

## Lista duplamente 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 que contém a implementação do TAD `ListaDupla` de **números inteiros** (usando uma **lista duplamente encadeada - LDE**). Na sua solução para a questão abaixo, **você pode utilizar/chamar** qualquer uma das operações que estejam disponíveis no projeto, exceto aquelas explicitamente desautorizadas no enunciado. Outras operações que você eventualmente deseje utilizar devem ser copiadas para sua resposta neste documento.

Implementar a operação `ListaDupla* ListaDupla::parte(int val)` para, dado um inteiro `val`, partir a lista em duas. A primeira partição deve permanecer na lista interna, incluindo o nó com valor `val`. A segunda partição deve ser incluída em uma nova lista, porém em ordem inversa. Esta nova lista deve ser retornada ao final da operação. Considere que a primeira partição se encerrará na primeira ocorrência de `val`, caso haja mais de uma. Se a lista estiver vazia ou se `val` não for encontrado, deve-se retornar `NULL`. A lista deve ser percorrida uma única vez, e as funções de inserção e remoção do TAD não devem ser utilizadas.

### Exemplos:

```
L1 (Antes): 9 19 48 39 24 27 38 32 4 12 | val = 27
L1 (Depois): 9 48 39 24 27
L2 (Depois): 12 4 32 38
```

```
L1 (Antes): 9 19 48 39 24 27 38 32 4 12 | val = 12
L1 (Depois): 9 19 48 39 24 27 38 32 4 12
L2 (Depois): Vazia
```

```
L1 (Antes): 9 19 48 39 24 27 38 32 4 12 | val = 9
L1 (Depois): 9
L2 (Depois): 12 4 32 38 27 24 39 48 19
```

```
L1 (Antes): 9 19 48 39 24 27 38 32 4 12 | val = 30
L1 (Depois): 9 19 48 39 24 27 38 32 4 12
L2 (Depois): NULL
```

Inserir seu código aqui:

```
ListaDupla* ListaDupla::parte(int val)
```

```

{
    ListaDupla *nova = NULL;

    NoDuplo *p = primeiro;
    while(p != NULL && p->getInfo() != val)
        p = p->getProx();

    if(p != NULL)
    {
        nova = new ListaDupla();
        nova->ultimo = p->getProx();
        if(nova->ultimo != NULL)
        {
            nova->ultimo->setAnt(NULL);
            nova->primeiro = ultimo;
        }
        else
            nova->primeiro = NULL;

        ultimo = p;
        p->setProx(NULL);

        for(NoDuplo *p = nova->primeiro; p != NULL; p = p->getProx())
        {
            NoDuplo *aux = p->getAnt();
            p->setAnt(p->getProx());
            p->setProx(aux);
        }
    }

    return nova;
}

```