

Computação 2

Aula 10

Listas Encadeadas

Prof^a. Fabiany
fabiany1@utfpr.edu.br

```

#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
#include <string.h>

#define TAM 2

struct Elemento
{
    char nome [100];
    char rua [100];
    char cidade [100];
    char estado [2];
    char cep [10];

    struct Elemento *proximo;
};

struct Elemento *primeiro;
struct Elemento *ultimo;

char menu ();

void inicia_lista ();

void cadastra ();

void mostra ();

void salva ();

void carrega ();

void limpaLista ();

```

```

int main()
{
    char escolha;
    inicia_lista ();

    for ( ; ; )
    {
        escolha = menu ();
        switch ( escolha )
        {
            case 'c' :
            case 'C' : { cadastra(); } break;

            case 'm' :
            case 'M' : { mostra(); } break;

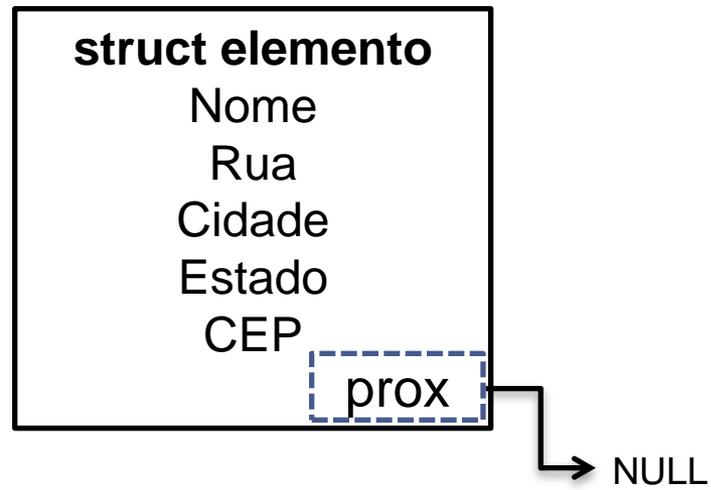
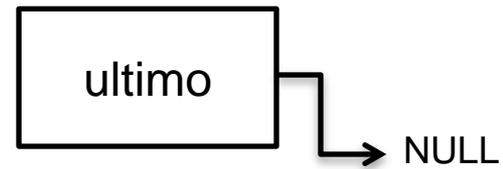
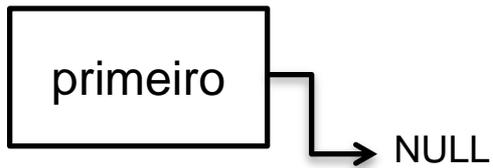
            case 't' :
            case 'T' : {
                limpaLista ();
                exit ( 0 );
            } break;
            default : { printf ( "Opcao invalida. \n" ); }
        }

        printf ( "\n \n \n" );
    }

    system ( "Pause" );
    return 0;
}

void inicia_lista ()
{
    primeiro = NULL;
    ultimo = NULL;
}

```



```

void cadastra ()
{
    system ( "cls" );
    printf ( "\n \n \n" );

    struct Elemento* novo;
    novo = malloc ( 1 * sizeof (struct Elemento) );

    novo->proximo = NULL;

    printf ( "Nome: \n" );
    fflush ( stdin );      gets ( novo->nome );

    printf ( "Rua: \n" );
    fflush ( stdin );      gets ( novo->rua );

    printf ( "Cidade: \n" );
    fflush ( stdin );      gets ( novo->cidade );

    printf ( "Estado: \n" );
    fflush ( stdin );      gets ( novo->estado );

    printf ( "CEP: \n" );
    fflush ( stdin );      gets ( novo->cep );

    if ( NULL == primeiro )
    {
        primeiro = novo;
        ultimo   = primeiro;
    }
    else
    {
        ultimo->proximo = novo;
        ultimo = novo;
    }
}

