UNIVERSIDADE
FERDERAL DO
MARANHÃO

Alocação de Memória

ARTHUR SALIM DA COSTA
LAIS SILVA COSTA
MARIA HELENA DE SOUSA COSTA

Refencias

O código usado tem como base o código mini-so-3.py do prof. Thales Valente

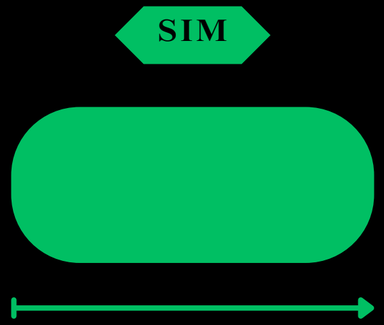
Pode ser observado dentro da class SimpleOSSimulated_v3 nas funções:

- def __init__
- def allocate_memory
- def free_memory
- def execute_program
- def program_calc
- def program_echo

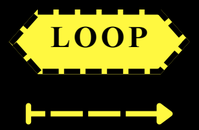
Pode ser observado também no teste de memória abaixo da classe

Legenda

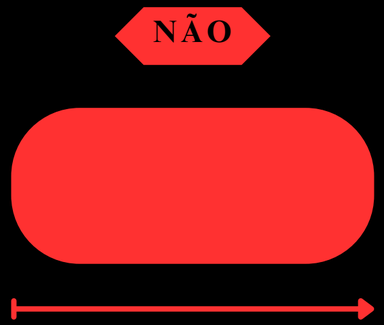
Ação corrente



Condição atendida

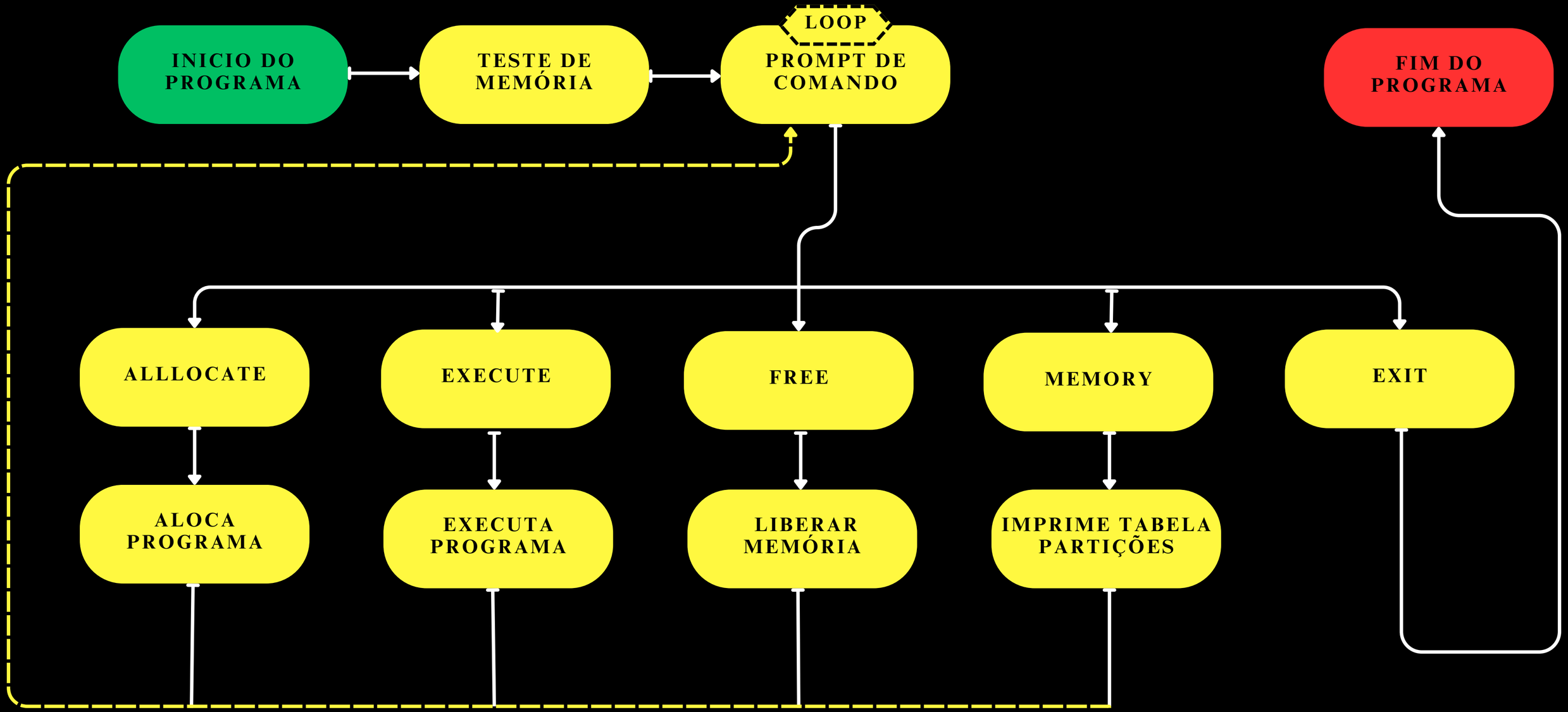


Comando em loop



Condição não atendida

