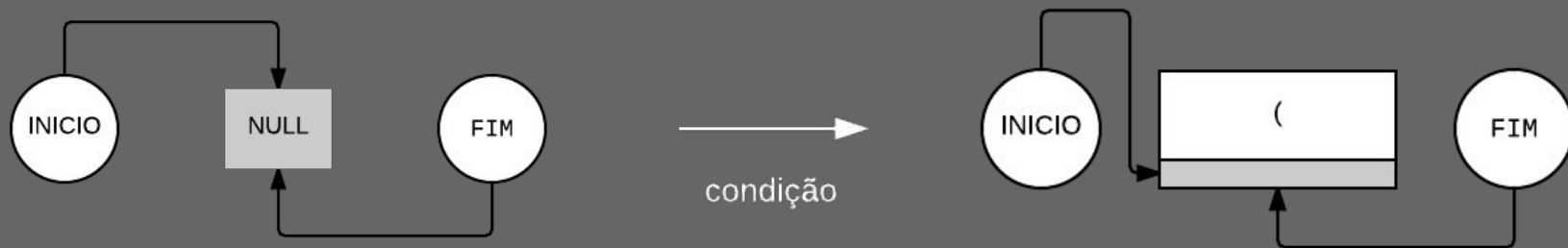
Obs: Não necessita de parênteses para indicar a ordem das operações

# Implementação

1. Recebe um string com a expressão na forma infixa
2. Checa se ela é uma expressão válida
3. Transforma a expressão para a forma pós-fixa
4. Avalia expressão
5. Retorna o resultado

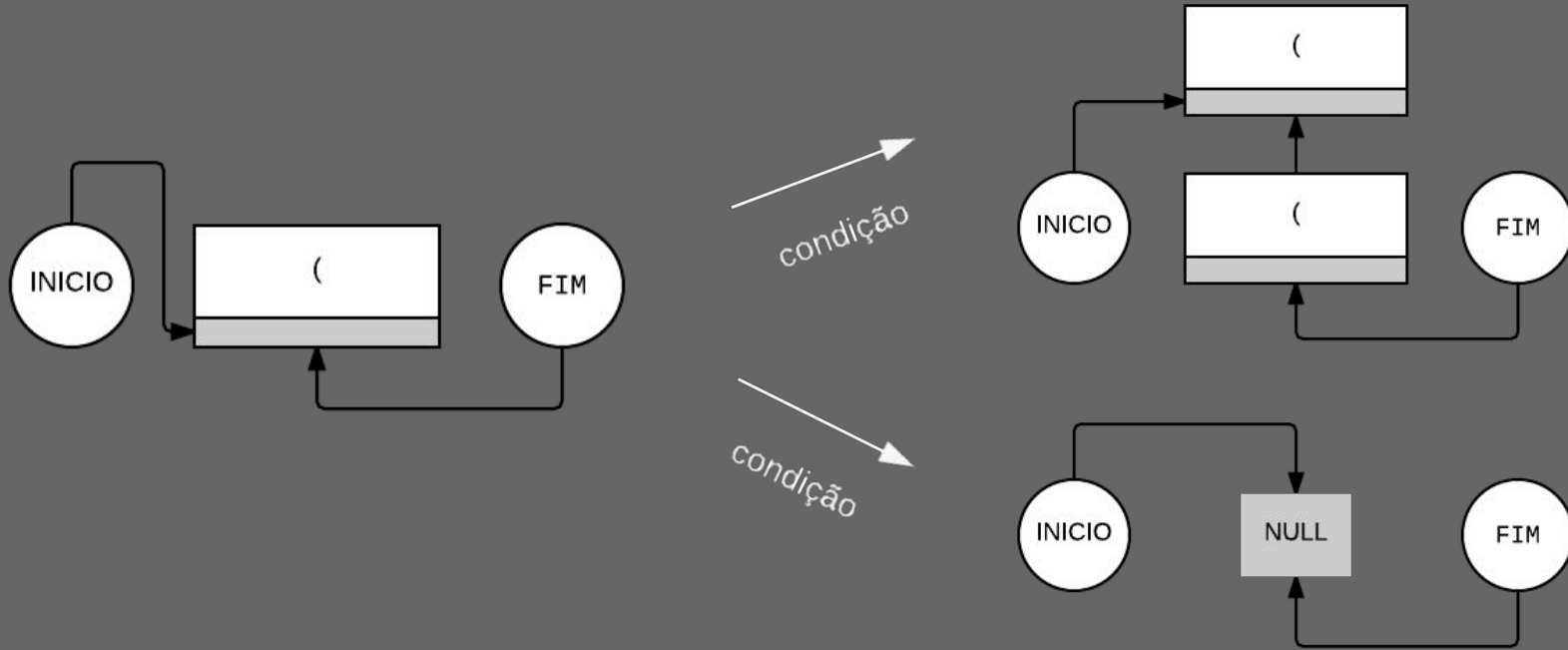
1. Recebe um string com a expressão na forma infixa
  - Recebe uma string infixa[] de tamanho máximo 50

