

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

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

BIANCA DE CASTRO AGUIAR FONTES

**ATIVIDADE 8**

Lista dinâmica

SÃO JOÃO EVANGELISTA

2023

SUMÁRIO

[1. FUNCIONALIDADE DE UMA LISTA COM ALOCAÇÃO DINÂMICA EM C++ 3](#_Toc151580435)

[2. CÓDIGO COMENTADO 4](#_Toc151580436)

[a. Bibliotecas e Namespace 4](#_Toc151580437)

[b. Estrutura de dados “Item” 4](#_Toc151580438)

[c. Estrutura da lista 5](#_Toc151580439)

[d. Ponteiro do tipo lista como nulo e função para criar lista vazia 5](#_Toc151580440)

[e. Função que verifica se a lista está vazia 5](#_Toc151580441)

[f. Função insere item na última posição 6](#_Toc151580442)

[g. Função que mostra os itens da lista 6](#_Toc151580443)

[h. Função insere item na primeira posição da lista 7](#_Toc151580444)

[i. Função que insere item em uma posição específica 8](#_Toc151580445)

[j. Função que remove o primeiro item da lista 9](#_Toc151580446)

[k. Função que remove o último item da lista 10](#_Toc151580447)

[l. Função que remove o item de uma posição específica 11](#_Toc151580448)

[m. Função menu 12](#_Toc151580449)

[n. Função main 13](#_Toc151580450)

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

A lista é uma estrutura de dados onde seus itens são organizados de forma automática durante a execução. Podemos inserir, remover e editar itens mantendo a lista organizada utilizando ponteiros que indicam o próximo item.

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

* próximo – aponta sempre o próximo item da lista. (tipo de estrutura item)
* primeiro – item cabeça, responsável por apontar sempre o primeiro item da lista. (tipo de estrutura item)
* Último – responsável por apontar último item da lista (tipo de estrutura item)
* L – que é de fato a lista dinâmica (tipo de estrutura lista)

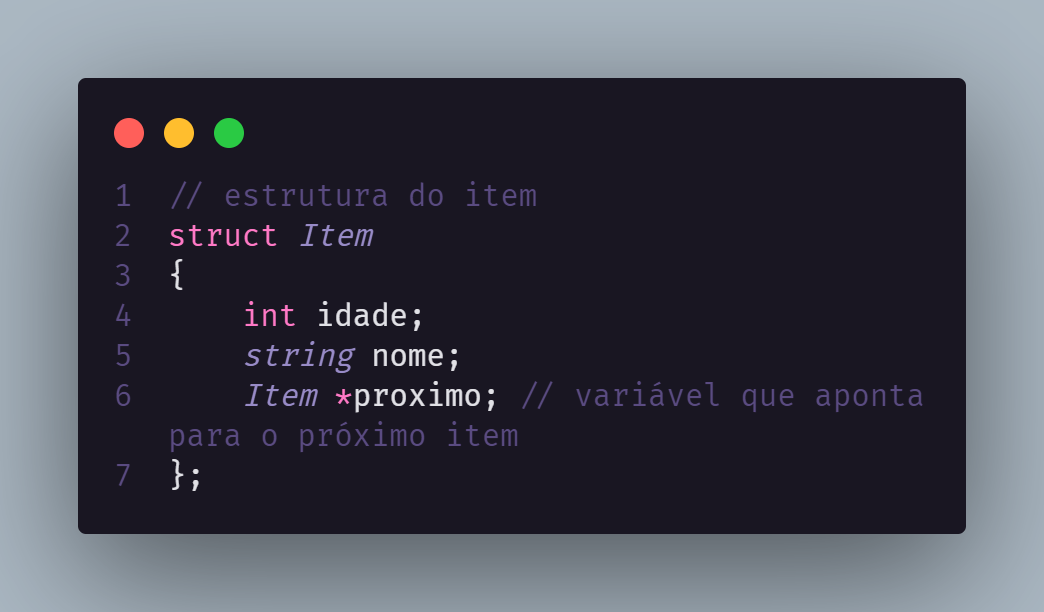
Ao inserir um elemento na lista dinâmica, o valor do ponteiro próximo do último item é atualizado para apontar para o novo elemento, expandindo assim a lista. Um processo semelhante ocorre ao remover um elemento, onde o ponteiro próximo do penúltimo item passa a apontar para o item nulo, liberando assim o espaço de memória do elemento removido com o delete na variável auxiliar. Esses processos são realizados de maneira coordenada pelos ponteiros mencionados para inserção e remoção em qualquer posição da lista, possibilitando a manipulação dinâmica.

