

# LISTAS - CONTINUAÇÃO

1

Profa. Fabrícia Damando Santos  
[fabriadamando@gmail.com](mailto:fabriadamando@gmail.com)

ATÉ AGORA:

Lista com 1-2 elementos

Lista com função:para criar a lista  
e para imprimir

# Par criar a função

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#define MAX 5
```

```
void cria_lista(struct nodo **inicio, struct nodo **fim)
{
    *inicio=NULL;
    *fim=NULL;
}
```



# Para imprimir

```
void mostra_lista(struct nodo *inicio)
{
    struct nodo *aux;
    aux=inicio;
    printf("\n");
    printf("lista==>");
    while (aux != NULL)
    {
        printf("%i", aux->dados, "--");
        aux = aux-> proximo;
    }
}
```



Lembre que sempre é necessário imprimir: endereço do nó, seu valor e o endereço do próximo nó.

PRÓXIMA META:

Criar mais de um nó  
de forma melhorada???

Com funções

Com inserção mais a direita / fim.

# CONTINUANDO ....CRIAÇÃO DA LISTA

```
procedimento Cria_Lista (ref registro nodo *lista);  
inicio
```

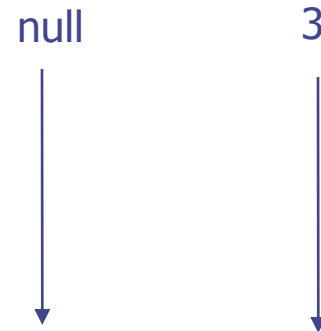
```
    lista = nulo;
```

Lista  $\longrightarrow$  nulo

```
fim
```

# Inserir - lista a direita = inserir no fim

**Primeira chamada da procedure para inserir um valor à direita na lista simplesmente encadeada (= inserir no final)**



Procedimento `insere_dir (ref registro nodo *primeiro, int valor);`

# LISTA ENCADEADA – INSERÇÃO A DIREITA

Procedimento insere\_dir (ref registro nodo \*primeiro, inteiro valor);

registro nodo \*p, \*aux ;

inicio

aloca (p);

se (p = nulo) { se não conseguiu alocar memória }

senao inicio

P->dado = valor;

p->proximo = nulo;

fim

se (primeiro = nulo) { se não existem valores na lista ainda.. }

entao primeiro = p;

senao inicio

aux = primeiro

enquanto (aux -> proximo <> nulo)

aux = aux->proximo;

p->proximo = aux;

fim;

fim;

primeiro → null

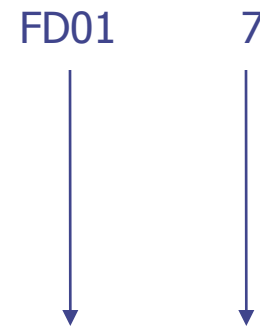
Valor  
3



p



Segunda chamada da procedure para inserir um valor à direita na lista simplesmente encadeada



Procedimento `insere_dir (ref registro nodo *primeiro, int valor);`

# LISTA ENCADEADA – INSERÇÃO A DIREITA

```
Procedimento insere_dir (ref registro nodo *primeiro; inteiro valor);  
registro nodo *p, *aux ;  
inicio
```

```
se (p = nulo) { se não conseguiu alocar memória }  
senao inicio
```

```
    P->dado = valor;  
    p->proximo = nulo;
```

```
fim
```

```
se (primeiro = nulo) { se não existem valores na lista ainda.. }
```

Valor  
7

```
entao primeiro = p;
```

```
senao inicio
```

```
    aux = primeiro
```

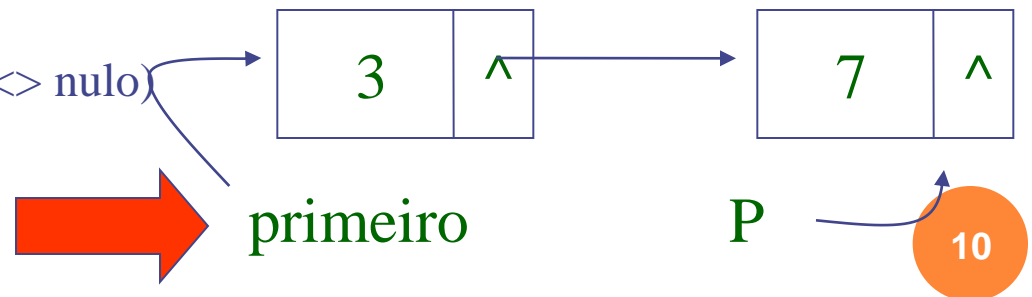
```
    enquanto (aux -> proximo <> nulo)
```

```
        aux = aux->proximo;
```

```
    p->proximo = aux;
```

```
    fim;
```

```
    fim;
```



# Portanto

- Crie a estrutura
- Cria a função de inserção a direita (fim).
- Cria a função imprime



# EXERCÍCIO

Utilizando listas simplesmente encadeadas escreva um programa capaz de:

- a) Inserir fim;
- b) Imprimir;