Curso: Ciência da Computação

Disciplina: Estrutura de Dados 1

Professor: Clayton Zambon

centro universitário unifacvest

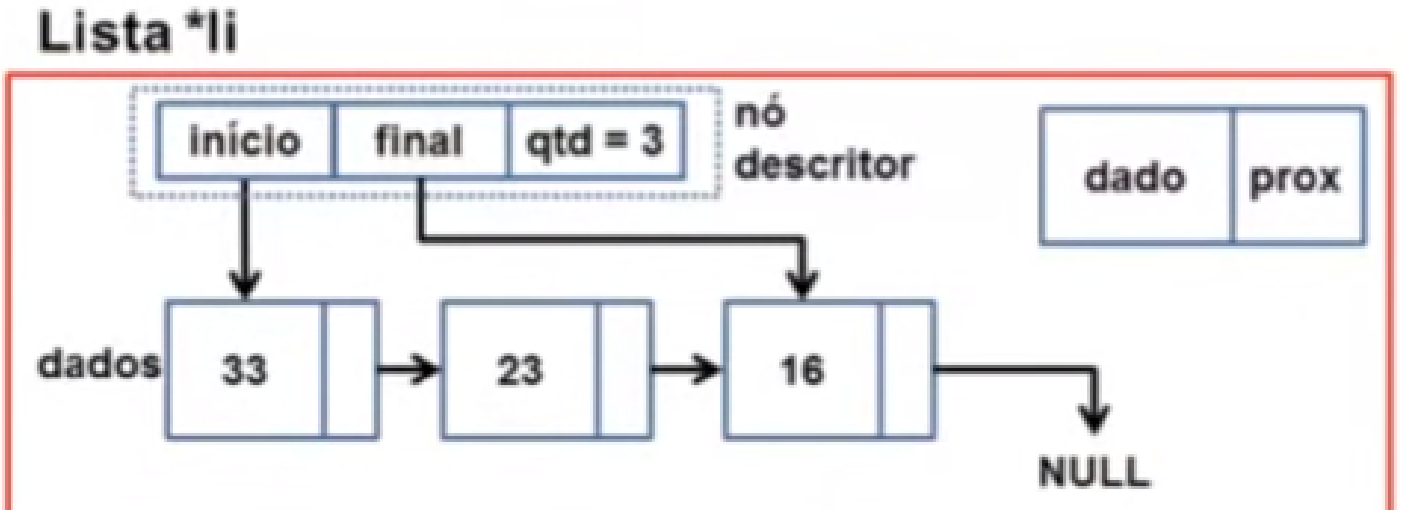
3. Lista

3.6.Dinâmica Encadeada com Nó Descritor;

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

Lista Dinâmica Encadeada

- Trata-se de uma variação da Lista Dinâmica Encadeada;
- Pode ser usada em Lista Simples, Duplamente Encadeada ou Circular;
- Usa um Nó Especial chamado de DESCRITOR para armazenar diversas informações sobre a lista;



3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- O Nó Descritor substitui o ponteiro para ponteiro que indica o início da lista;

- Ele permite armazenar informações como:

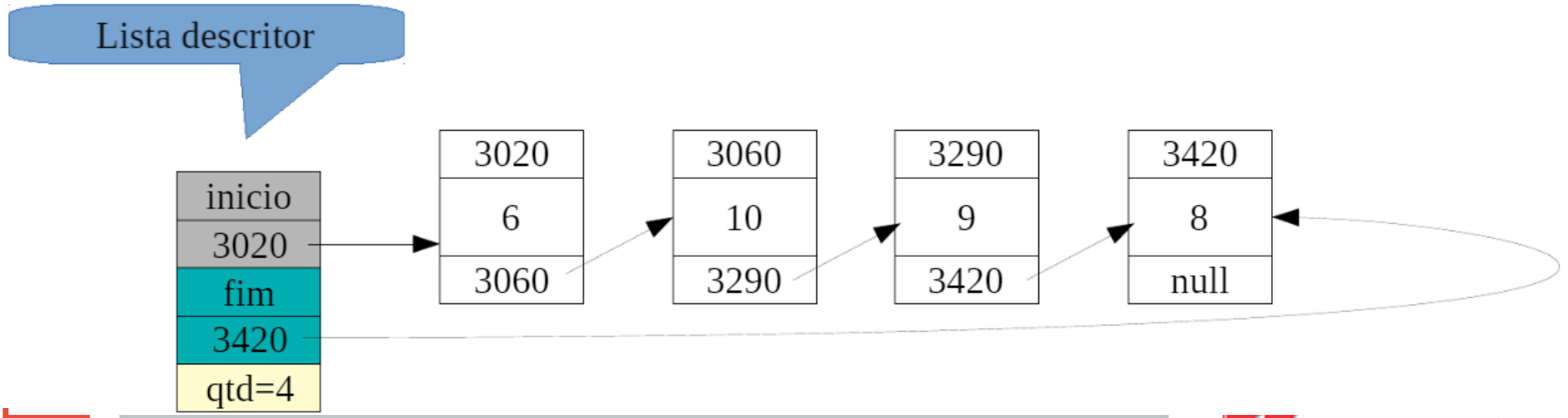
- Ponteiro para o início da Lista;
- Ponteiro para o final da Lista;
- Tamanho da Lista;

- Isto facilita e otimiza algumas operações;

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

Como Funciona?

- Há um ponteiro início que aponta para o primeiro elemento, e um Fim que aponta para o último;
- Um inteiro qtd, para definir quantidade de elementos da lista
- Elementos possuem sucessor (o último é null)



3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Vantagens:

- Melhor utilização dos recursos de memória;
- Não precisa movimentar os elementos nas operações de inserção e remoção;
- Adicionalmente, pode-se controlar o início e fim da lista;
- A função count é um O(1), ou seja, um passo;

- Desvantagens:

- Acesso indireto aos elementos;
- Necessidade de percorrer a lista para acessar um elemento;

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

Implementando um Lista Dinâmica Encadeada com Nó Descritor

- Arquivo "ListaDinEncadDesc.h": definir

- Protótipos das funções;
- O tipo de dados armazenado na lista;
- O ponteiro lista;

- Arquivo "ListaDinEncadDesc.c": definir

- O tipo de dados lista;
- O tipo de dados Descritor;
- Implementar as funções;

- Arquivo "main.c": definir

- Fazer as chamadas das funções;

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Definindo os Tipos, Structs e Ponteiro;

```c
struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,n3;
};
typedef struct descritor Lista;
```

```c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h"
int main(){
    Lista *li;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h" //inclui os protótipos
//Definição do tipo lista
struct elemento{
    struct aluno dados;
    struct elemento *prox;
};
typedef struct elemento Elem;

//Definição do Nó Descritor
struct descritor{
    struct elemento *inicio;
    struct elemento *final;
    int qtd;
};
```

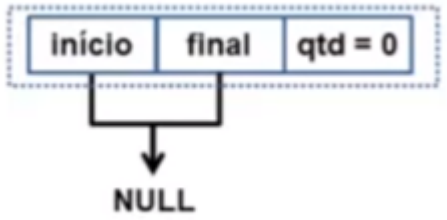
3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando a Função de "criar_lista";

```c
struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,n3;
};
typedef struct descritor Lista;
Lista* cria_lista();
```

```c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h"
int main(){
    Lista *li;
    li = cria_lista();
}
```

li




```c
//Arquivo ListaDinEncadDesc.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h" //inclui os protótipos
//Definição do tipo lista
struct elemento{
    struct aluno dados;
    struct elemento *prox;
};
typedef struct elemento Elem;

//Definição do Nó Descritor
struct descritor{
    struct elemento *inicio;
    struct elemento *final;
    int qtd;
};
//*****************************
Lista* cria_lista(){
    Lista* li = (Lista*) malloc(sizeof(Lista));
    if(li != NULL){
        li->inicio = NULL;
        li->final = NULL;
        li->qtd = 0;
    }
    return li;
}
```