# CÓDIGO COMENTADO

## Bibliotecas e Namespace



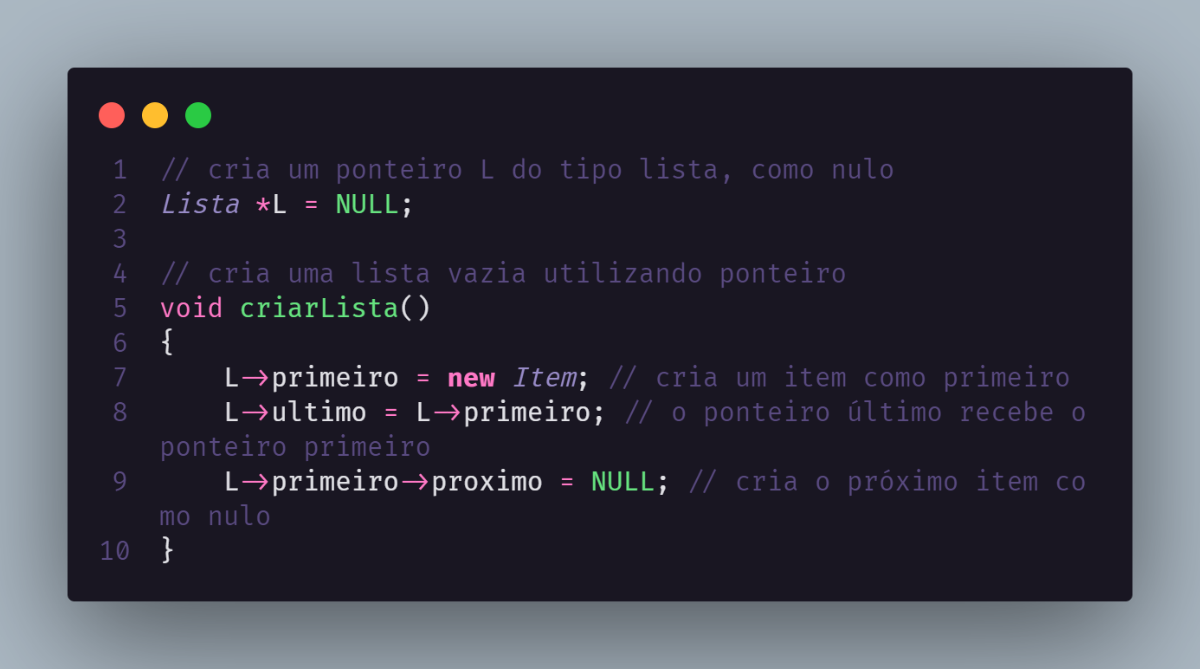
## Estrutura de dados “Item”