Visão Geral do programa

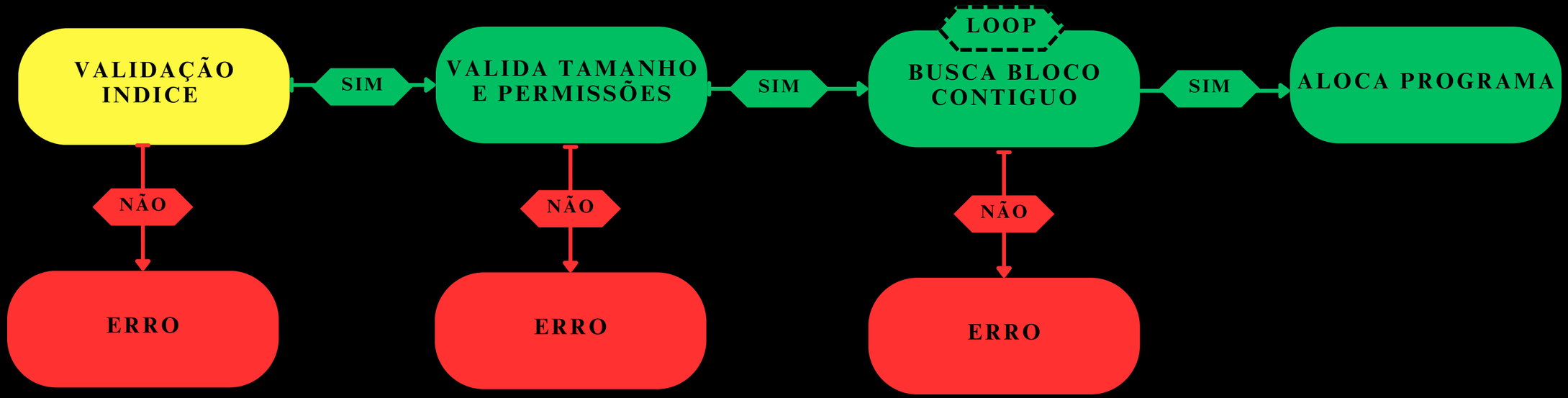


Início do Sistema Operacional Simulado

1. **Configuração da Memória**
2. **Tabela de Partições e Alocação de Memória**
3. **Inicialização da Tabela de Segmentos**

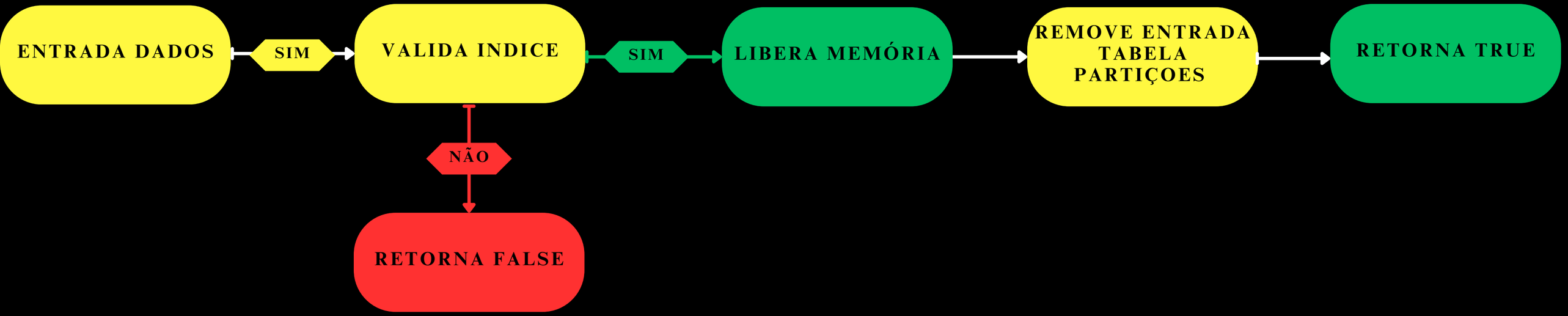
Alocação Memória

1. Alocação de Memória:
2. Validação de Parâmetros
3. Busca por Bloco Contíguo
4. Processo de Alocação



Liberação de Memória:

1. VAlidação dos dados
2. Remoção da Entrada na Tabela de Partições
3. Resultado



Impressão da Tabela de Partições

1. Exibição de Programas Alocados
2. Verificação de Programa ou Função
3. Detecção de Locais Livres
4. Exibição da Tabela de Partições

Impressão da Tabela de Partições

Partições = [4,3,3]