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando a Função de

“liberar_lista”;

```c
//Arquivo ListaDinEncadDesc.h
struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,n3;
};
typedef struct descritor Lista;
Lista* cria_lista();
void libera_lista(Lista* li);
```

```c
//Arquivo main.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h"
int main(){
    Lista *li;
    li = cria_lista();
    libera_lista(li);
}
```

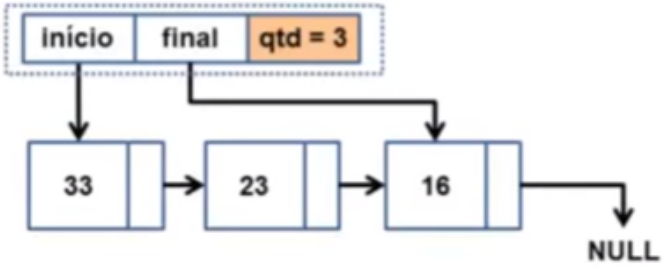
```c
//Arquivo ListaDinEncadDesc.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h" //inclui os protót
//Definição do tipo lista
struct elemento{
    struct aluno dados;
    struct elemento *prox;
};
typedef struct elemento Elem;

//Definição do Nó Descritor
struct descritor{
    struct elemento *inicio;
    struct elemento *final;
    int qtd;
};
//*****************************
void libera_lista(Lista* li){
    if(li != NULL){
        Elem* no;
        while((li->inicio) != NULL){
            no = li->inicio;
            li->inicio = li->inicio->prox;
            free(no);
        }
        free(li);
    }
}
```

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando a Função "tamanho_lista";

```c
//Arquivo ListaDinEncadDesc.h
struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,n3;
};
typedef struct descritor Lista;
int tamanho_lista(Lista* li);
```




```c
//Arquivo main.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h"
int main(){
    Lista *li;
    li = cria_lista();
    tamanho_lista(li);
    libera_lista(li);
}
```

```c
//Arquivo ListaDinEncadDesc.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h" //inclui
//Definição do tipo lista
struct elemento{
    struct aluno dados;
    struct elemento *prox;
};
typedef struct elemento Elem;

//Definição do Nó Descritor
struct descritor{
    struct elemento *inicio;
    struct elemento *final;
    int qtd;
};
//*****************************
int tamanho_lista(Lista* li){
    if(li == NULL)
        return 0;
    return li->qtd;
}
```

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando a Função “lista_vazia”;

```c
//Arquivo ListaDinEncadDesc.h
struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,n3;
};
typedef struct descritor Lista;
int lista_vazia(Lista* li);
```

```c
//Arquivo main.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h"
int main(){
    Lista *li;
    li = cria_lista();
    tamanho_lista(li);
    lista_vazia(li);
    libera_lista(li);
}
```

```c
//Arquivo ListaDinEncadDesc.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h" //inclui os pro
//Definição do tipo lista
struct elemento{
    struct aluno dados;
    struct elemento *prox;
};
typedef struct elemento Elem;

//Definição do Nó Descritor
struct descritor{
    struct elemento *inicio;
    struct elemento *final;
    int qtd;
};
//*****************************
int lista_vazia(Lista* li){
    if(li == NULL)
        return 1;
    if(li->inicio == NULL)
        return 1;
    return 0;
}
```

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando as funções de inserção. Podemos inserir:

- No início;
- No meio;
- No final: não há necessidade de percorrer a lista;

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando a Função “insere_lista_inicio”;