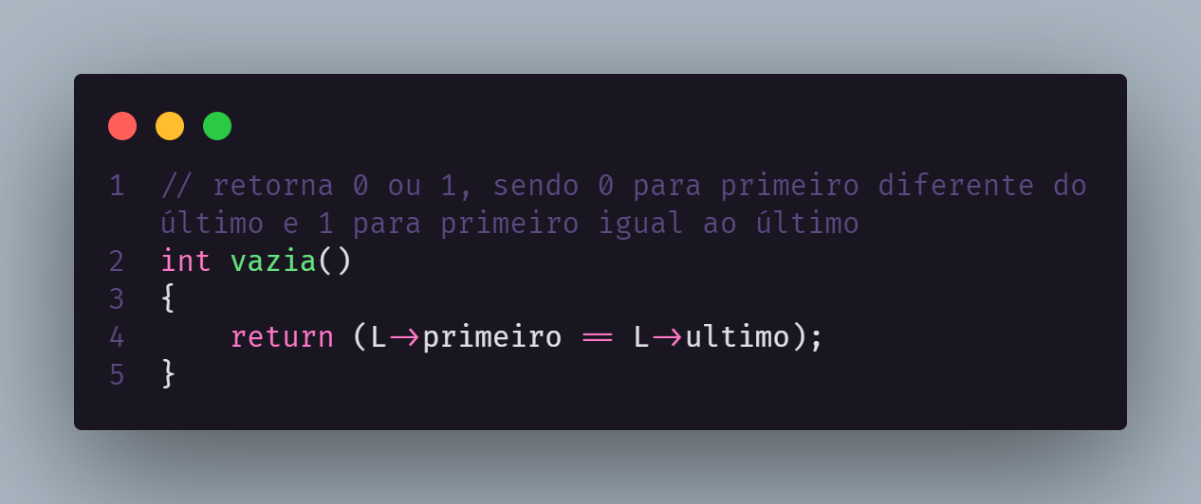
## Estrutura da lista



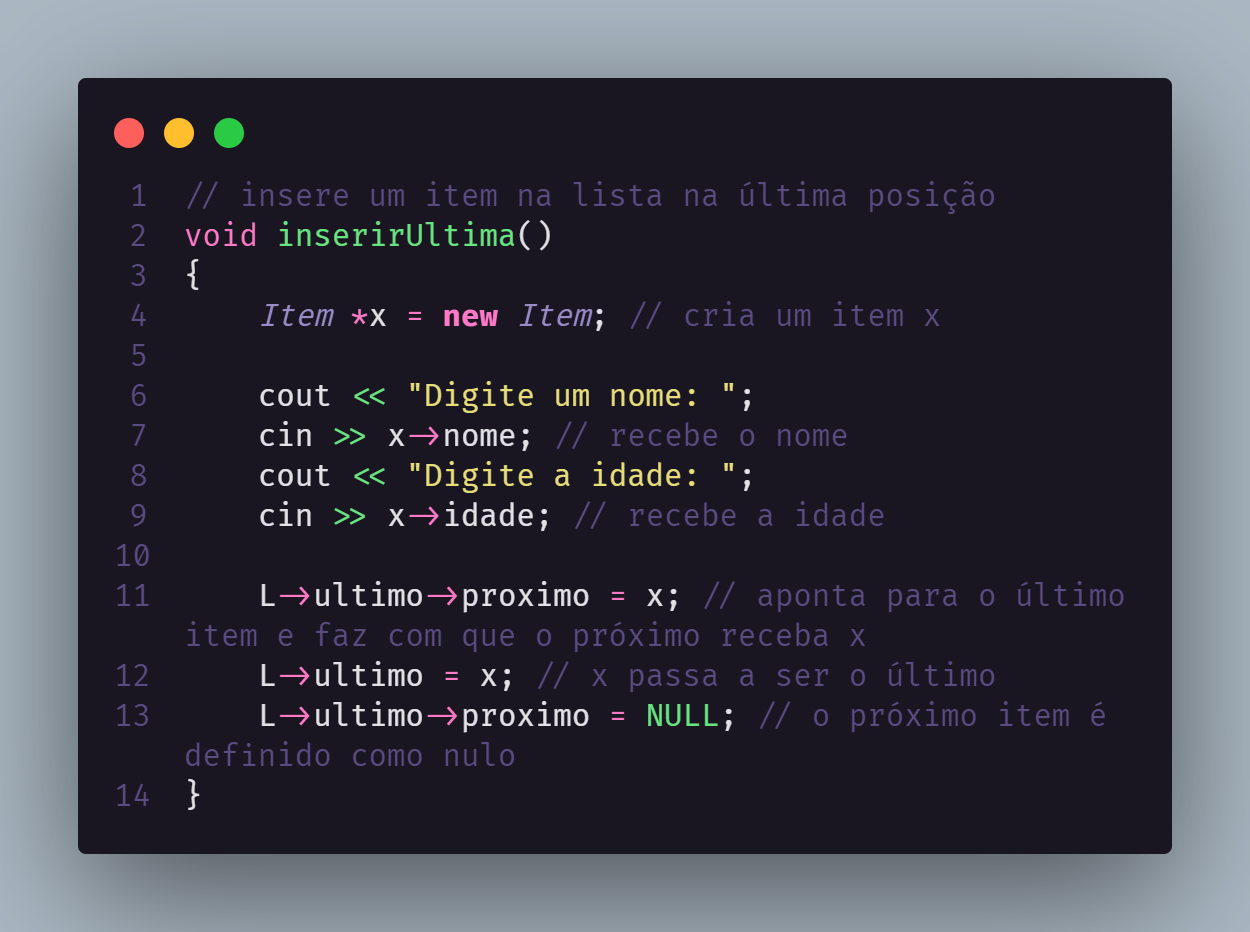
## Ponteiro do tipo lista como nulo e função para criar lista vazia