programa echo alocado partição 0

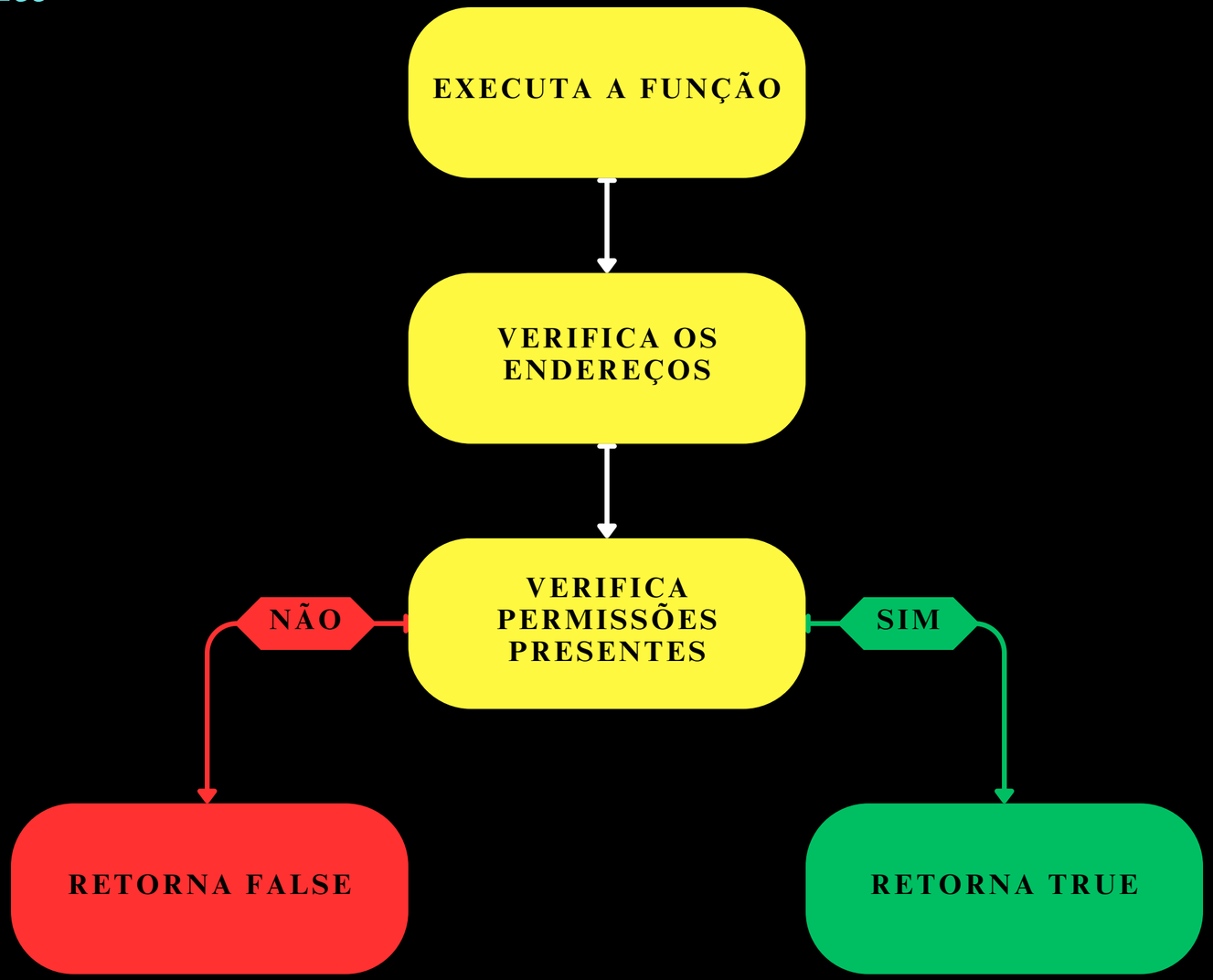
programa calc alocado partição 2

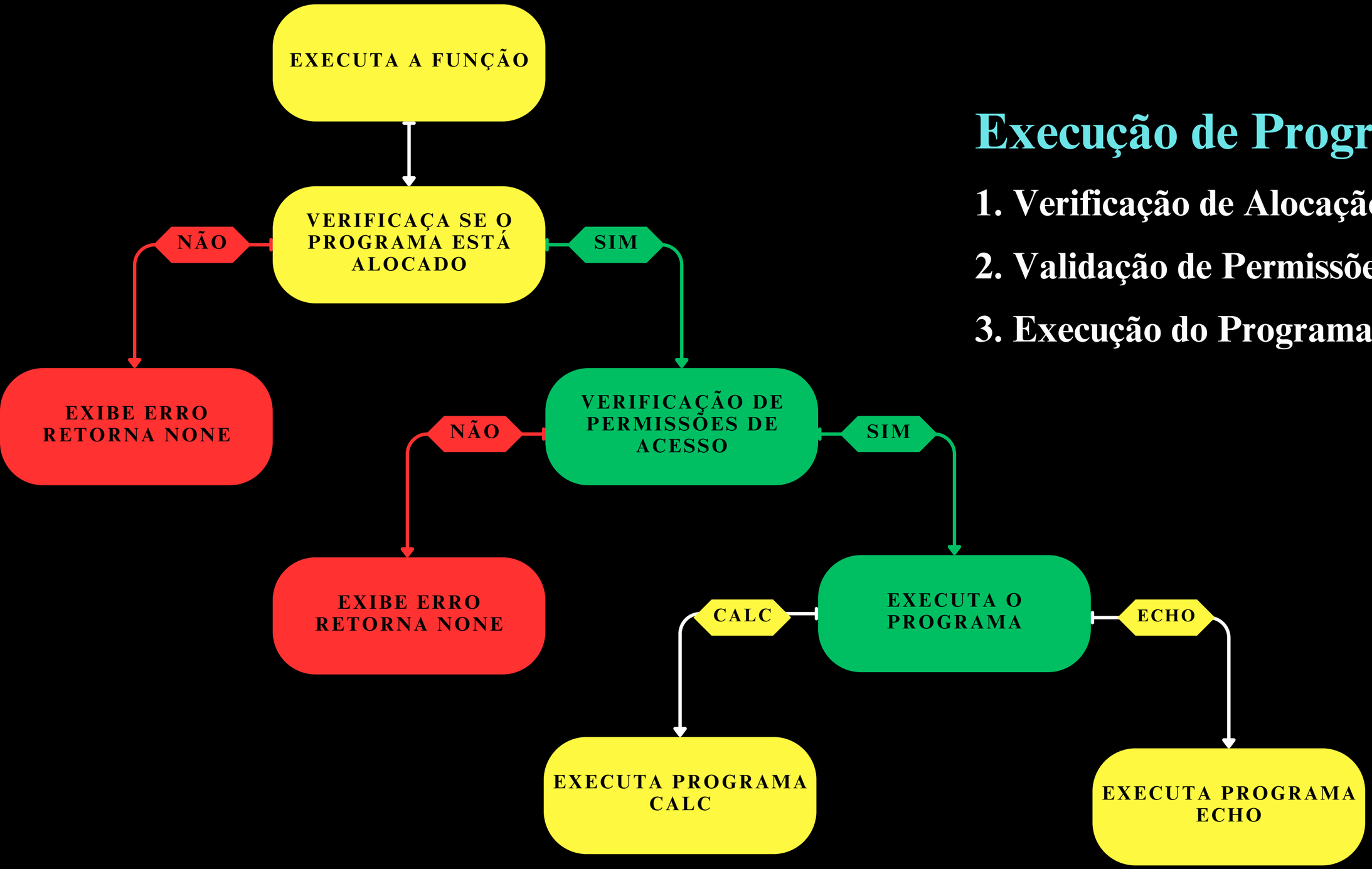
```
============================================================
Tabela de Partições:
Partição 0: Programa echo, Size: 2
Partição 1: Empty
Partição 2: Programa calc, Size: 3

Locais Livres:
Partição 0: Locais livres nos índices [2, 3]
Partição 1: Locais livres nos índices [0, 1, 2]
Partição 2: Nenhum local livre
============================================================
```