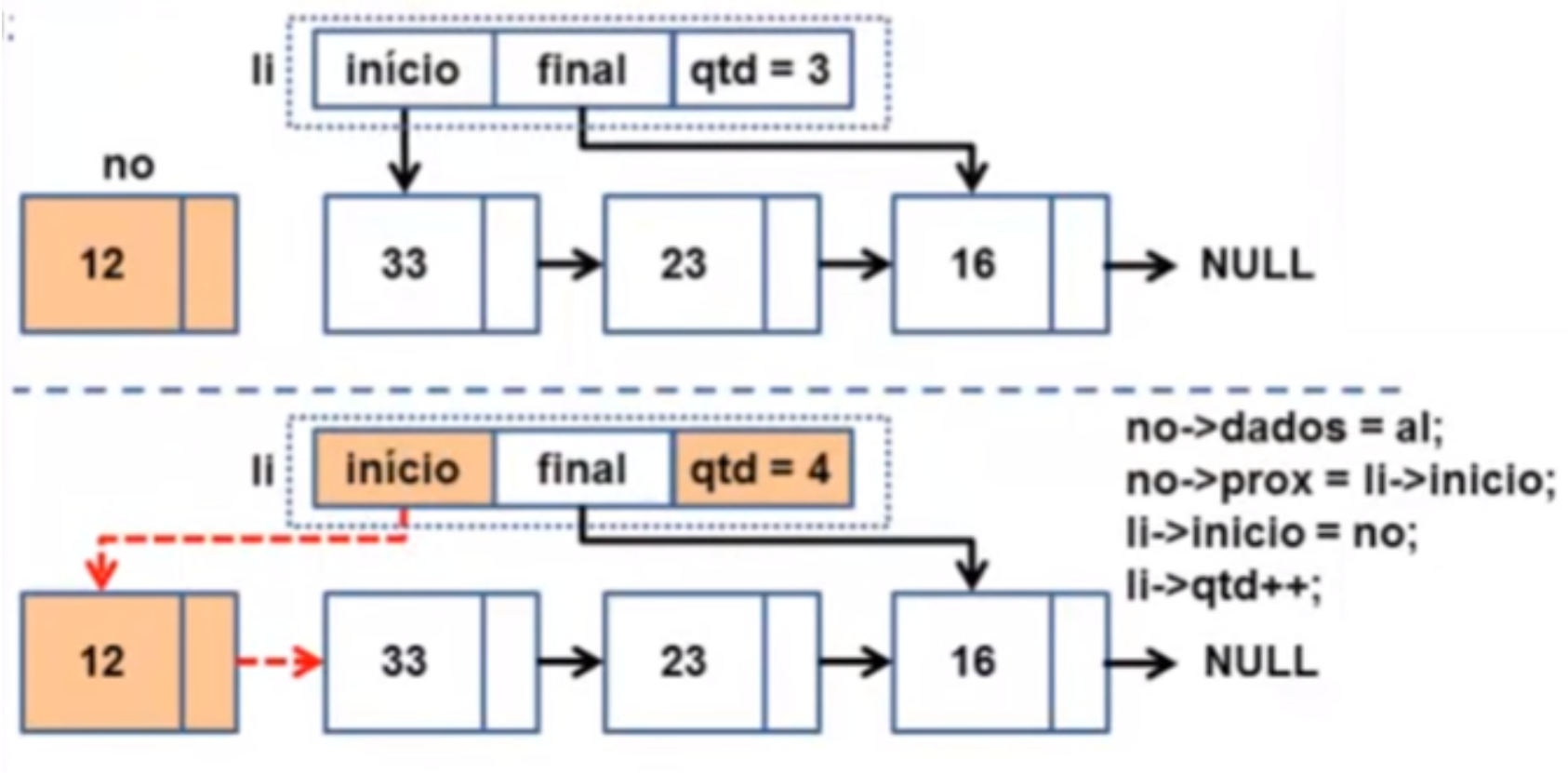
```c
//Arquivo ListaDinEncadDesc.h
struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,n3;
};
typedef struct descritor Lista;
int insere_lista_inicio(Lista* li, struct aluno al);
```

```c
//Arquivo main.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h"
int main(){
    Lista *li;
    struct aluno dados_aluno;
    li = cria_lista();
    int x = insere_lista_inicio(li, dados_aluno);
    tamanho_lista(li);
    lista_vazia(li);
    libera_lista(li);
}
```

```c
//Arquivo ListaDinEncadDesc.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h" //inclui os protótipos
//Definição do tipo lista
struct elemento{
    struct aluno dados;
    struct elemento *prox;
};
typedef struct elemento Elem;
//Definição do Nó Descritor
struct descritor{
    struct elemento *inicio;
    struct elemento *final;
    int qtd;
};
//***************************
int insere_lista_inicio(Lista* li, struct aluno al){
    if(li == NULL)
        return 0;
    Elem* no;
    no = (Elem*) malloc(sizeof(Elem));
    if(no == NULL)
        return 0;
    no->dados = al;
    no->prox = li->inicio;
    if(li->inicio == NULL)
        li->final = no;
    li->inicio = no;
    li->qtd++;
    return 1;
}
```

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando a Função “insere_lista_inicio”;



3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando a Função “insere_lista_final”;

```c
//Arquivo ListaDinEncadDesc.h
struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,n3;
};
typedef struct descritor Lista;
int insere_lista_final(Lista* li, struct aluno al);
```

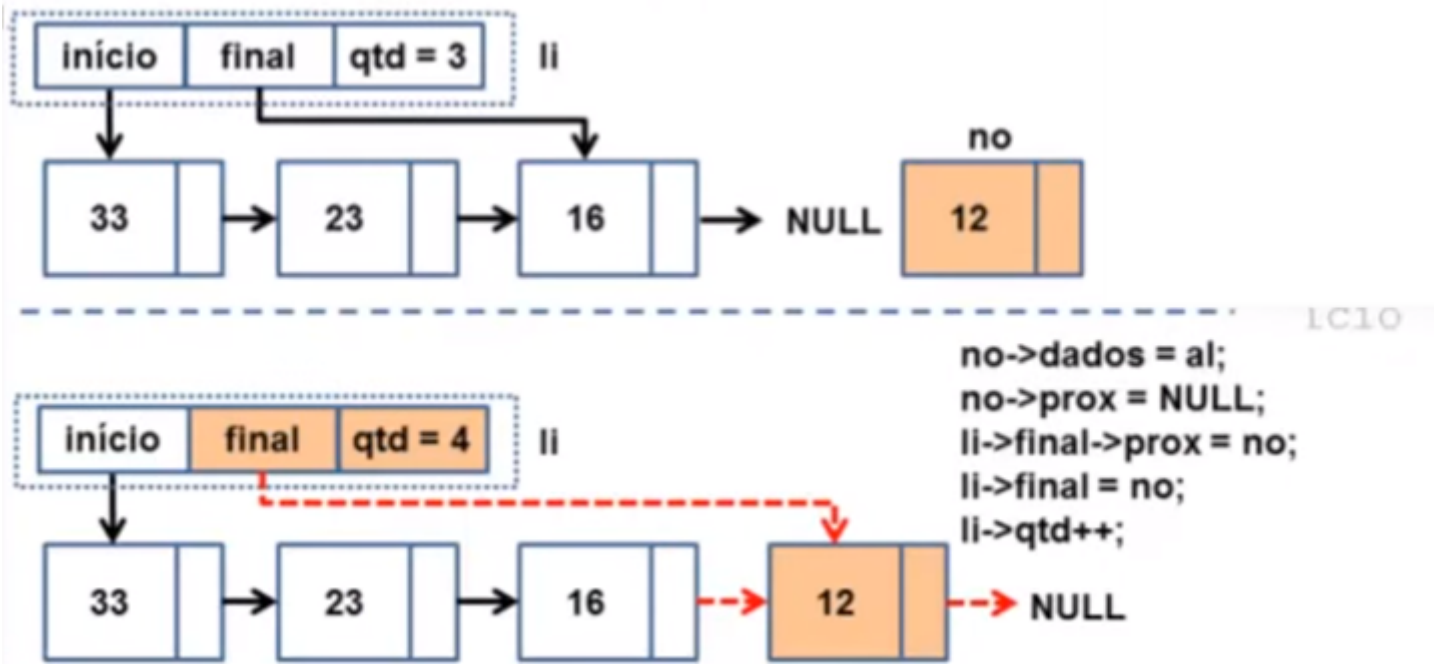
```c
//Arquivo main.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h"
int main(){
    Lista *li;
    struct aluno dados_aluno;
    li = cria_lista();
    int x = insere_lista_final(li, dados_aluno);
    tamanho_lista(li);
    lista_vazia(li);
    libera_lista(li);
}
```

```c
//Arquivo ListaDinEncadDesc.c
int insere_lista_final(Lista* li, struct aluno al){
    if(li == NULL)
        return 0;
    Elem *no;
    no = (Elem*) malloc(sizeof(Elem));
    if(no == NULL)
        return 0;
    no->dados = al;
    no->prox = NULL;
    if(li->inicio == NULL)//lista vazia: insere início
        li->inicio = no;
    else
        li->final->prox = no;

    li->final = no;
    li->qtd++;
    return 1;
}
```

