ATIVIDADE 5

ESTRUTURA DE DADOS

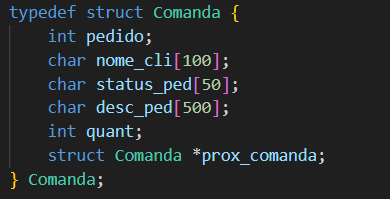
Mariana Alfonsetti Moderoso

01/04/25

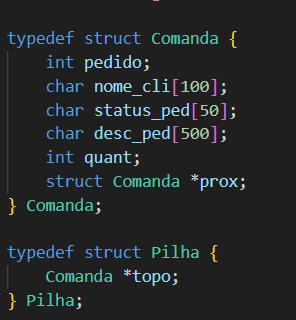
ALTERAÇÕES FEITAS PARA TRANSFORMAR UMA LISTA LIGADA EM PILHA

Nas estruturas eu adicionei mais uma struct para Pilha

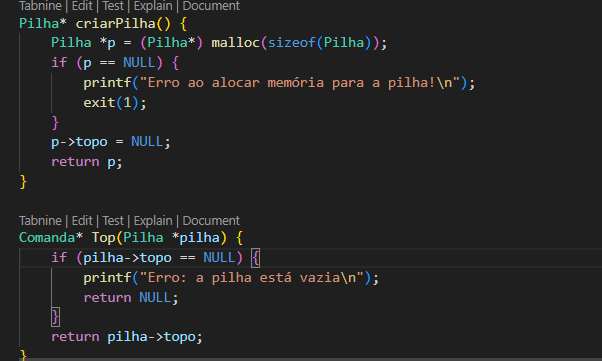
Código 1 - COMANDA EM LISTA LIGADA



Código 2 - COMANDA EM PILHA



Código 2 - EM PILHA (adicionei as funções de criar e ler a lista do topo)



Dessa forma eu conseguiria listar os pedidos do topo e ver se a pilha estava vazia ou não.

Essas mudanças em nomes e estrutura continuaram a ocorrer tanto na parte de inserir (PUSH) quanto na parte dos nós (que nas pilhas é como funcionará a ordem das pilhas...)

Como referenciado abaixo:

Código 1 - Em lista ligada

    nova\_comanda->pedido = pedido\_cli;

    strcpy(nova\_comanda->nome\_cli, nome);

    strcpy(nova\_comanda->status\_ped, status);

    strcpy(nova\_comanda->desc\_ped, desc);

    nova\_comanda->quant = quantidade;

    nova\_comanda->prox\_comanda = \*lista;

\*lista = nova\_comanda;

Código 2 - Em pilha

nova\_comanda->pedido = pedido\_cli;

    strcpy(nova\_comanda->nome\_cli, nome);

    strcpy(nova\_comanda->status\_ped, status);

    strcpy(nova\_comanda->desc\_ped, desc);

    nova\_comanda->quant = quantidade;

    nova\_comanda->prox = \*topo;

    \*topo = nova\_comanda;

Outra mudança significativa foi a alterar o “inserir comanda” para PUSH.

Código 1 - Em lista ligada

void inserircomanda(Comanda \*\*lista, int pedido\_cli, char nome[], char status[], char desc[], int quantidade) {

    Comanda \*nova\_comanda = (Comanda \*)malloc(sizeof(Comanda));

    if (nova\_comanda == NULL) {

        printf("Erro ao alocar memória\n");

        return;

    }

Código 2 - Em pilha

void push(Comanda \*\*topo, int pedido\_cli, char nome[], char status[], char desc[], int quantidade) {

    Comanda \*nova\_comanda = (Comanda \*)malloc(sizeof(Comanda));

    if (nova\_comanda == NULL) {

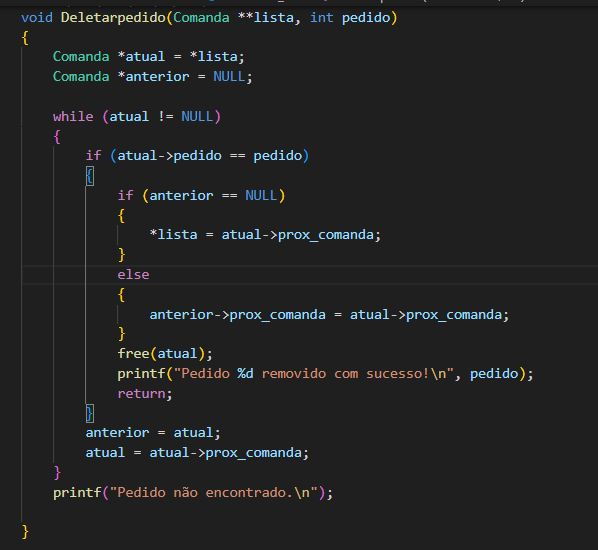
        printf("Erro ao alocar memória\n");

        return;

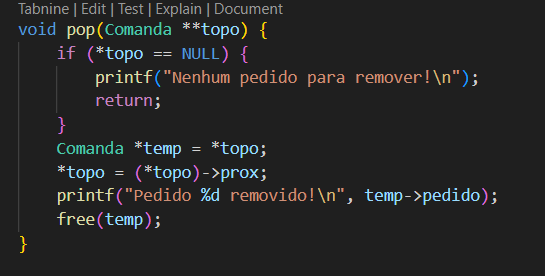
    }

Essas alterações afetaram diretamente em como eu manipulei os elementos, porque agora com essas nomenclaturas eu consegui refazer a estrutura.

A principal diferença aconteceu na parte de excluir pois não é possível excluir comandas de forma flexível quando estou fazendo em pilha:  
  
Em lista ligada o deletar pegava o pedido escolhido e removia da lista, com essa função:



Em pilha isso não é possível porque a função POP deleta o último pedido adicionado na pilha.



Outro ponto importante foi que eu adicionei uma forma de listar os pedidos em pilha, por que não adiantaria pesquisar se eu não soubesse onde exatamente estavam os pedidos e qual eu deletaria, o que como dito em sala de aula é extremamente ruim para esse tipo de sistema.