Código

#include <stdio.h>

#include <stdlib.h>

*// Protótipo*

typedef struct no No;

*// Definição do nó*

struct no {

    int dado;

    struct no \*prox;

};

*// Função para criar um novo nó*

No \*criar\_no(int dado) {

*// Alocação dinâmica do novo nó*

    No \*no = malloc(sizeof(No));

    if (no != NULL) {

        no->dado = dado;

        no->prox = NULL;

    }

    return no;

}

*// Função para listar os elementos da lista usando um loop while*

void listar\_while(No \*inicio) {

    No \*no = inicio;

    while (no != NULL) {

        printf("(%d)->", no->dado);

        no = no->prox;

    }

    printf("NULL\n");

}

*// Função para listar os elementos da lista usando um loop for*

void listar\_for(No \*inicio) {

    for (No \*no = inicio; no != NULL; no = no->prox) {

        printf("(%d)->", no->dado);

    }

    printf("NULL\n");

}

*// Função para deletar o nó no início da lista*

void deletar\_inicio(No \*\*inicio) {

    if (\*inicio != NULL) {

        No \*temp = \*inicio;

        \*inicio = (\*inicio)->prox;

        free(temp);

    }

}

*// Função para deletar o nó no fim da lista*

void deletar\_fim(No \*\*inicio) {

    if (\*inicio != NULL) {

        if ((\*inicio)->prox == NULL) { *// Se houver apenas um nó na lista*

            free(\*inicio);

            \*inicio = NULL;

        } else {

            No \*anterior = NULL;

            No \*atual = \*inicio;

            while (atual->prox != NULL) {

                anterior = atual;

                atual = atual->prox;

            }

            anterior->prox = NULL;

            free(atual);

        }

    }

}

*// Função para deletar um nó específico com um dado específico*

void deletar(No \*\*inicio, int dado) {

    if (\*inicio != NULL) {

        No \*anterior = NULL;

        No \*atual = \*inicio;

*// Encontra o nó a ser excluído*

        while (atual != NULL && atual->dado != dado) {

            anterior = atual;

            atual = atual->prox;

        }

*// Se o nó for encontrado*

        if (atual != NULL) {

*// Se for o primeiro nó*

            if (anterior == NULL) {

                \*inicio = atual->prox;

            } else {

                anterior->prox = atual->prox;

            }

            free(atual);

        }

    }

}

*// Função principal para testar as operações*

int main() {

    No \*inicio = criar\_no(1);

    inicio->prox = criar\_no(2);

    inicio->prox->prox = criar\_no(3);

    printf("Lista original:\n\n");

    listar\_while(inicio);

*// Deletar nó no início*

    deletar\_inicio(&inicio);

    printf("Lista apos deletar o no no inicio:\n\n");

    listar\_while(inicio);

*// Deletar nó no fim*

    deletar\_fim(&inicio);

    printf("Lista apos deletar o no no fim:\n\n");

    listar\_while(inicio);

*// Deletar nó com dado específico*

    deletar(&inicio, 2);

    printf("Lista apos deletar o no com dado especifico:\n\n");

    listar\_while(inicio);

    return 0;

}

TERMINAL

A screen shot of a computer

Description automatically generated