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando a Função “insere_lista_final”;



3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando a Função “imprime_lista”;

```c
//Arquivo ListaDinEncadDesc.h
struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,n3;
};
typedef struct descritor Lista;
void imprime_lista(Lista* li);
```

```c
//Arquivo main.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h"
int main(){
    Lista *li;
    struct aluno dados_aluno;
    li = cria_lista();
    lista_vazia(li);
    int x = insere_lista_final(li, dados_aluno);
    imprime_lista(li);
    tamanho_lista(li);
    libera_lista(li);
}
```

```c
//*****************************
//Arquivo ListaDinEncadDesc.c
void imprime_lista(Lista* li){
    if(li == NULL || li->inicio == NULL)
        return;
    Elem* no = li->inicio;
    while(no != NULL){
        printf("Matricula: %d\n",no->dados.matricula);
        printf("Nome: %s\n",no->dados.nome);
        printf("Notas: %f %f %f\n",no->dados.n1,
                                   no->dados.n2,
                                   no->dados.n3);
        printf("--------------------------------\n");
        no = no->prox;
    }
}
```

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

EXERCÍCIO:

- Fazer um menu para acessar as funções;

- Fazer as implementações descritas anteriormente;

- Popular a lista com informações do aluno na lista;

- Implementar mensagens para informar o usuário as ações que foram realizadas;

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando as funções de REMOÇÃO. Podemos REMOVER:

- Do início;
- Do meio;
- Do final;

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando a Função “remove_lista_inicio”;

```c
//Arquivo ListaDinEncadDesc.h
struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,n3;
};
typedef struct descritor Lista;
int remove_lista_inicio(Lista* li);
```

