

# Laboratório de Programação II

## Listas Encadeadas

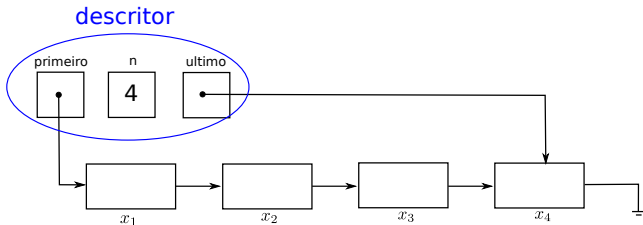
Universidade Federal de Juiz de Fora  
Departamento de Ciência da Computação

# Aula de Hoje

- ▶ Listas
  - ▶ Contíguas
  - ▶ **Encadeadas**

# Lista Encadeada

- ▶ TAD para uma **Lista Simplesmente Encadeada**
- ▶ Nesta estrutura de dados, um nó deve conter além de seu valor, uma indicação (ponteiro ou apontador) para o nó seguinte, representando uma contiguidade lógica.



# Nó da Lista Encadeada

```
class No
{
private:
    int info; // valor armazenado no No
    No *prox; // ponteiro para o proximo No

public:
    No ()                { };
    ~No ()               { };
    int getInfo()         { return info; };
    No* getProx()         { return prox; };
    void setInfo(int val) { info = val; };
    void setProx(No *p)   { prox = p; };
};
```

# Lista Encadeada

```
class ListaEncad
{
    public:
        ListaEncad();
        ~ListaEncad();
        bool busca(int val);
        int get(int k);
        void set(int k, int val);
        void insereInicio(int val);
        void insereFinal(int val);
        void removeInicio();
        void removeFinal();

    private:
        No *primeiro;    /// primeiro No da lista
        No *ultimo;      /// ultimo No da lista
        int n;           /// numero de nos na lista
};
```

# Exercícios

0. **Faça uma leitura** do código fonte deste TAD com **atenção**, o qual se encontra nos arquivos `ListaEncad.h` e `ListaEncad.cpp`.
1. Implemente as seguintes operações:
  - ▶ Imprimir a lista;
  - ▶ Retornar o número de nós de uma lista;
  - ▶ Verificar se existe um elemento na lista com valor maior do que um valor fornecido. Retornar a posição do primeiro elemento que seja maior que o dado de entrada ou -1 caso contrário. ;
  - ▶ “Limpar” a lista, isto é, remover todos os elementos;
  - ▶ Calcular e retornar a média dos valores armazenados;
  - ▶ Concatenar (por cópia) a lista l2 ao final da lista interna;
  - ▶ Partir a lista interna em duas partes, de maneira que o nó de índice  $x$  se torne o primeiro nó da nova lista, que deve ser retornada pela função..

# Exercícios

1. (cont.) Utilize os seguintes protótipos:

```
void ListaEncad::imprime();  
int  ListaEncad::numNos();  
int  ListaEncad::buscaMaior(int val);  
void ListaEncad::limpar();  
float ListaEncad::calculaMedia();  
void ListaEncad::concatena(ListaEncad *l2);  
ListaEncad* ListaEncad::partir(int x);
```

## Exercícios

2. Criar um TAD Lista Encadeada de forma que todos os nós da lista fiquem sempre em **ordem crescente**. Dica: desenvolver uma única operação de inserção na lista de tal forma que o valor do novo nó mantenha a lista em ordem crescente.

```
void ListaEncad::insereOrdenado(int val);
```

Em seguida, modifique o programa principal e utilize a operação para inserir os nós de forma ordenada e, ao final do programa, imprima a lista.