Checagem de Acesso à Memória

1. Validação do Índice de Partição
2. Verificação de Intervalo de Endereços
3. Validação de Permissões Requeridas
4. Resultado da Checagem de Acesso



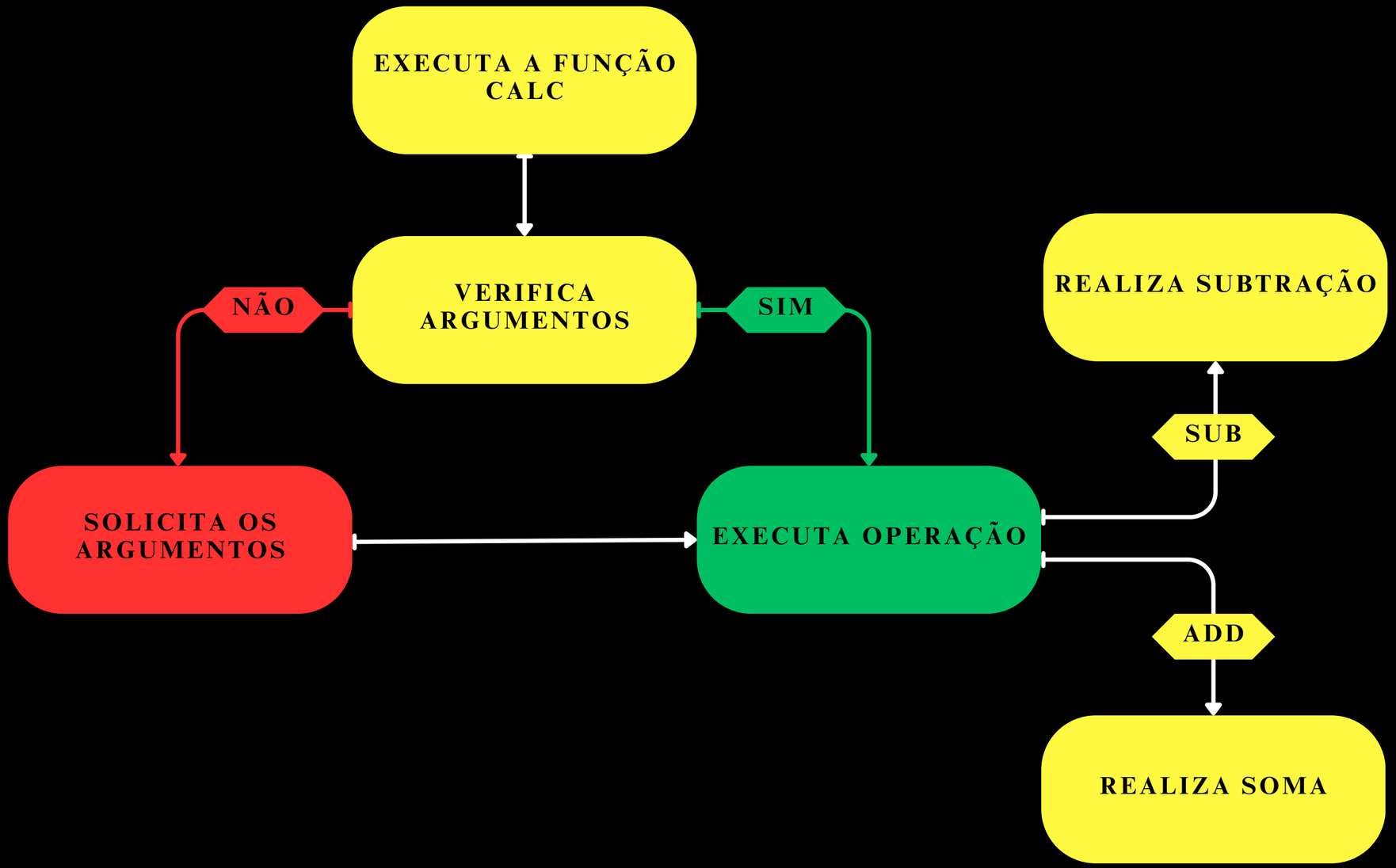


Execução de Programas

1. Verificação de Alocação na Partição
2. Validação de Permissões de Acesso
3. Execução do Programa

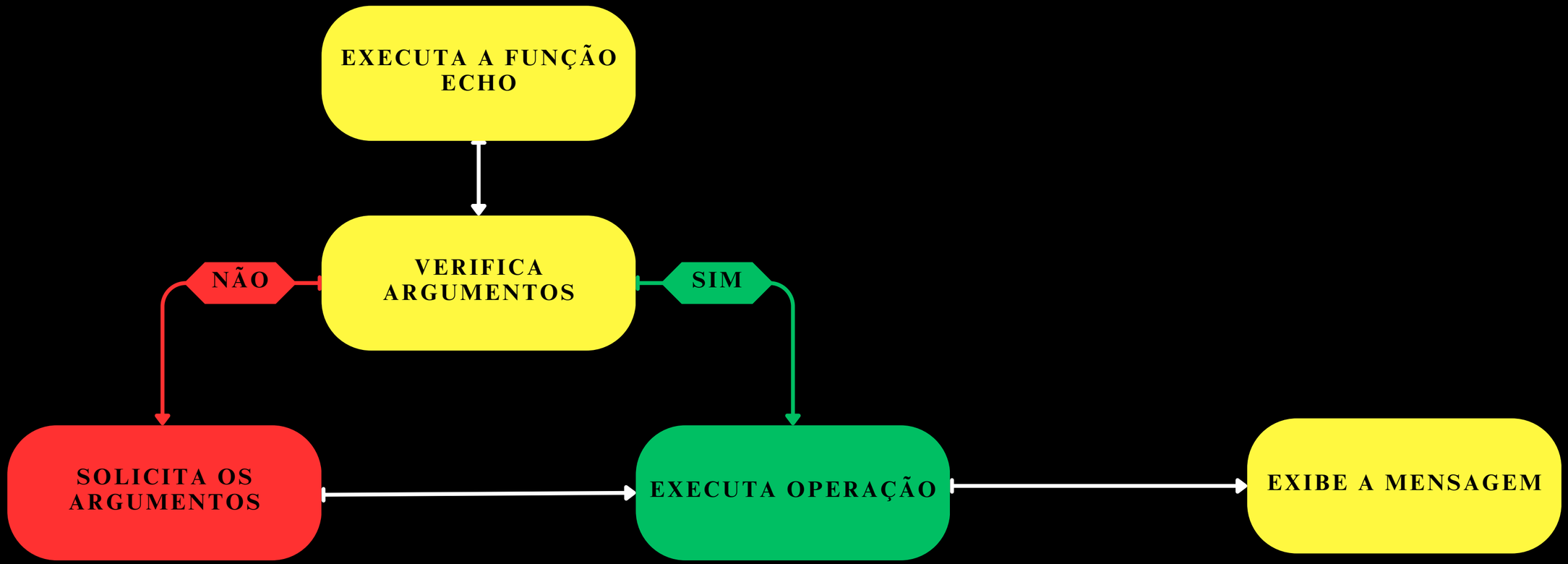
Programas

1. Funcionamento do programa CALC



Programas

1. Funcionamento do programa ECHO



