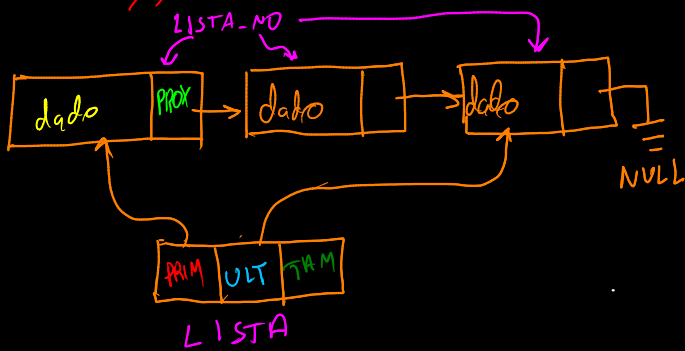


$V = \begin{bmatrix} 3 & 0 & 1 & 2 & 4 \\ \uparrow \text{ponteiros} & 0 & 1 & 2 & 3 & 4 \end{bmatrix}$

$\&V[1] = 104$

$V: 100$

`int v[5];`
 4 bytes / elemento



```

typedef struct LISTA_NO {
    int dado;
    struct LISTA_NO* prox;
} LISTA_NO;
  
```

```

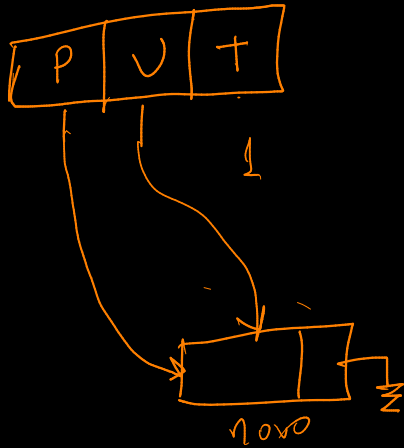
typedef struct LISTA {
    LISTA_NO* primeiro;
    LISTA_NO* ultimo;
    int tam;
} LISTA;
  
```

```

LISTA* LISTA_CRIAR();
VOID LISTA_INSERIR(LISTA *L, int dado);
LISTA_NO* LISTA_BUSCAR(LISTA *L, int dado);
bool LISTA_VAZIA(LISTA *L);
int LISTA_TAMANHO(LISTA *L);
VOID LISTA_IMPRIMIR(LISTA *L);
VOID LISTA_DESTRUIR(LISTA *L);
bool LISTA_REMOVER(LISTA *L, int dado);
  
```

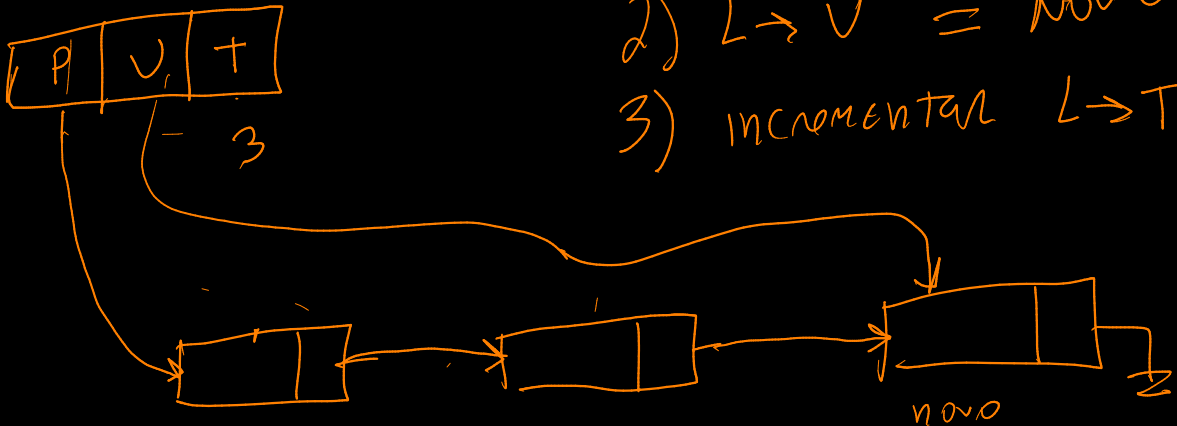
INSERÇÃO

Caso 1: LISTA VAZIA



- 1) $L \rightarrow \text{PRIMEIRO} = \text{novo}$
- 2) $L \rightarrow \text{ULTIMO} = \text{novo}$
- 3) incremental $L \rightarrow T$

Caso 2 LISTA \hat{n} VAZIA



- 1) $L \rightarrow V \rightarrow \text{prox} = \text{novo}$
- 2) $L \rightarrow V = \text{novo}$
- 3) incremental $L \rightarrow T$