```

```

void mostra ()
{
    system ( "cls" );

    printf ( "\n \n \n" );

    struct Elemento* aux;

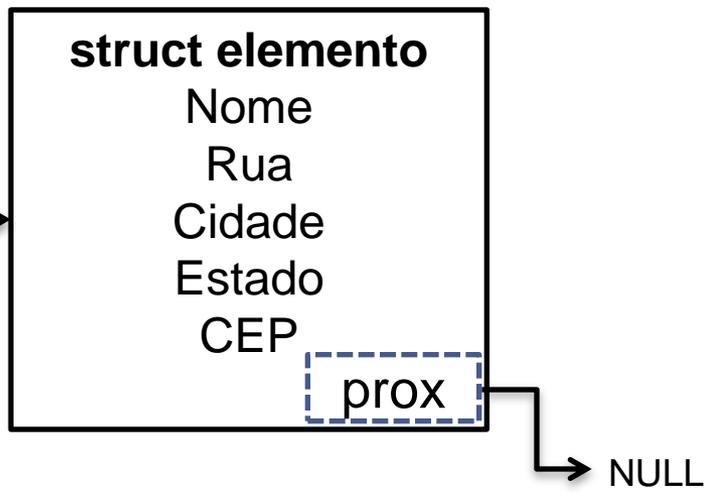
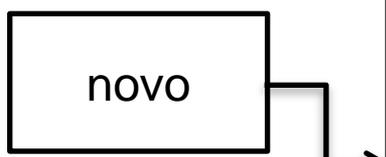
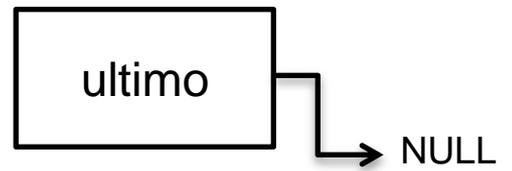
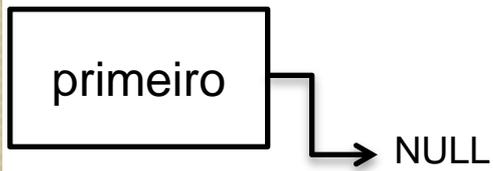
    aux = primeiro;

    while ( aux != NULL )
    {
        printf ("%s \n", aux->nome      );
        printf ("%s \n", aux->rua      );
        printf ("%s \n", aux->cidade   );
        printf ("%s \n", aux->estado   );
        printf ("%s \n", aux->cep      );
        printf ("\n");

        aux = aux->proximo;

    }
}

```



```

void cadastra ()
{
    system ( "cls" );
    printf ( "\n \n \n" );

    struct Elemento* novo;
    novo = malloc ( 1 * sizeof (struct Elemento) );

    novo->proximo = NULL;

    printf ( "Nome: \n" );
    fflush ( stdin );      gets ( novo->nome );

    printf ( "Rua: \n" );
    fflush ( stdin );      gets ( novo->rua );

    printf ( "Cidade: \n" );
    fflush ( stdin );      gets ( novo->cidade );

    printf ( "Estado: \n" );
    fflush ( stdin );      gets ( novo->estado );

    printf ( "CEP: \n" );
    fflush ( stdin );      gets ( novo->cep );

    if ( NULL == primeiro )
    {
        primeiro = novo;
        ultimo   = primeiro;
    }
    else
    {
        ultimo->proximo = novo;
        ultimo = novo;
    }
}

```

```

void mostra ()
{
    system ( "cls" );

    printf ( "\n \n \n" );

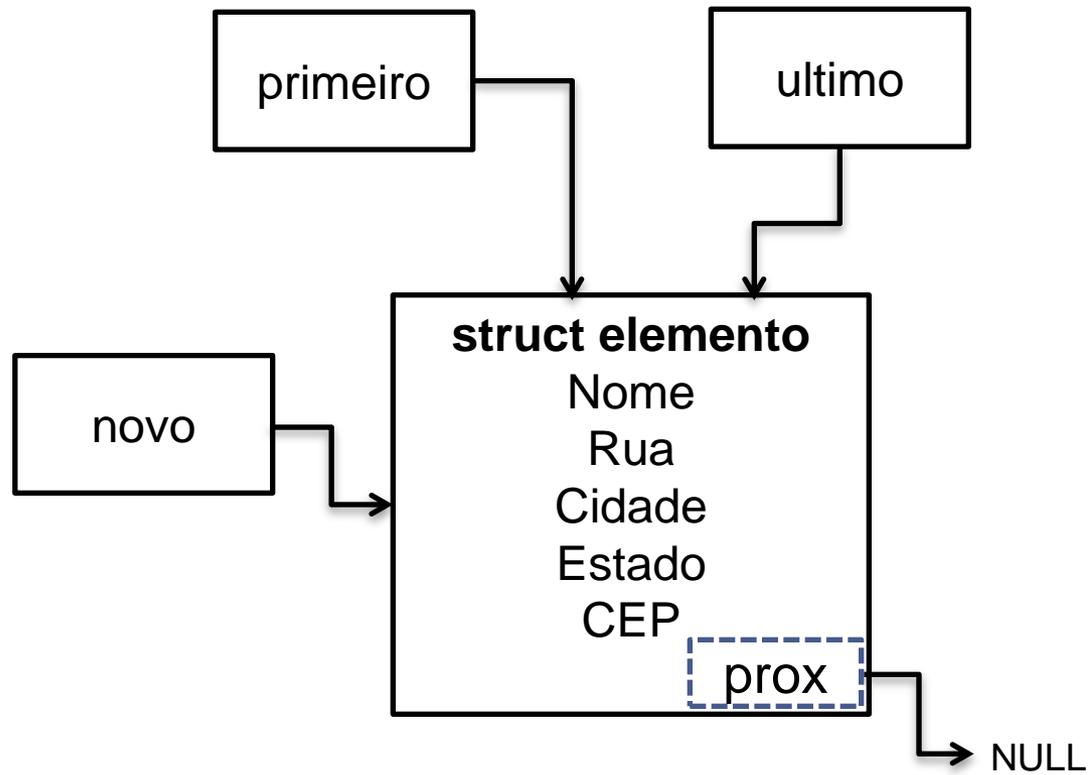
    struct Elemento* aux;

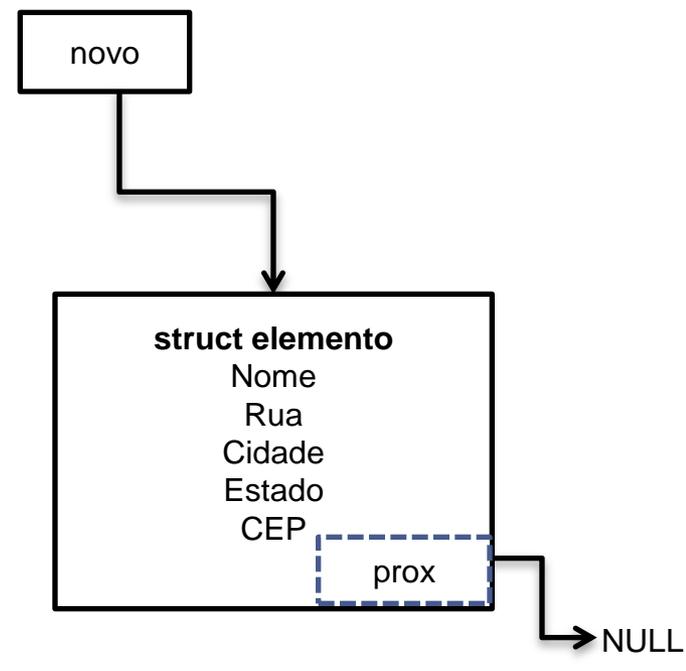
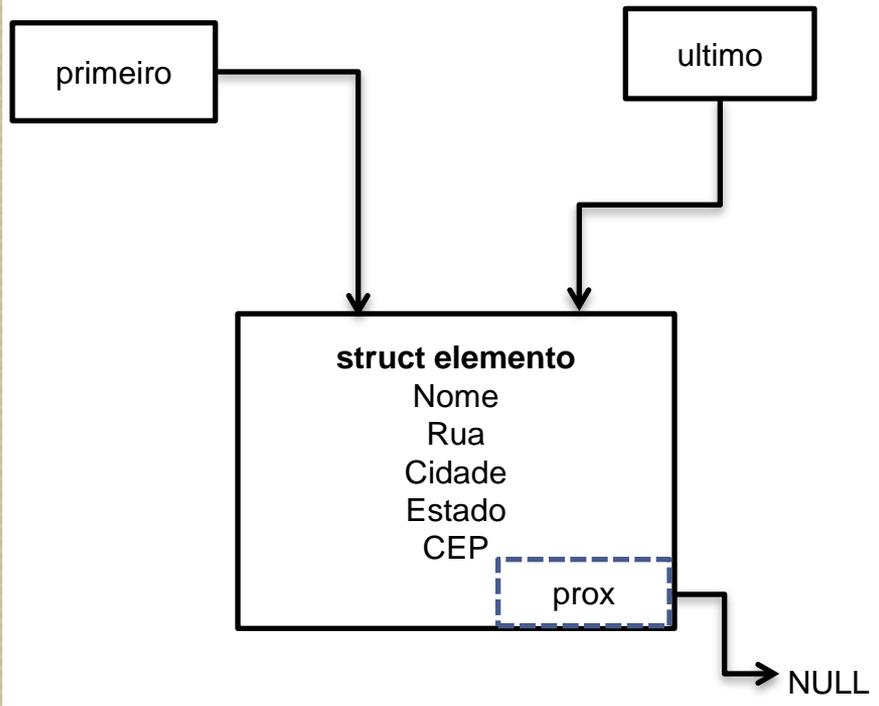