Programas

1. Funcionamento do programa CALC

```
Digite a operação a ser realizada (add, sub): add
Digite os operandos separados por espaço: 435 234 52
Resultado da execução: 721
==========================================================
```

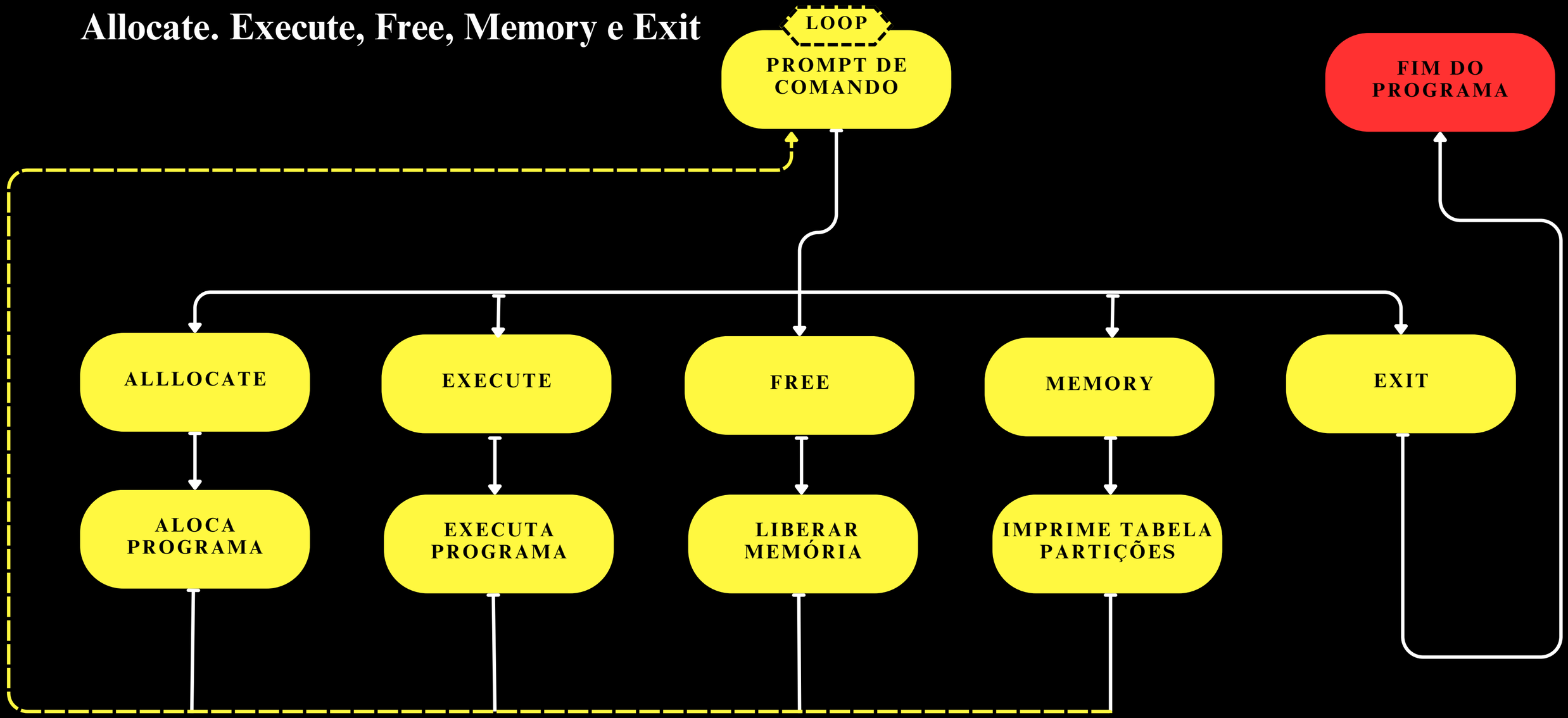
1. Funcionamento do programa ECHO

```
Digite o nome do programa a ser executado (calc, echo): echo
Digite a mensagem a ser ecoada: Hello World! :)
Resultado da execução: Hello World! :)
```

