

Laboratório de Programação II

Listas Duplamente Encadeadas

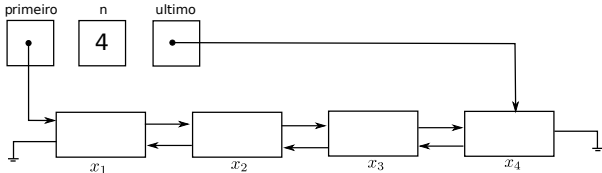
Universidade Federal de Juiz de Fora
Departamento de Ciência da Computação

Aula de Hoje

- ▶ Listas
 - ▶ Contíguas
 - ▶ Encadeadas
 - ▶ Simplesmente Encadeadas
 - ▶ **Duplamente Encadeadas**

Lista Duplamente Encadeada

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



Nó da Lista Duplamente Encadeada

```
class NoDuplo
{
public:
    NoDuplo();
    ~NoDuplo();
    void setAnt(NoDuplo *p);
    void setProx(NoDuplo *p);
    void setInfo(float val);
    NoDuplo* getAnt();
    NoDuplo* getProx();
    float getInfo();
private:
    NoDuplo *ant;    // ponteiro para anterior
    float info;      // informacao do no
    NoDuplo *prox;   // ponteiro para proximo
};
```

Lista Duplamente Encadeada

```
class ListaDupla
{
    public:
        ListaDupla();
        ~ListaDupla();
        bool busca(float val);
        void insereInicio(float val);
        void eliminaInicio();
        void insereFinal(float val);
        void eliminaFinal();
    private:
        NoDuplo *primeiro;
        NoDuplo *ultimo;
        int n;
};
```

Exercícios

0. **Faça uma leitura** do código fonte deste TAD com **atenção**, o qual se encontra nos arquivos `ListaDupla.h` e `ListaDupla.cpp`.
1. Implemente as seguintes operações:
 - ▶ Inserir no final;
 - ▶ Eliminar do final;
 - ▶ Imprimir a partir do primeiro;
 - ▶ Imprimir a partir do último (reverso);
 - ▶ Concatenar duas listas em uma única lista (retornar a lista resultante);
 - ▶ Partir uma lista em duas a partir de um nó cujo valor x é dado.

Exercícios

1. Utilize os seguintes protótipos:

```
void ListaDupla::insereFinal(float val);  
void ListaDupla::eliminaFinal();  
void ListaDupla::imprime();  
void ListaDupla::imprimeReverso();  
ListaDupla* ListaDupla::concatena(ListaDupla *l2);  
ListaDupla* ListaDupla::partir(float x);
```

Exercícios

2. Crie uma operação para remover todas as ocorrências de um dado valor em uma lista duplamente encadeada utilizando o protótipo abaixo:

```
void ListaDupla::removeOcorrencias(float val);
```