

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA MINAS GERAIS**

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

BIANCA DE CASTRO AGUIAR FONTES

**ATIVIDADE 7**

Pilha dinâmica

SÃO JOÃO EVANGELISTA

2023

SUMÁRIO

[1. FUNCIONALIDADE DE UMA PILHA COM ALOCAÇÃO DINÂMICA EM C++ 3](#_Toc149723345)

[2. CÓDIGO COMENTADO 4](#_Toc149723346)

[a. Bibliotecas e Namespace 4](#_Toc149723347)

[b. Estrutura de dados “Item” 4](#_Toc149723348)

[c. Ponteiro “topo” 5](#_Toc149723349)

[d. Cabeçalhos das funções 5](#_Toc149723350)

[e. Função “main” 6](#_Toc149723351)

[f. Função “menu” 7](#_Toc149723352)

[g. Função “empilhar” 7](#_Toc149723353)

[h. Função “verificarSeTemAlgumaCoisa” 8](#_Toc149723354)

[i. Função “desempilhar” 8](#_Toc149723355)

[j. Função “mostrar” 9](#_Toc149723356)

# FUNCIONALIDADE DE UMA PILHA COM ALOCAÇÃO DINÂMICA EM C++

A pilha é uma estrutura de dados que segue uma regra onde o último elemento a ser inserido é o primeiro a ser removido, ou seja, os elementos são inseridos ou removidos do topo.

A alocação dinâmica consiste em alocar memória durante a execução do programa, ao contrário de declará-la no código. Se torna bem usual em exemplos onde o tamanho da estrutura de dados não é definida ou varia durante a execução do programa.

Para que uma pilha com alocação dinâmica funcione em um programa desenvolvido em c++, precisamos de dois ponteiros:

* topo – aponta sempre o último elemento inserido na pilha.
* aux – aponta sempre para o elemento anterior da fila, possibilitando assim uma ligação entre os elementos da pilha.

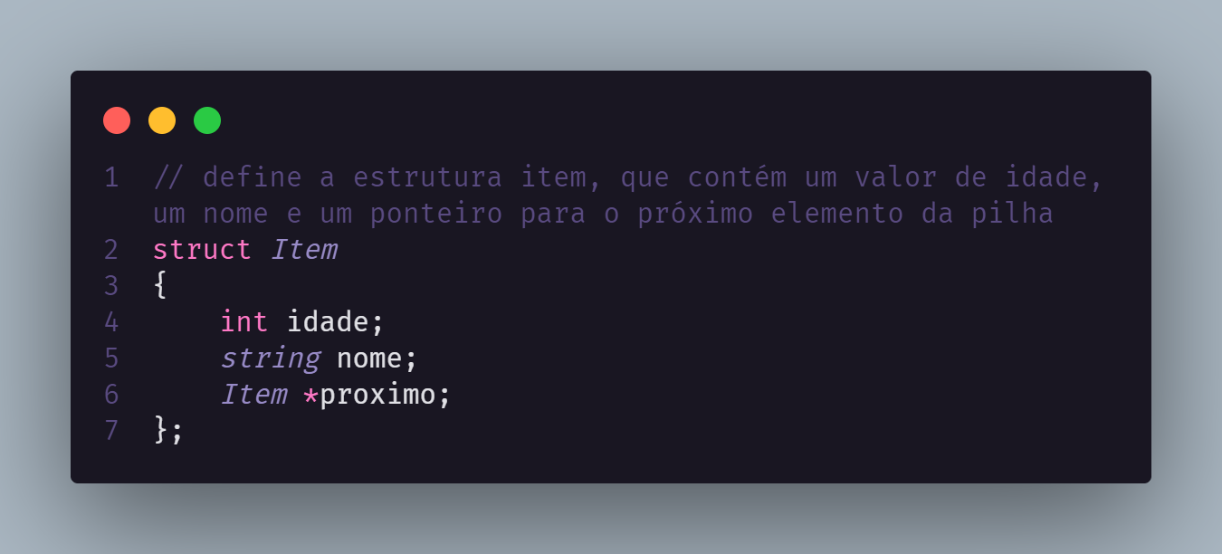
Ao inserir um elemento na pilha, o valor do ponteiro “topo” aumenta e aloca automaticamente a memória para o novo elemento, um processo parecido ocorre ao remover um elemento, já que o ponteiro “topo” diminui o valor e libera o espaço de memória que estava alocado para o elemento removido. Esses processos são realizados utilizando os ponteiros citados, onde se relacionam, recebem e passam valores entre si, possibilitando a criação da pilha.

# CÓDIGO COMENTADO

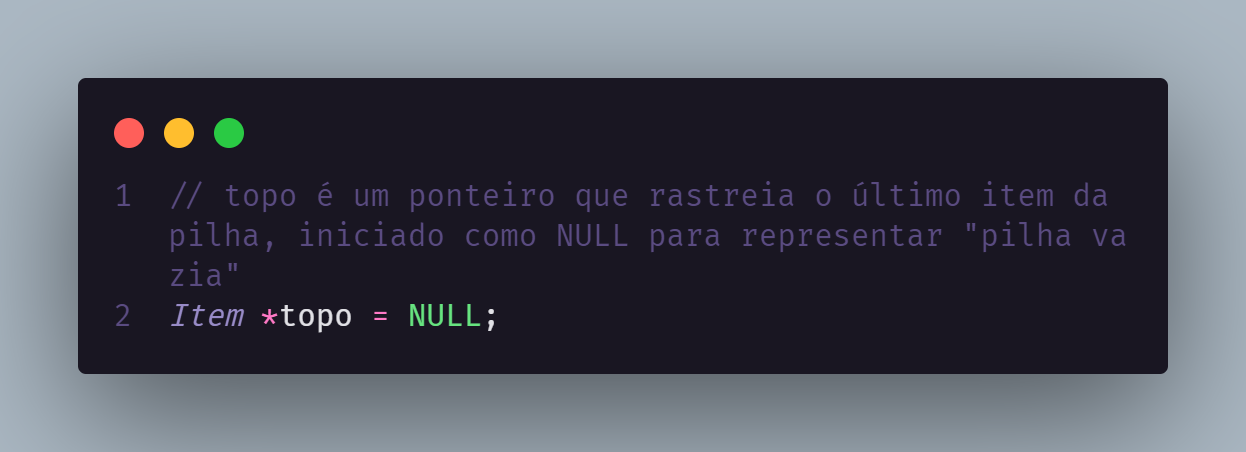
## Bibliotecas e Namespace



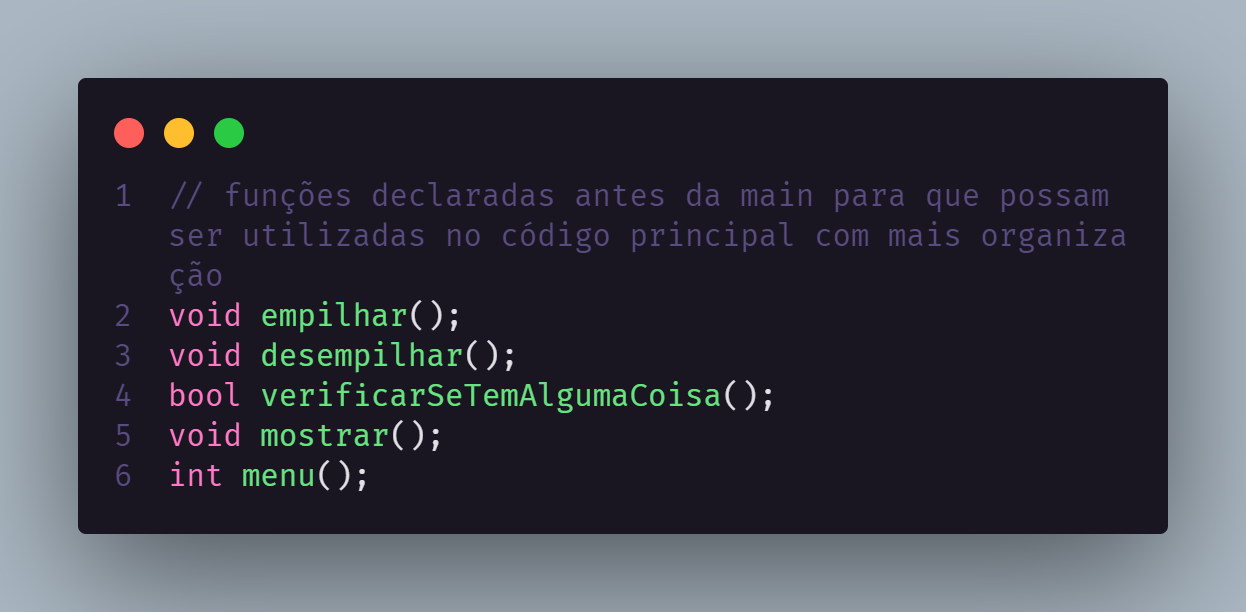
## Estrutura de dados “Item”



## Ponteiro “topo”



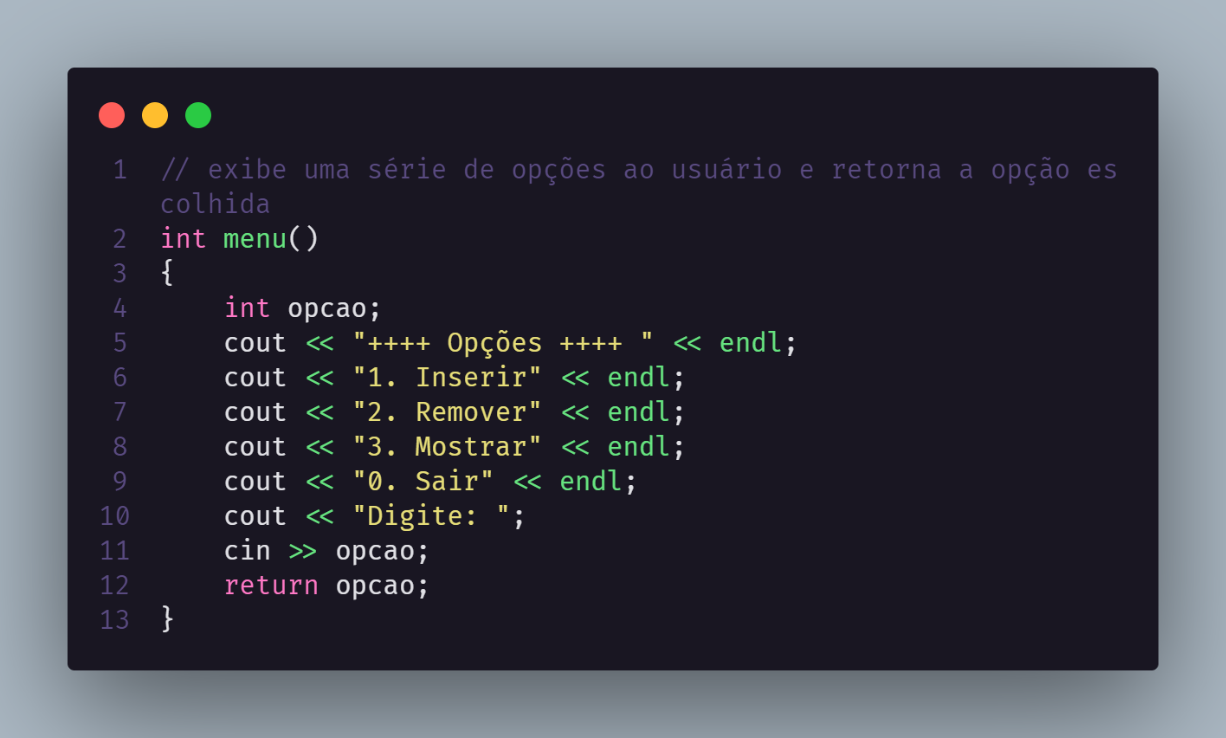
## Cabeçalhos das funções



## Função “main”



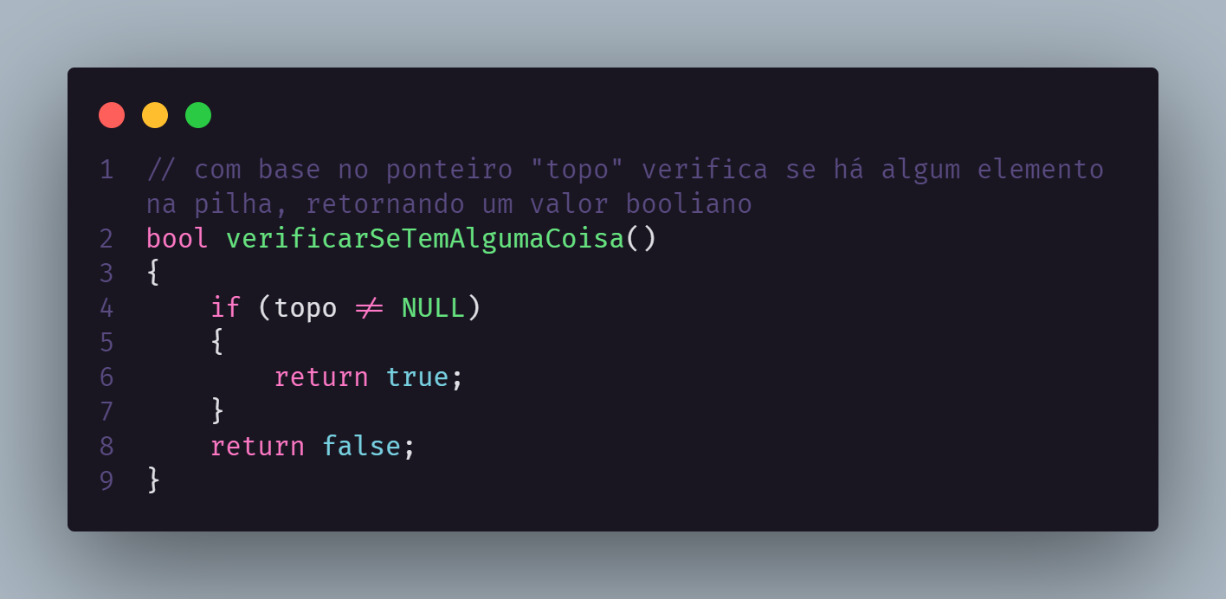
## Função “menu”



## Função “empilhar”



## Função “verificarSeTemAlgumaCoisa”



## Função “desempilhar”

## Função “mostrar”