Prompt de Comando

Exibição de menu do promp:

Allocate. Execute, Free, Memory e Exit



Prompt de Comando

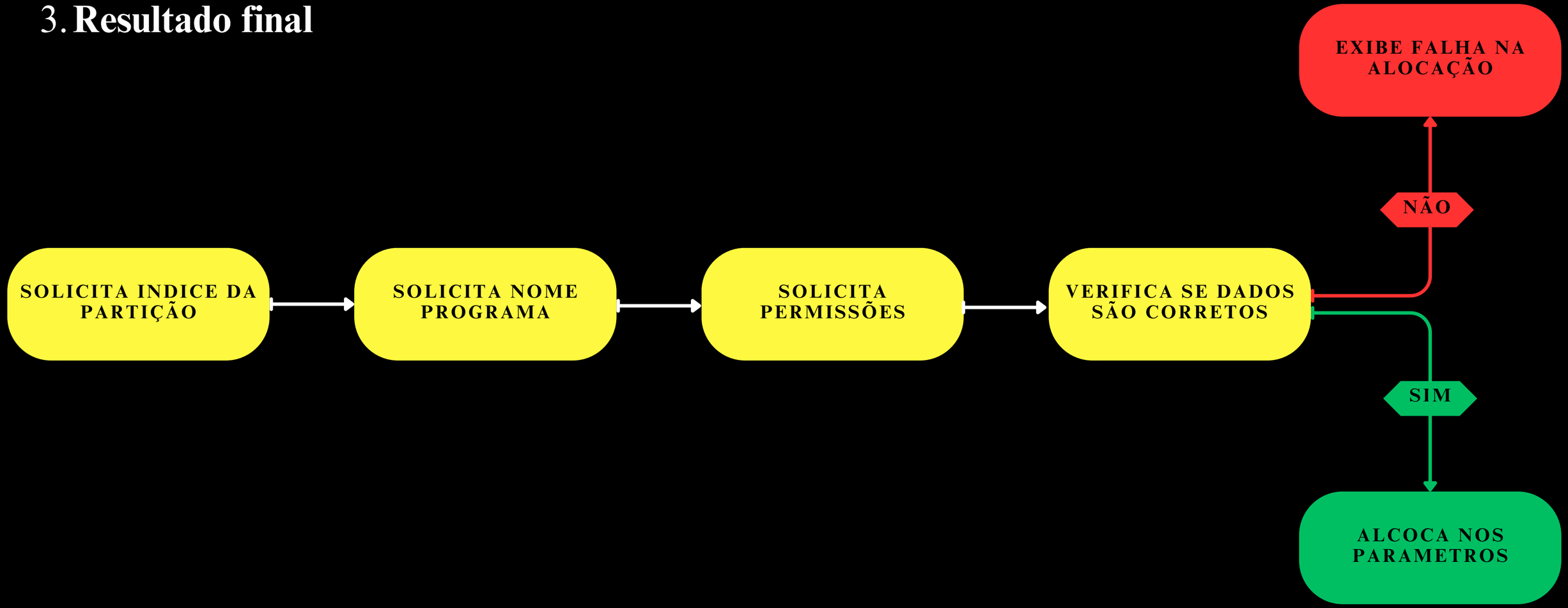
Exibição de menu do promp

- **ALLOCATE**
- **FREE**
- **EXECUTE**
- **MEMORY**
- **EXIT**

```
============================================================
                    PROMPT DE COMANDO
============================================================
Digite um comando (allocate, free, execute, memory, exit): allocate
============================================================
```