    aux = primeiro;

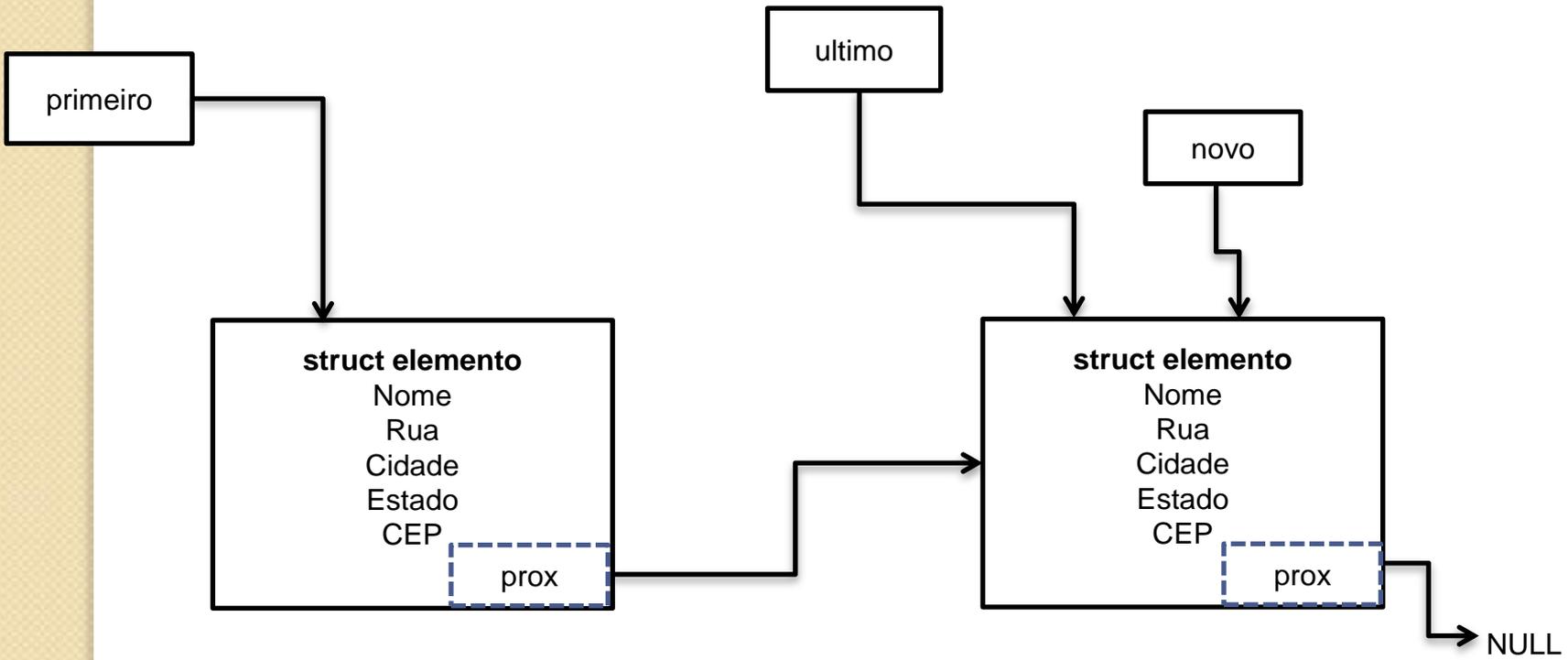
    while ( aux != NULL )
    {
        printf ("%s \n", aux->nome      );
        printf ("%s \n", aux->rua      );
