#### Trabalho Prático

# **Lista Duplamente Encadeada**

**Objetivo do Trabalho:** trabalho opcional visando recuperar conceito do primeiro bimestre. O aluno que tiver interesse, deverá implementar ***individualmente*** o este trabalho e entregar para avaliação do professor. Caso o professor considere o produto satisfatório, poderá acrescentar **até** um ponto de bônus na média final do primeiro bimestre. Prazo de entrega e trabalhos idênticos com evidências de cópia serão aspectos relevantes da avaliação

**Prazo de entrega:** **23/05/2019**

Em estruturas lineares implementadas com encadeamento e alocação dinâmica de memória, quando é necessário caminhar nos dois sentidos é interessante termos referências (apontadores) para o **sucessor** e para o **antecessor** de cada item. Esta solução é chamada de ***lista duplamente encadeada***, e está ilustrada abaixo.

***Celula***

antecessor|elemento|sucessor

inicio tamanho fim

***Lista***

O trabalho consiste no seguinte:

**Parte 1 – Projeto da solução**: Projetar e implementar em C uma solução para o modelo acima. Conforme sugerido na figura, a solução consiste em construir um ***TAD ListaDupla***, que mantém duas estruturas de dados: uma Lista, com referências para *início* e *fim* da lista e o *tamanho* dinâmico desta; e outra estrutura *Celula*, com referências para o *sucessor* e *antecessor* e para o *elemento de dados* mantido na lista.

Para o ***TAD ListaDupla***, projetar e implementar em C as operações:

* *void criaLista (Lista \*lista);*
* *int vazia (Lista \*lista);*
* *void insereFinal (Elemento x, Lista \*lst);*
* *Ponteiro buscaLista(Elemento x, Lista \*lst);*
* *int excluiLista(Elemento x, Lista \*lista);*

**Nota**: o termo *projetar* presente na especificação acima significa desenhar, de forma semelhante à apresentada na apostila “Listas Lineares – implementação por Encadeamento”, onde é demonstrado o projeto (desenhos e descrição de passos das estruturas de dados e dos algoritmos de operações do *TAD Lista simplesmente encadeada*. Por isso, sugere-se fortemente seguir a mesma abordagem da apostila e também reaproveitar o que já foi demonstrado na prova de conceito de Lista Encadeada (reutilize os arquivos ***TAD-ListaEncad.c*** e ***ListaEncad-ProvaConc.c***).

**Parte 2 – Prova de conceito (PoC) da solução**: Para testar o ***TAD ListaDupla***, considerar o problema de manter uma lista de números inteiros que deve ser mostrada nos dois sentidos. Para resolver este problema, usar obrigatoriamente o ***TAD ListaDupla*** adaptado para armazenar *números inteiros* e implementar uma função ***main*** com um *menu* oferecendo as seguintes opções:

1. Inserir um número no inicio da lista;
2. Excluir um número informado;
3. Menu consultas.

No menu consultas, implementar as seguintes funcionalidades:

1. Consultar por número;
2. Listar do início para fim;
3. Listar do fim para início.

Observação: para a funcionalidade Listar do início para fim, obrigatoriamente caminhar na lista usando a referência (ponteiro) “*sucessor*” e para a funcionalidade Listar do fim para início, obrigatoriamente caminhar na lista usando a referência (ponteiro) “*antecessor*”.

Deverá ser apresentado ao professor:

1. Os arquivos **.c** implementando o ***TAD ListaDupla*** e a ***prova de conceito***;
2. A aplicação executando todas as funcionalidades de forma consistente.

Dúvidas poderão ser dirimidas com o professor. Pontos omissos na especificação e decisões de implementação deverão ser resolvidos pelo próprio aluno, devidamente justificados na apresentação do projeto.