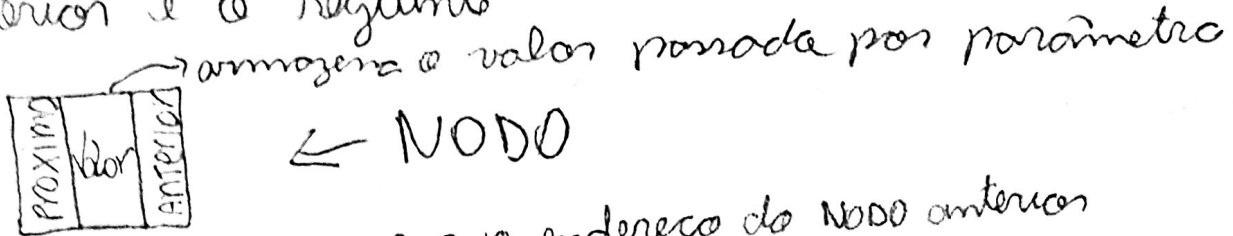


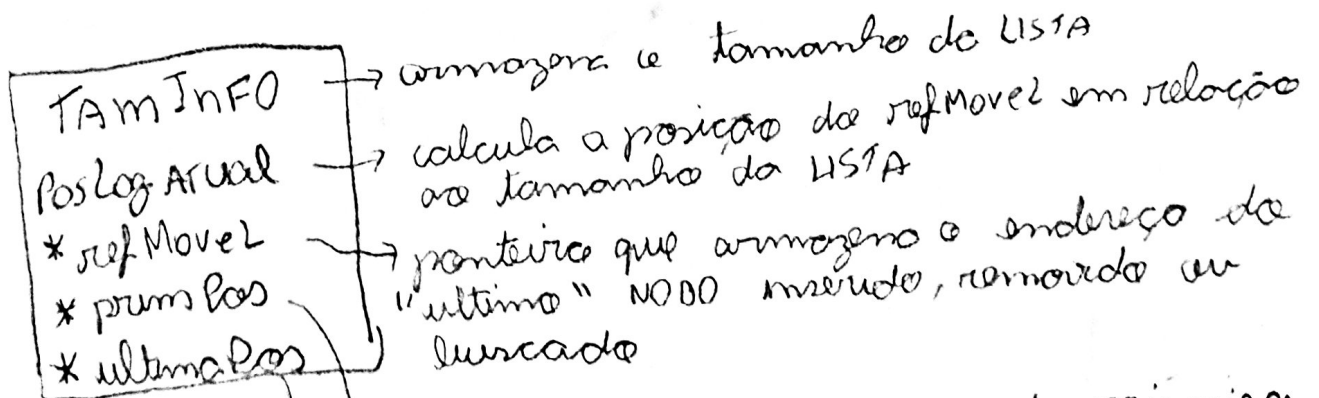
# IDEIA DE IMPLEMENTAÇÃO

- Criação de 2 estruturas. NODO e DESCRITOR, na qual NODO → armazena o valor passado além do elemento anterior e o seguinte



→ armazena o endereço do NODO anterior  
→ armazena o endereço do Próximo NODO

- Descritor → armazena informações da LISTA. Como o tamanho da lista, o endereço da primeira e última posição. Além de variáveis de controle. Uma será um ponteiro auxiliar no qual aponta para onde houve alguma função na lista, como uma remoção, busca ou inserção de um nó. A outra variável localiza a posição do descritor pelo auxiliar em relação ao tamanho da lista.



DESCRITOR

→ armazena o endereço da primeira NODO da LISTA

→ armazena o endereço do último NODO da lista

a) Tamanho da lista

Função implementada

```
int tamanhoDaLista (Descricao * lista)
```

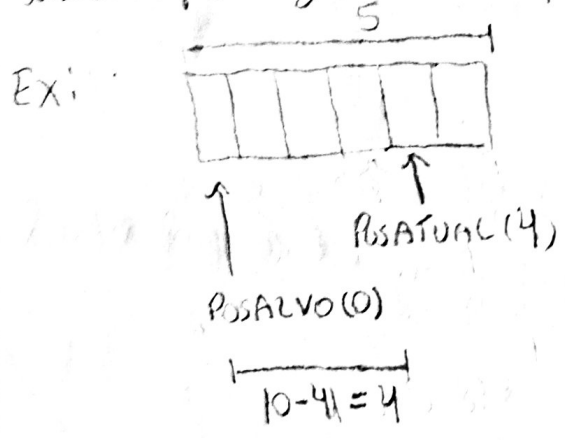
```
return lista -> Tam Info.
```

Como o descritor  $ht$  possui uma variável que armazena o tamanho da lista, basta retornar ele.

# b) Busca na Posição Função implementada.

INT buscaNaPosicao (Descretores \* lista, int posALVO) {

- Primeiro é verificado se a posição passada como parâmetro é válida. Se não for, apenas retornar uma mensagem de erro. Se for a função irá fazer a busca
- Como a lista é circular (último elemento aponta para o primeiro da lista), antes de buscar, o algoritmo irá fazer o cálculo de qual caminho é mais rápido  
 Se  $|posALVO - lista \rightarrow posLogATUAL| < |(lista \rightarrow tamInfo - 1 - lista \rightarrow posLogATUAL) + posALVO|$



$$15-4=11$$

$$15-4+0=11$$

Como  $4 < 11 = \text{False}$

Então



Senão



## C) Remove da Posição

Função implementada

int removeDaPosicao (descricao \* lista, int posAtual)

- Primeiro é verificado se a posição é válida
- Se a posição for válida, é usada a função de busca na posição para "levar o descritor" até a posição de remoção.
- São criados dois ponteiros auxiliares. Um armazena o endereço no qual estava o NODO anterior ao elemento a ser removido. O outro armazena o endereço do NODO próximo ao NODO a ser removido.
- Free "limpa" da memória o nó que agora foi removido
- O algoritmo faz a troca dos variáveis Proximo e Anterior para que os auxiliares apontem. É a ref. movel

Ex:

