

## Lista encadeada

Você pode utilizar qualquer ambiente de programação para desenvolver sua atividade. Ao final, **copie e cole o seu código-fonte com a resposta aqui mesmo neste documento**, dentro dos espaços indicados para isso e **preservando a indentação do código**. **Depois que terminar sua avaliação, não se esqueça de entregar sua atividade!** Fique atento ao relógio, pois as atividades entregues com atraso não serão aceitas.

Para resolver esta atividade, [clique aqui para baixar](#) o projeto da aula 7 de laboratório de programação 2, que contém a implementação do TAD ListaEncad para lista encadeada de números inteiros. Na sua solução para a questão abaixo, [você pode utilizar/chamar](#) qualquer uma das operações que estejam disponíveis no projeto (exatamente do jeito que ele se encontra no site da disciplina). Quaisquer outras operações necessárias para a sua resposta devem ser implementadas e incluídas neste documento.

Implementar a operação `void ListaEncad::aumentar_n(int n1)`; para aumentar o valor  $n$  (número de nós) de uma **lista simplesmente encadeada com descritor** (LSED) para o novo valor  $n1$ . O parâmetro  $n1$  tem que ser maior que  $n$ , emitir a mensagem "Valor invalido" caso  $n1$  não seja válido.

Não basta apenas atribuir à  $n$  o seu novo valor  $n1$ , um número adequado de nós deve ser adicionado no **início** da lista para que ela fique coerente com o novo valor para  $n$ . O valor dos novos nós deve ser uma média do valor do primeiro com o valor do último nó original, se lista não vazia. Ou deve ser 0 (zero) se a lista estiver vazia ( $n = 0$ ).

**Exemplo 1:** Considere a lista simplesmente encadeada com descritor  $L$  com os valores  $L = [8 \ 32 \ 17]$ :

- O comando `L.aumentar_n(6)` torna  $L = [12 \ 12 \ 12 \ 8 \ 32 \ 17]$ .
- O comando `L.aumentar_n(2)` obtém a mensagem: "Valor invalido".

**Exemplo 2:** Considere a lista simplesmente encadeada com descritor  $L = []$  (vazia):

- O comando `L.aumentar_n(6)` torna  $L = [0 \ 0 \ 0 \ 0 \ 0 \ 0]$ .

```
void ListaEncad::aumentar_n(int n1)
{
    if(n1 > n)
        cout << "Valor inválido" << endl;
    {
        int *vet = new int[n];
        No* p = primeiro;
        for(int i = 0; i < n-1; i++)
```

```
        {
            if(p->getInfo() < p->getProx()->getInfo())
                vet[i] = 1;
            else
                vet[i] = 0;
            p = p->getProx();
        }
    vet[n-1] = 1;
    return vet;
}
else
    return NULL;
}
```