## 2. Checa se ela é uma expressão válida

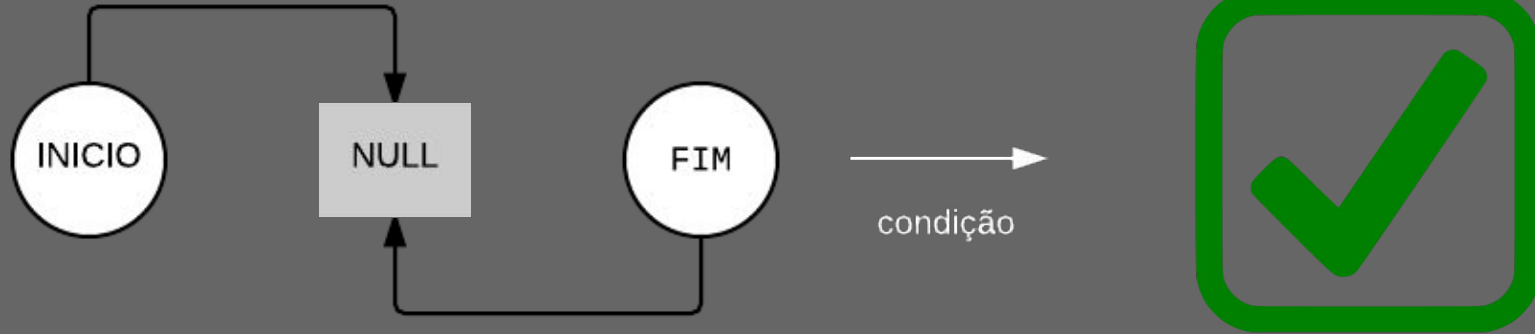




## 2. Checa se ela é uma expressão válida



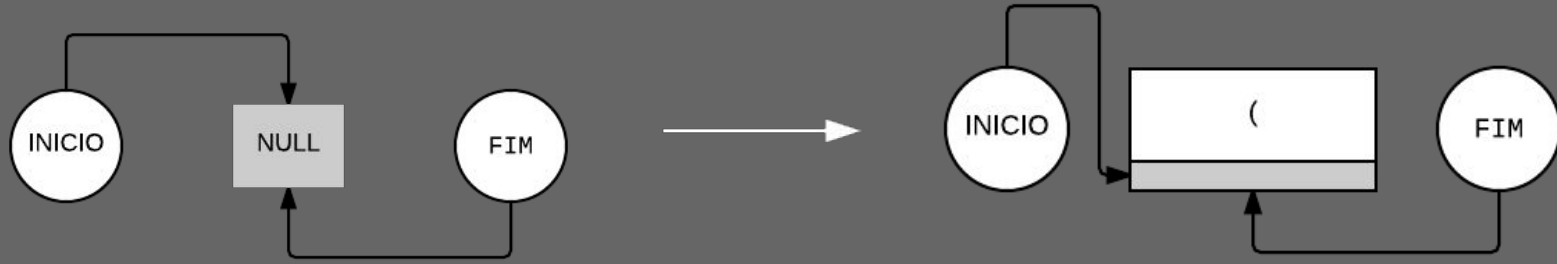
## 2. Checa se ela é uma expressão válida



### 3. Transforma a expressão para a forma pós-fixa

Expressão:  $(1+2)^*3$

Pilha:



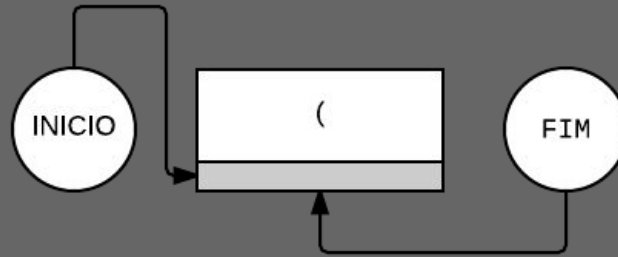
char analisado: (

Expressão pós-fixa:

### 3. Transforma a expressão para a forma pós-fixa

Expressão:  $(1+2)^*3$

Pilha:



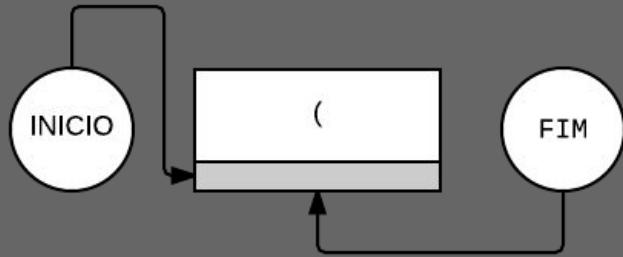
char analisado: 1

Expressão pós-fixa: 1

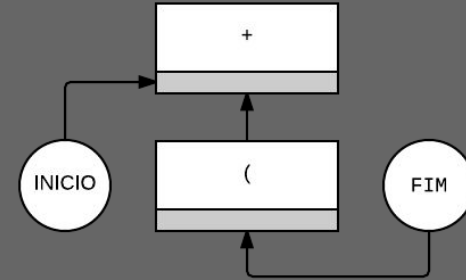
### 3. Transforma a expressão para a forma pós-fixa

Expressão:  $(1+2)^*3$

Pilha:



char analisado: +

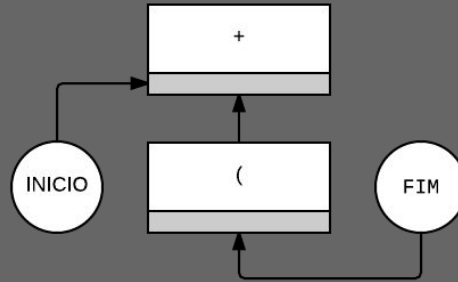


Expressão pós-fixa: 1

### 3. Transforma a expressão para a forma pós-fixa

Expressão:  $(1+2)*3$

Pilha:



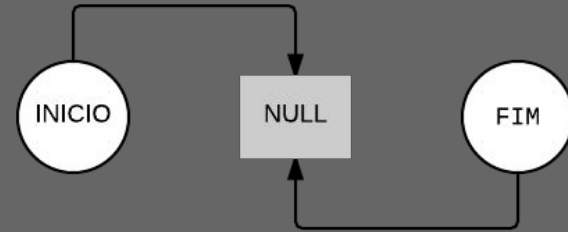
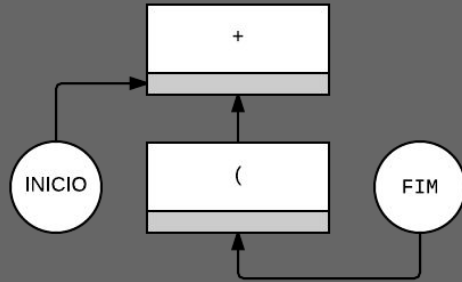
char analisado: 2

Expressão pós-fixa: 1 2

### 3. Transforma a expressão para a forma pós-fixa

Expressão:  $(1+2)^*3$

Pilha:



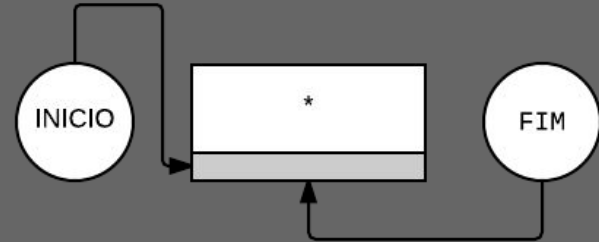
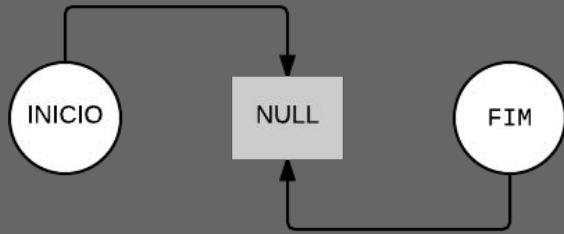
char analisado: )