        printf ("%s \n", aux->cidade    );
        printf ("%s \n", aux->estado    );
        printf ("%s \n", aux->cep      );
        printf ("\n");

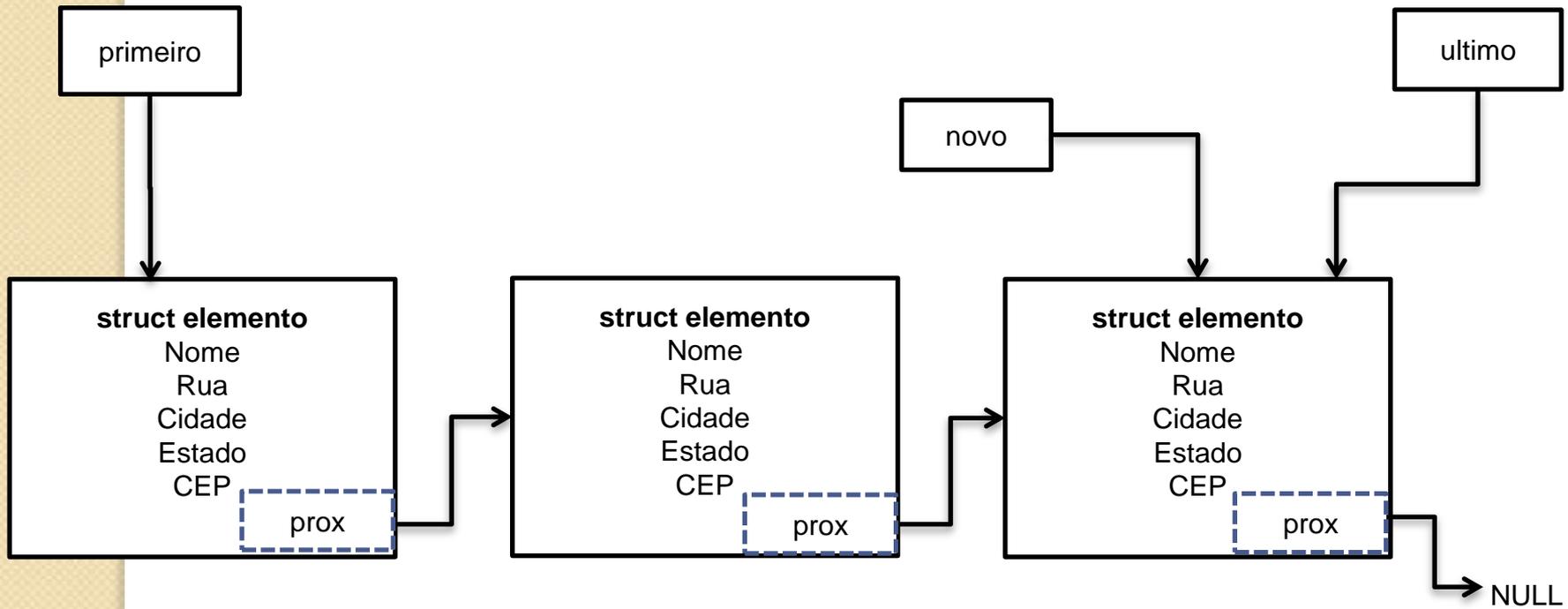
        aux = aux->proximo;
    }
}

```









```
void limpaLista ()
{
    struct Elemento* aux;

    aux = primeiro;

    while ( aux != NULL )
    {

        primeiro = primeiro->proximo;

        free ( aux );

        aux = primeiro;

    }

    primeiro = NULL;
    ultimo  = NULL;

}
```

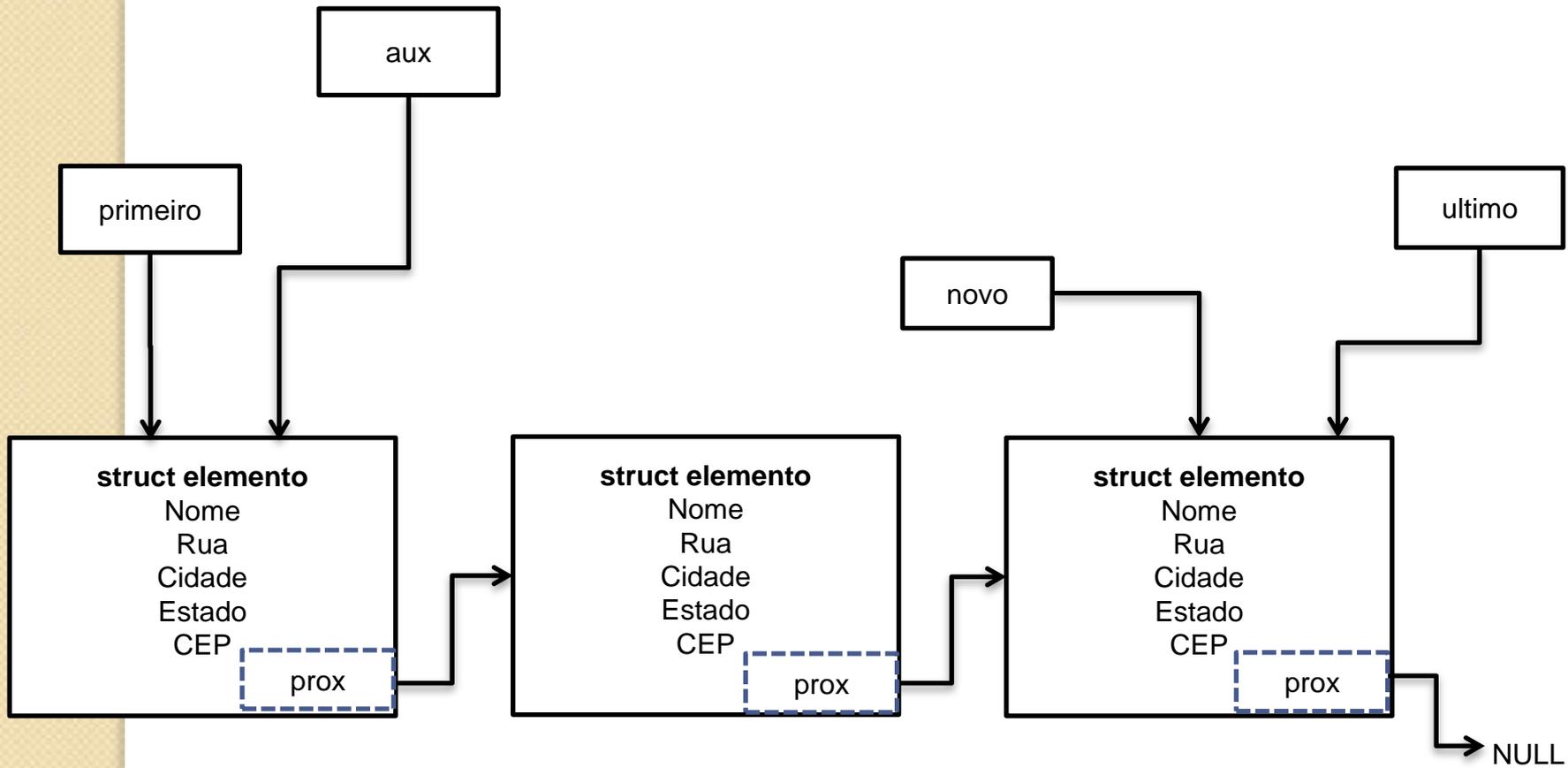
```
char menu ()
{
    printf ("\n \n \n");
    char opcao;

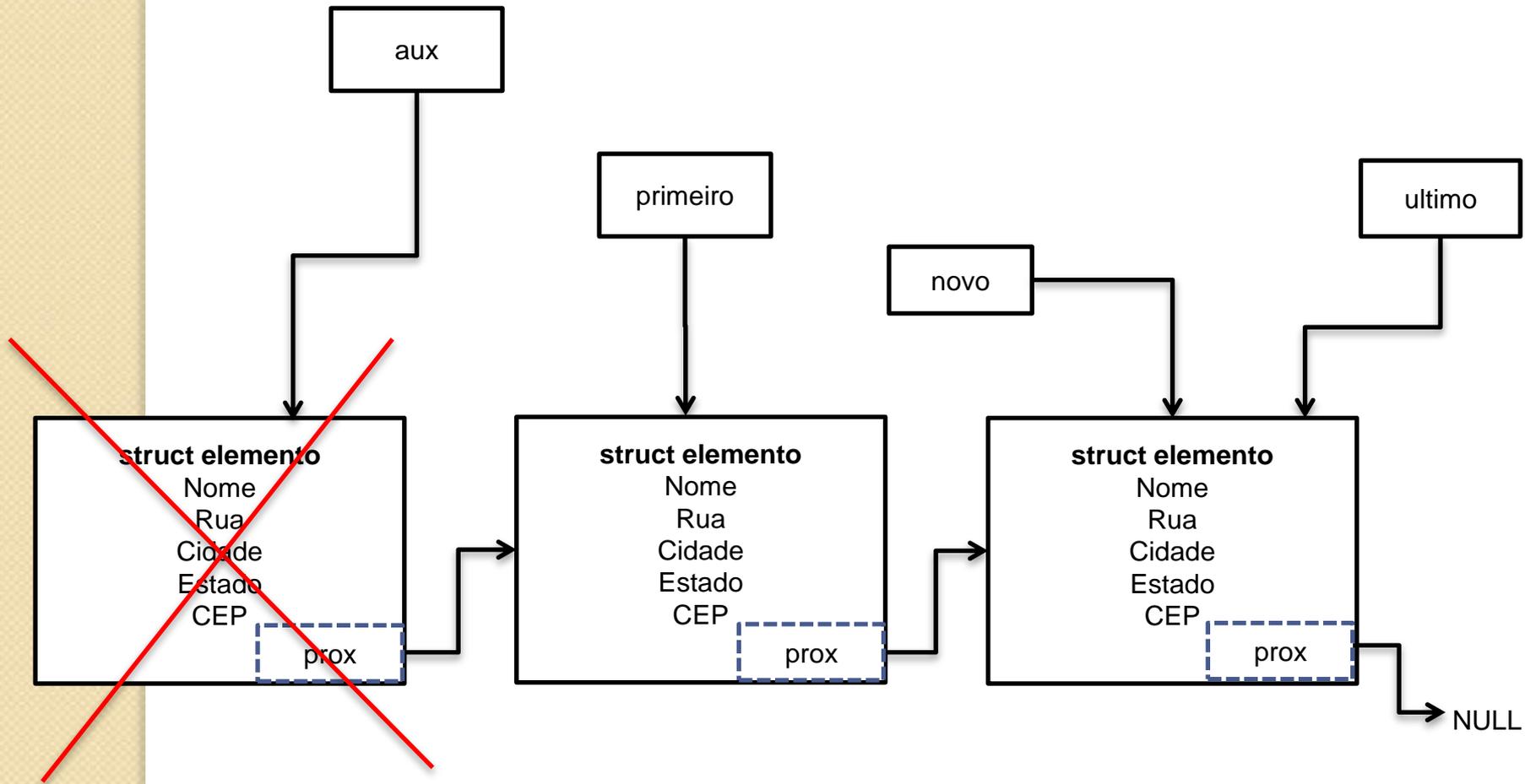
    printf ( "(C)adastrar. \n" );
    printf ( "(M)ostrar.  \n" );
    printf ( "(T)erminar. \n" );

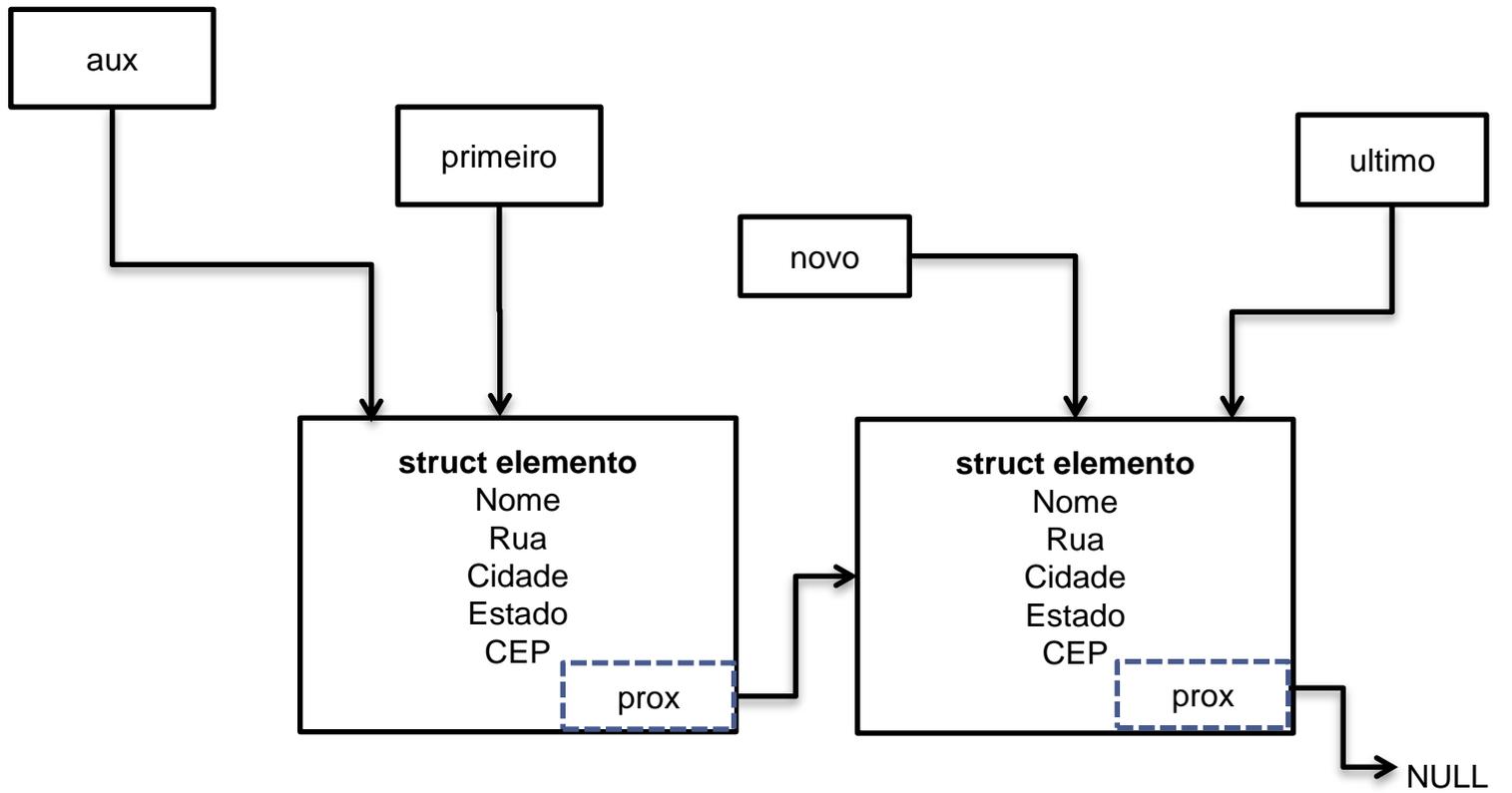
    fflush ( stdin );
    scanf ( "%c", &opcao );

    return opcao;

}
```







Exercícios

- 1) Fazer um programa para inserir números usando listas encadeadas, a lista sempre deve estar em ordem crescente. O programa deve permitir a exclusão de números mantendo a ordem.
- 2) Refazer o exercício a seguir usando listas encadeadas:
 - a) Faça um programa que realize o cadastro de contas bancárias com as seguintes informações: número da conta, nome do cliente e saldo. O banco permitirá o cadastramento de várias contas e não poderá haver mais que uma conta com o mesmo número. Crie o menu de opções a seguir:
 - 1) Cadastrar conta;
 - 2) Fazer movimentações na conta (saque e depósito);
 - 3) Buscar cliente por conta e mostrar as informações;
 - 4) Excluir uma conta;
 - 5) Sair.

Referências Bibliográficas

- Baseado nos slides do Professor Jean Simão disponível em:
<http://www.pessoal.utfpr.edu.br/jeansimao/Fundamentos1/Fundamentos1.htm>
- **SHILDT**, H. C, Completo e Total, 3a edição, rev. e atual. Ed. Makron. São Paulo, c1997.