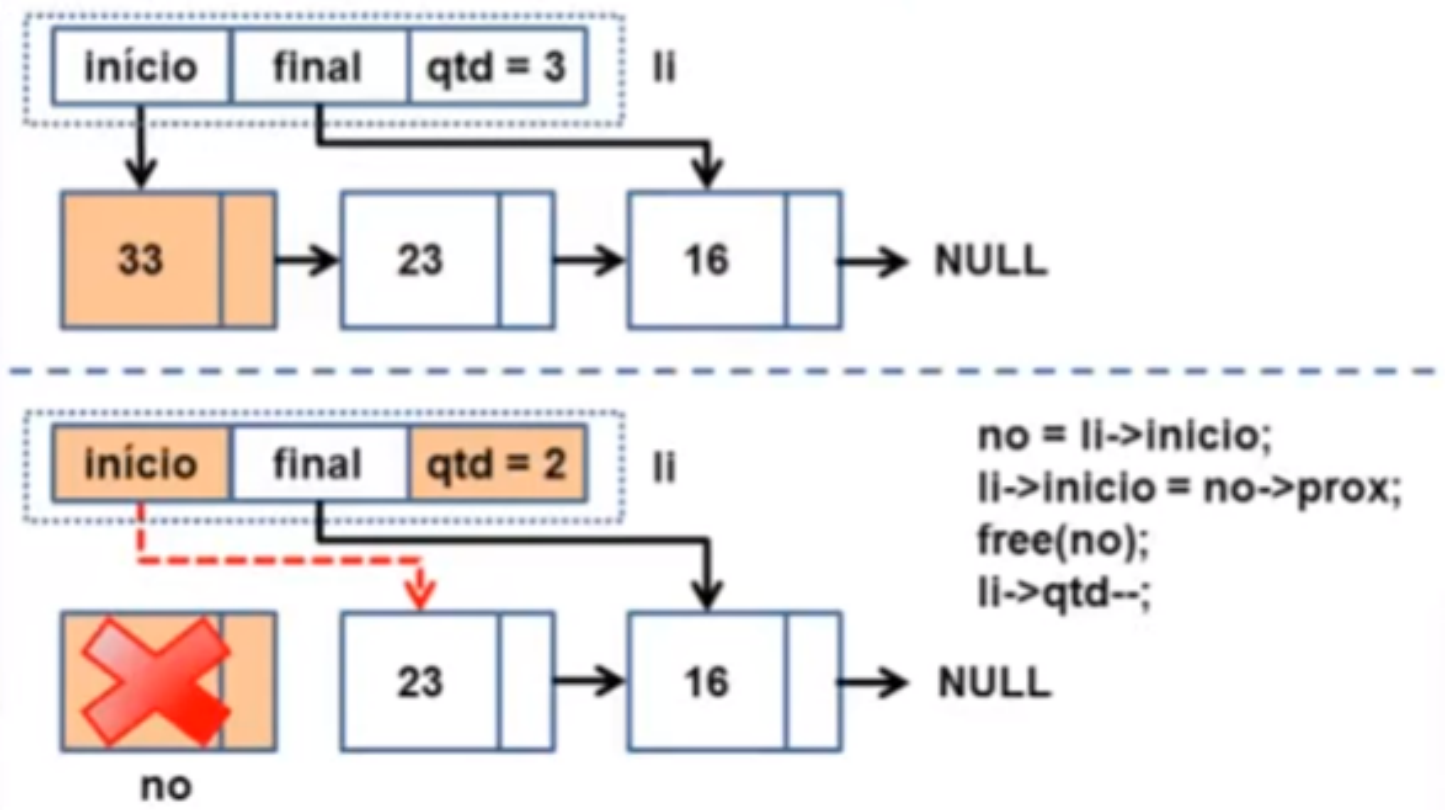
```c
//Arquivo main.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h"
int main(){
    Lista *li;
    struct aluno dados_aluno;
    li = cria_lista();
    lista_vazia(li);
    int x = insere_lista_final(li, dados_aluno);
    imprime_lista(li);
    tamanho_lista(li);
    x = remove_lista_inicio(li);
    libera_lista(li);
}
```

```c
//*****************************
//Arquivo ListaDinEncadDesc.c
int remove_lista_inicio(Lista* li){
    if(li == NULL)
        return 0;
    if(li->inicio == NULL)//lista vazia
        return 0;

    Elem *no = li->inicio;
    li->inicio = no->prox;
    free(no);
    if(li->inicio == NULL)
        li->final = NULL;
    li->qtd--;
    return 1;
}
```

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando a Função “remove_lista_inicio”;



3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando a Função “remove_lista_final”;

```c
//Arquivo ListaDinEncadDesc.h
struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,n3;
};
typedef struct descritor Lista;
int remove_lista_final(Lista* li);
```

```c
//Arquivo main.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h"
int main(){
    Lista *li;
    struct aluno dados_aluno;
    li = cria_lista();
    lista_vazia(li);
    int x = insere_lista_final(li, dados_aluno);
    imprime_lista(li);
    tamanho_lista(li);
    x = remove_lista_final(li);
    libera_lista(li);
}
```

```c
//*****************************
//Arquivo ListaDinEncadDesc.c
int remove_lista_final(Lista* li){
    if(li == NULL)
        return 0;
    if(li->inicio == NULL)//lista vazia
        return 0;

    Elem *ant, *no = li->inicio;
    while(no->prox != NULL){
        ant = no;
        no = no->prox;
    }
    if(no == li->inicio){//remover o primeiro?
        li->inicio = NULL;
        li->final = NULL;
    }else{
        ant->prox = no->prox;
        li->final = ant;
    }
    free(no);
    li->qtd--;
    return 1;
}
```

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando as funções de CONSULTA. Podemos CONSULTAR um elemento de uma lista de duas formas:

- Pela posição;
- Pelo Conteúdo;
- Ambos dependem de busca sendo necessário percorrer a lista até encontrar o elemento desejado;

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando a Função “consulta_lista_pos”;

```c
int consulta_lista_pos(Lista* li, int pos, struct aluno *al){
    if(li == NULL || li->inicio == NULL || pos <= 0)
        return 0;
    Elem *no = li->inicio;//primeiro elemento
    int i = 1;
    while(no != NULL && i < pos){
        no = no->prox;
        i++;
    }
    if(no == NULL)
        return 0;
    else{
        *al = no->dados;
        return 1;
    }
}
```

```c
//Arquivo main.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h"
int main(){
    Lista *li;
    struct aluno dados_aluno;
    int = posicao;
    li = cria_lista();
    lista_vazia(li);
    int x = insere_lista_final(li, dados_aluno);
    x = consulta_lista_pos(li, posicao, dados_aluno);
    imprime_lista(li);
    tamanho_lista(li);
    x = remove_lista_final(li);
    libera_lista(li);
}
```

```c
//Arquivo ListaDinEncadDesc.h
struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,n3;
};
typedef struct descritor Lista;
int consulta_lista_pos(Lista* li, int pos, struct aluno *al);
```

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

- Implementando a Função “consulta_lista_mat”;

```c
int consulta_lista_mat(Lista* li, int mat, struct aluno *al){
    if(li == NULL || li->inicio == NULL)
        return 0;
    Elem *no = li->inicio;
    while(no != NULL && no->dados.matricula != mat)
        no = no->prox;
    if(no->dados.matricula != mat)
        return 0;
    else{
        *al = no->dados;
        return 1;
    }
}
```

```c
//Arquivo main.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaDinEncadDesc.h"
int main(){
    Lista *li;
    struct aluno dados_aluno;
    int = posicao, matricula;
    li = cria_lista();
    lista_vazia(li);
    int x = insere_lista_final(li, dados_aluno);
    x = consulta_lista_mat(li, matricula, dados_aluno);
    imprime_lista(li);
    tamanho_lista(li);
    x = remove_lista_final(li);
    libera_lista(li);
}
```

```c
//Arquivo ListaDinEncadDesc.h
struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,n3;
};
typedef struct descritor Lista;
int consulta_lista_mat(Lista* li, int mat, struct aluno *al);
```

3. Lista
3.6. Dinâmica Encadeada com Nó Descritor

EXERCÍCIO:

- Ao consultar pela posição ou matrícula, exibir as informações do elemento;

- Quando não encontrar um elemento exibir uma mensagem para o usuário;

- Quando não for possível realizar a operação, exibir a mensagem de lista vazia ou lista cheia conforme o caso.

- Quando inserir ou remover um elemento informar ao usuário se a operação ocorreu com sucesso ou não.

Referências

EDELWEISS, Nina; GALANTE, Renata. Estruturas de Dados. Porto Alegre, BOOKMAN, 2009.

HEINZLE, Roberto. Estruturas de Dados: implementações com C e Pascal. Blumenau, DIRETIVA, 2006.

TENENBAUM, Aron M. Estrutura de Dados usando C. São Paulo, Makron Books, 1995.

FORBELLONE, André Luiz Villar: EBERSPÄCHER, Henri Frederico. Lógica de Programação: a construção de algoritmos e estruturas de dados. 3. ed. São Paulo, PRENTICE HALL, 2005.

KOFFMAN, Elliot B.; WOLFGANG, Paul A. T. Objetos, abstração, estruturas de dados e projeto usando C++. Rio de Janeiro, LTC, 2008.

PEREIRA, Silvio do lago. Estruturas de dados fundamentais: conceitos e aplicações. São Paulo, Érica, 1996.

VILLAS, Marcos Viana et al. Estruturas de dados – Conceitos e técnicas de implementação. Rio de Janeiro, Campus, 1993.

VELOSO, Paulo et al. Estrutura de dados. Rio de Janeiro, Campus, 1996.

Canal do Youtube: Linguagem C Programação Descomplicada

Obrigado!