Comando Allocate

1. Chamada da Função allocate_memory
2. Solicita ao usuário dados sobre alocação
3. Resultado final

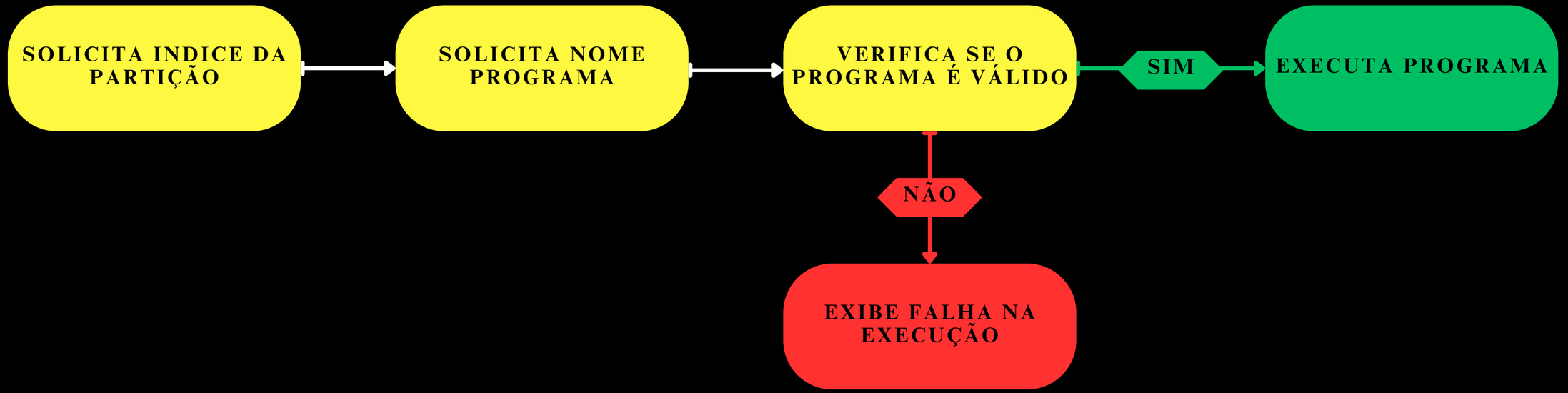


Comando Allocate

```
============================================================
                    PROMPT DE COMANDO
============================================================
Digite um comando (allocate, free, execute, memory, exit): allocate
============================================================
Digite o índice da partição (0, 1, 2): 0
Digite o nome do programa: calc
Digite o tamanho do programa: 3
Digite as permissões (r, w, x): rwx
Memória alocada com sucesso: Partição 0, Início 0, Tamanho 3, Permissões: rwx
                    PROMPT DE COMANDO
============================================================
Digite um comando (allocate, free, execute, memory, exit): allocate
============================================================
Digite o índice da partição (0, 1, 2): 2
Digite o nome do programa: echo
Digite o tamanho do programa: 3
Digite as permissões (r, w, x): rx
Falha ao alocar memória.
```

Comando Execute

1. Solicita ao usuário a partição desejada
2. Solicita o nome do programa a ser executado
3. Resultado da execução

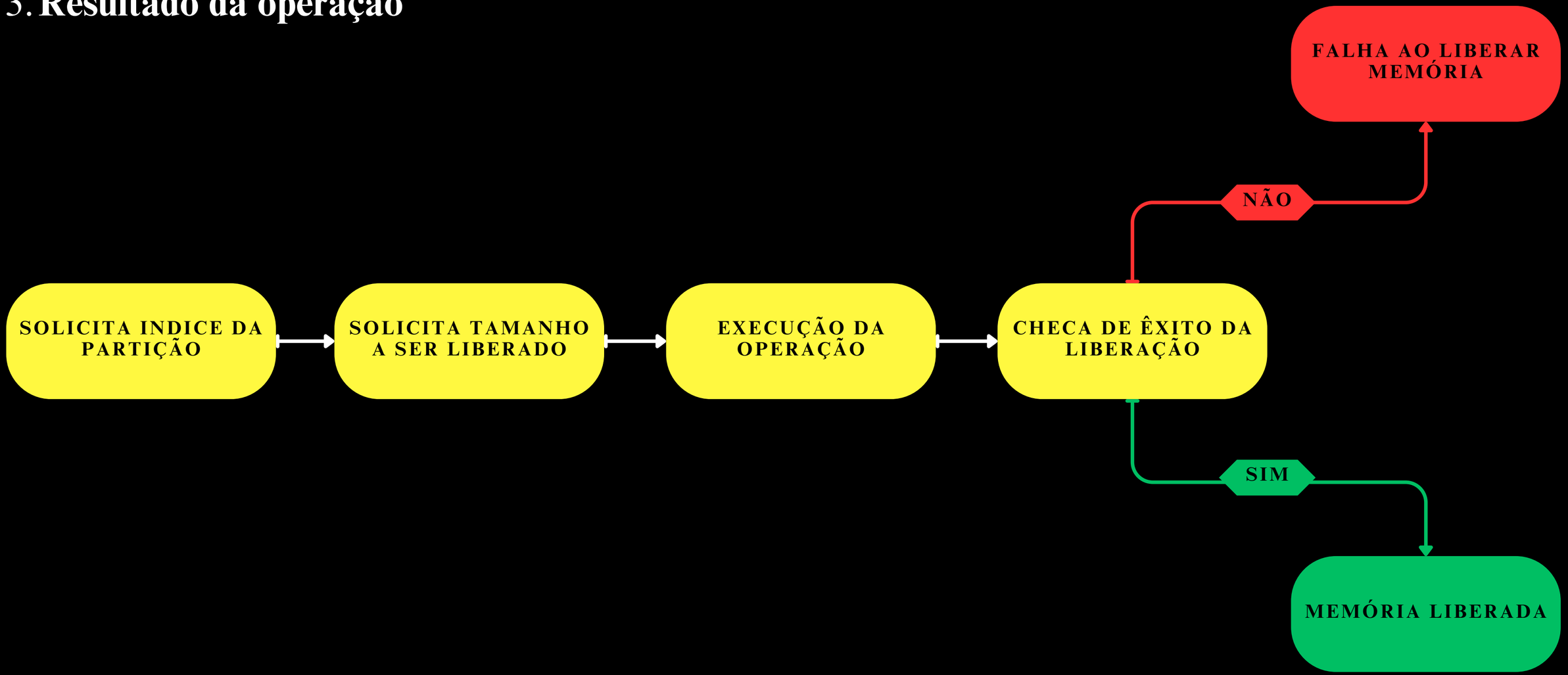


Comando Execute

```
                    PROMPT DE COMANDO
=====================================================
Digite um comando (allocate, free, execute, memory, exit): execute
=====================================================
Digite o índice da partição (0, 1, 2): 0
Digite o nome do programa a ser executado (calc, echo): calc
Digite a operação a ser realizada (add, sub): sub
Digite os operandos separados por espaço: 999 323 123
Resultado da execução: 553
=====================================================
```