## Função que verifica se a lista está vazia

****

## Função insere item na última posição



## Função que mostra os itens da lista



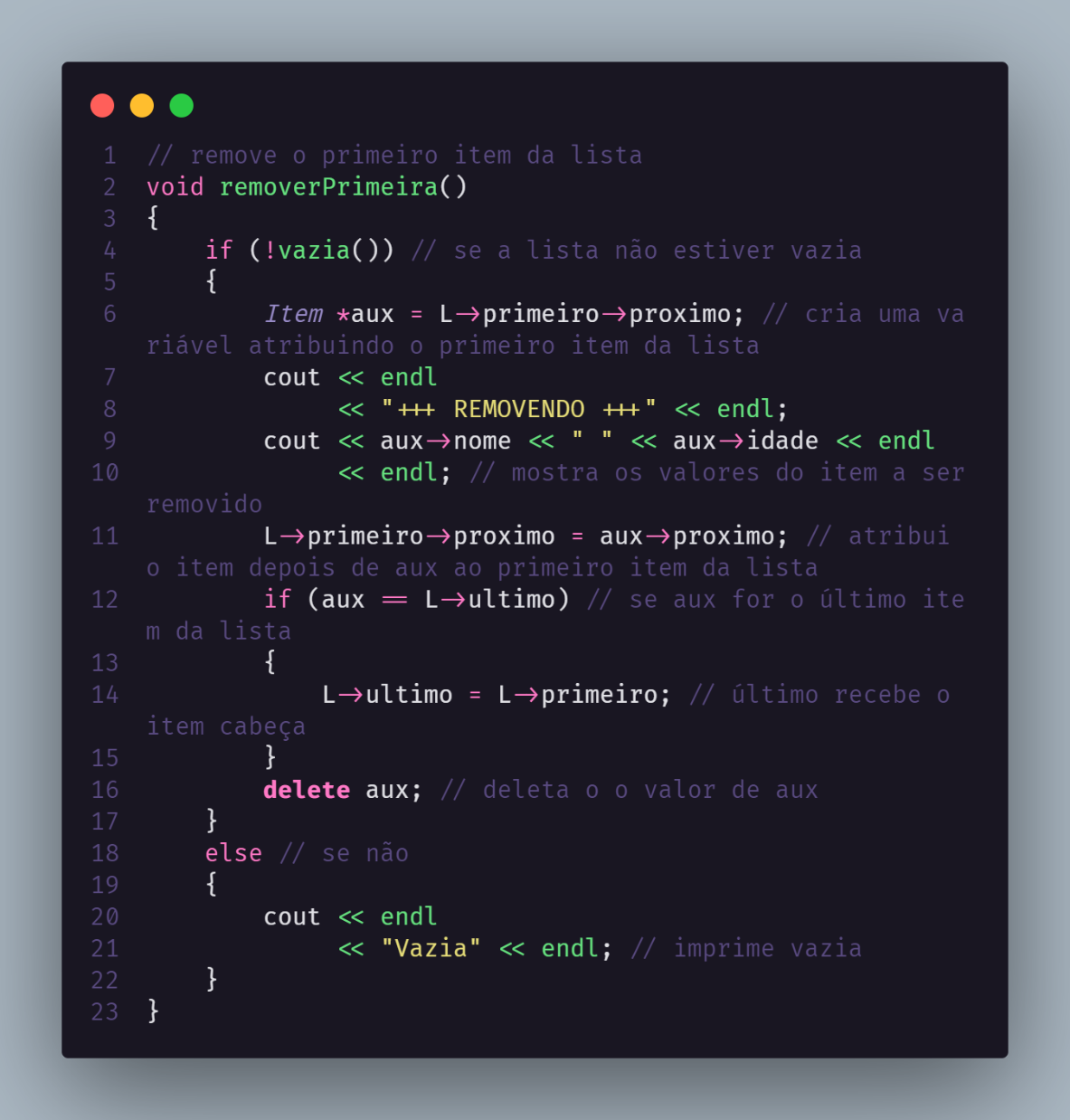
## Função insere item na primeira posição da lista