Expressão pós-fixa: 1 2 +

### 3. Transforma a expressão para a forma pós-fixa

Expressão:  $(1+2)^*3$

Pilha:



char analisado: \*

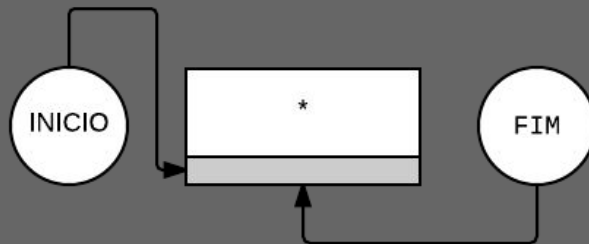
Expressão pós-fixa: 1 2 +



### 3. Transforma a expressão para a forma pós-fixa

Expressão:  $(1+2)*3$

Pilha:



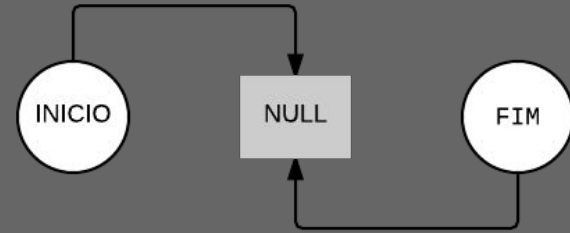
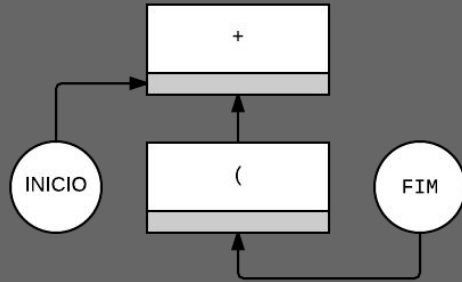
char analisado: 3

Expressão pós-fixa: 1 2 + 3

### 3. Transforma a expressão para a forma pós-fixa

Expressão:  $(1+2)^*3$

Pilha:



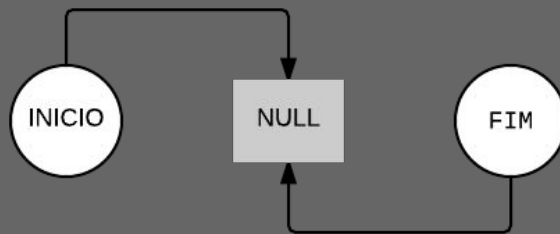
char analisado: \0

expressão pós-fixa: 1 2 + 3 \*

### 3. Transforma a expressão para a forma pós-fixa

Expressão:  $(1+2)^*3$

Pilha:

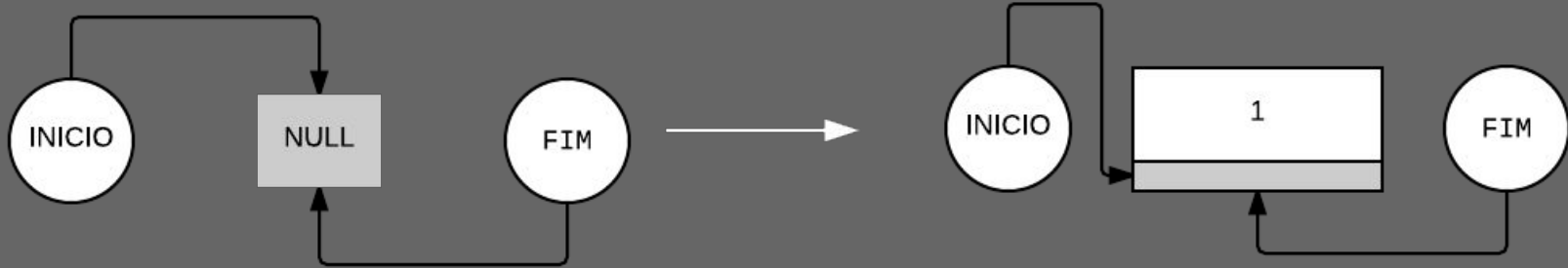


Expressão final pós-fixa:  $1\ 2\ +\ 3\ ^*$

## 4. Avalia expressão

Expressão: 1 2 + 3 \*

Pilha:

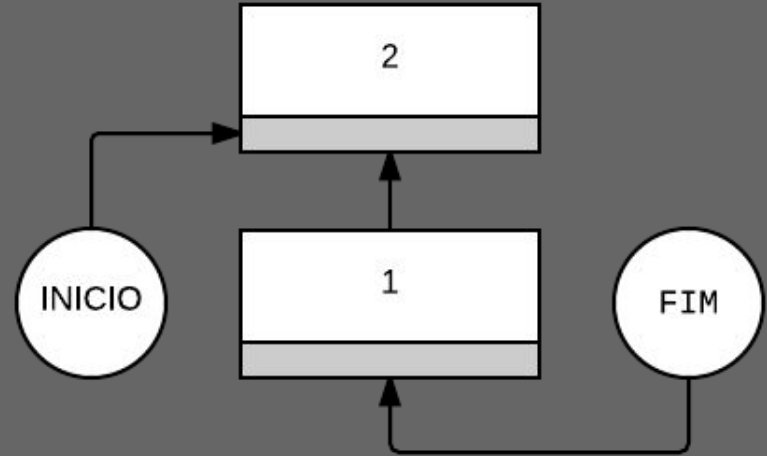
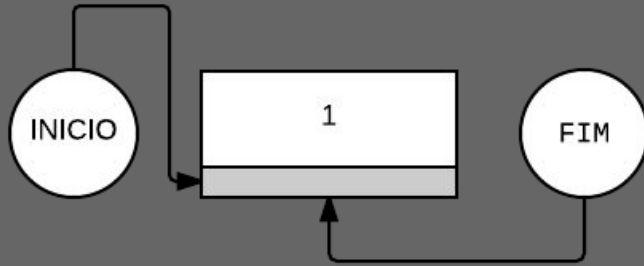


char analisado: 1

## 4. Avalia expressão

Expressão: 1 2 + 3 \*

Pilha:

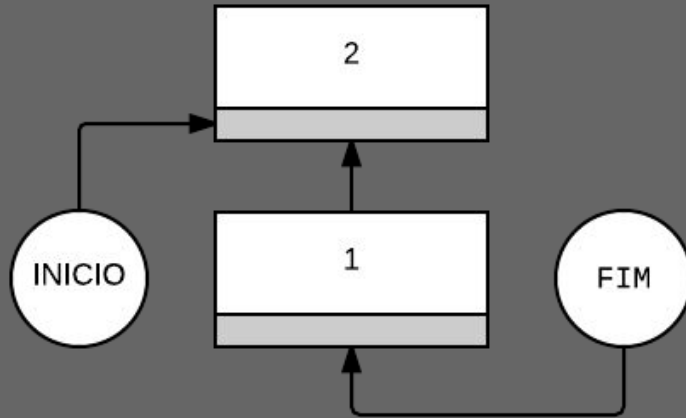


char analisado: 2

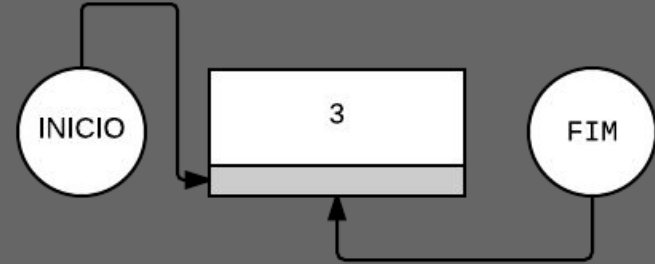
## 4. Avalia expressão

Expressão: 1 2 + 3 \*

Pilha:



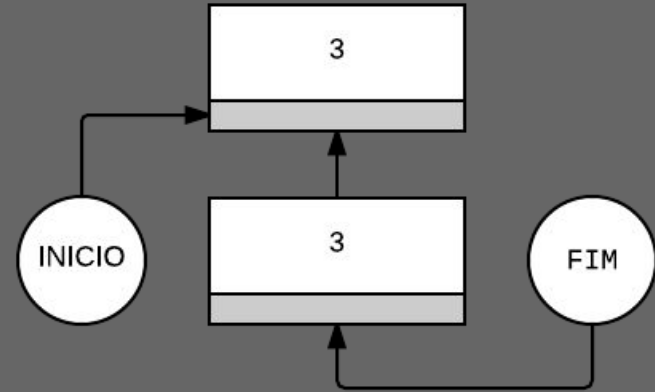
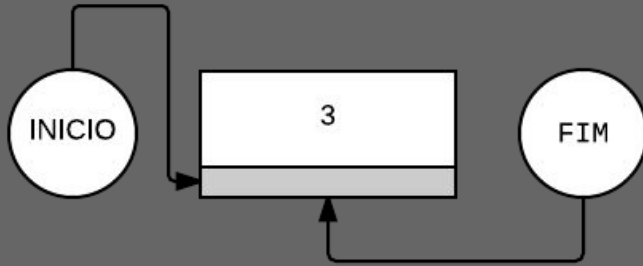
char analisado: +



## 4. Avalia expressão

Expressão: 1 2 + 3 \*

Pilha:

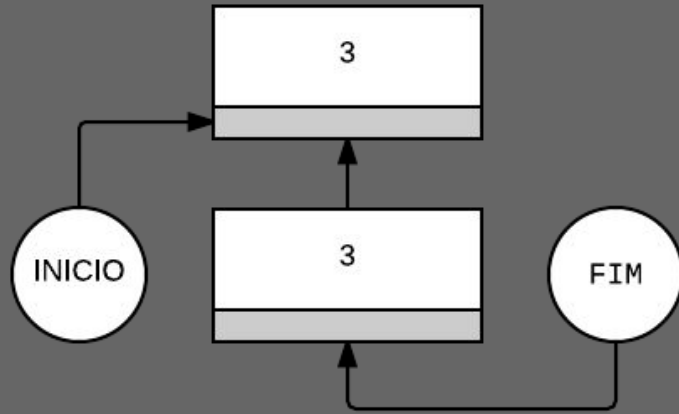


char analisado: 3

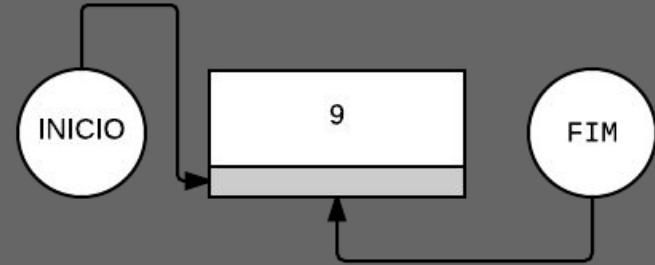
## 4. Avalia expressão

Expressão: 1 2 + 3 \*

Pilha:



char analisado: \*

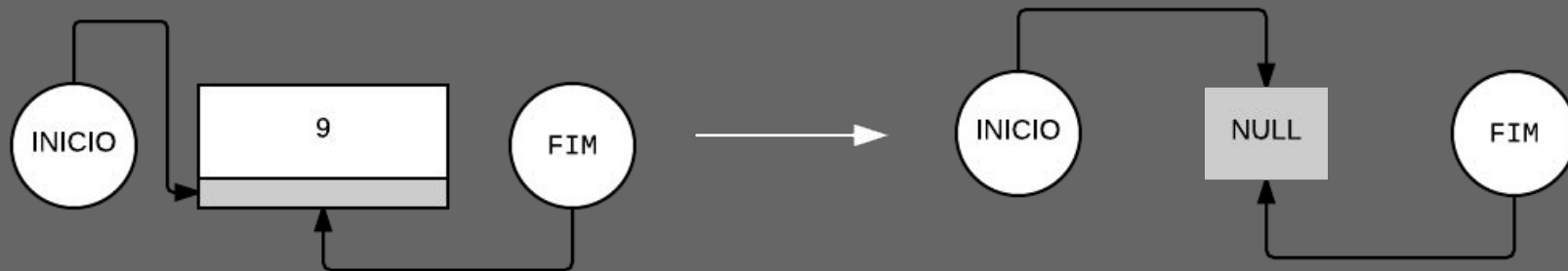




## 5. Retorna o resultado

Expressão:  $1\ 2 + 3 *$

Pilha:



Resultado: 9