Comando Free

1. Solicita ao usuário dados para liberação de memória
2. Verifica os dados fornecidos
3. Resultado da operação



Comando Free

1. **Solicita ao usuário dados para liberação de memória**
2. **Verifica os dados fornecidos**
3. **Resultado da operação**

```
============================================================
Digite um comando (allocate, free, execute, memory, exit): free
============================================================
Digite o índice da partição (0, 1, 2): 0
Digite o tamanho do bloco a ser liberado: 3
Memória liberada com sucesso: Partição 0, Tamanho 3.
                    PROMPT DE COMANDO
============================================================
Digite um comando (allocate, free, execute, memory, exit): free
============================================================
Digite o índice da partição (0, 1, 2): 2
Digite o tamanho do bloco a ser liberado: 2
Memória liberada com sucesso: Partição 2, Tamanho 2.
```

Comando Memory

1. chama
2. Exibe Tabela de partições

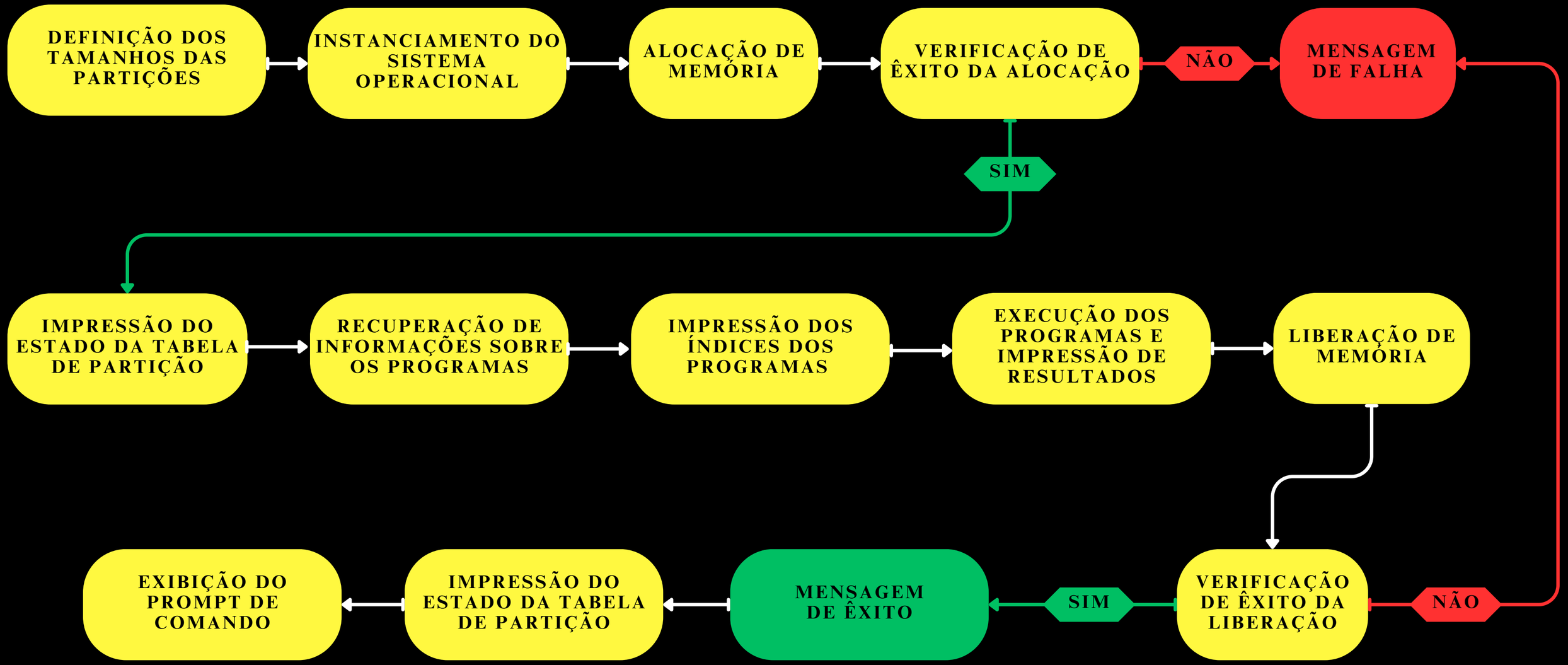
Comando Exit

1. Encerra o programa

```
============================================================
                    PROMPT DE COMANDO
============================================================
Digite um comando (allocate, free, execute, memory, exit): memory
============================================================
Tabela de Partições:
Partição 0: Programa calc, Size: 3
Partição 1: Programa echo, Size: 3
Partição 2: Empty

Locais Livres:
Partição 0: Locais livres nos índices [3]
Partição 1: Nenhum local livre
Partição 2: Locais livres nos índices [0, 1, 2]
============================================================
                    PROMPT DE COMANDO
============================================================
Digite um comando (allocate, free, execute, memory, exit): exit
Sistema encerrado.
```

TESTE DE FUNCIONAMENTO



Reconhecimentos e Direitos Autorais

@autor: MARIA HELENA DE SOUSA COSTA
LAIS SILVA COSTA
ARTHUR SALIM DA COSTA

@contato: maria.hsc@discente.ufma.br
lais.sc@discente.ufma.br
arthur.salim@discente.ufma.br

@data última versão: [10.12.2023]
@versão: 1.0

@Agradecimentos: Universidade Federal do Maranhão (UFMA), Professor Doutor Thales Levi Azevedo Valente, e colegas de curso.