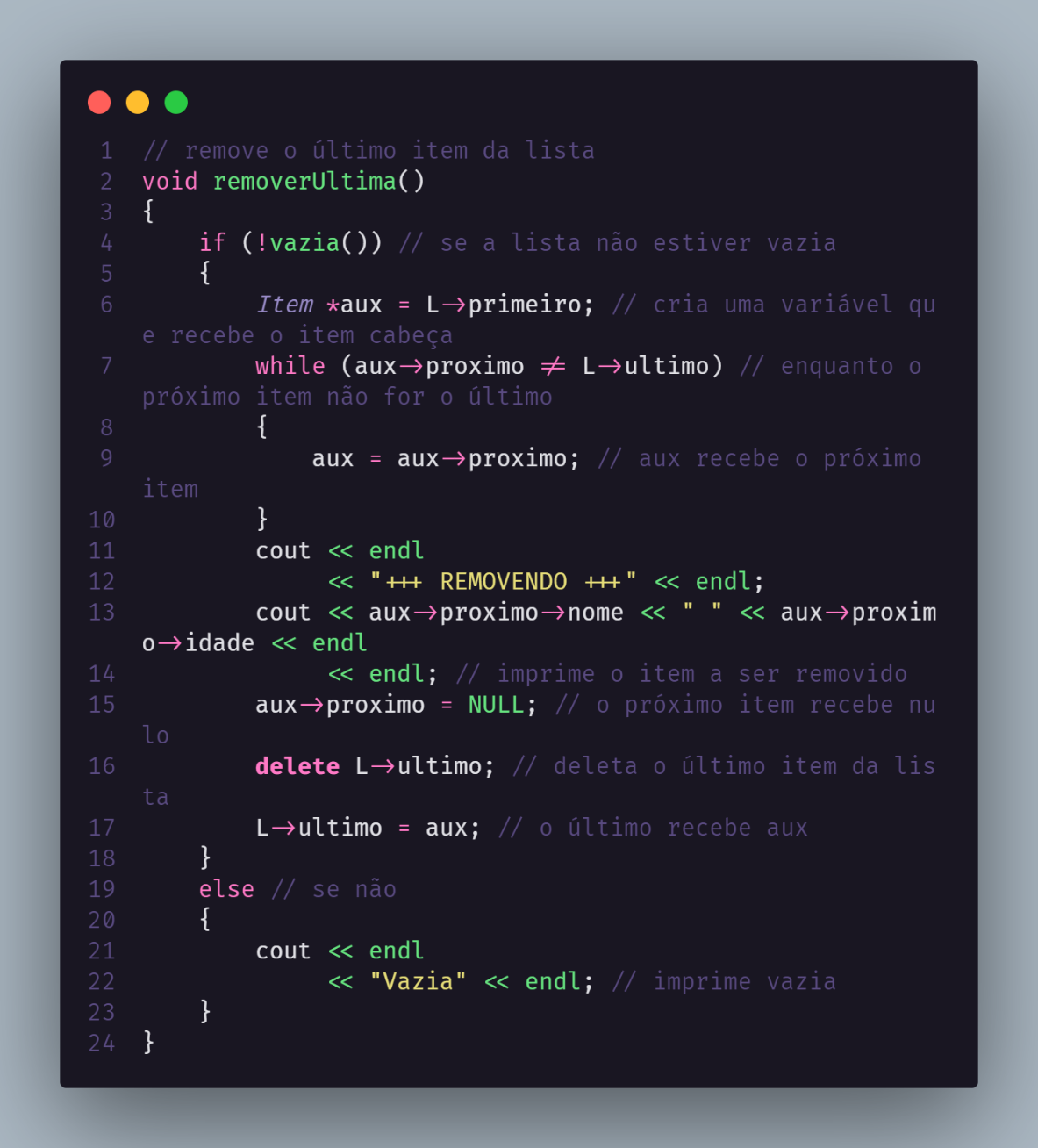
## Função que insere item em uma posição específica



## Função que remove o primeiro item da lista



## Função que remove o último item da lista



## Função que remove o item de uma posição específica



## Função menu



## Função main

****