

Listas sequenciais (vetores)

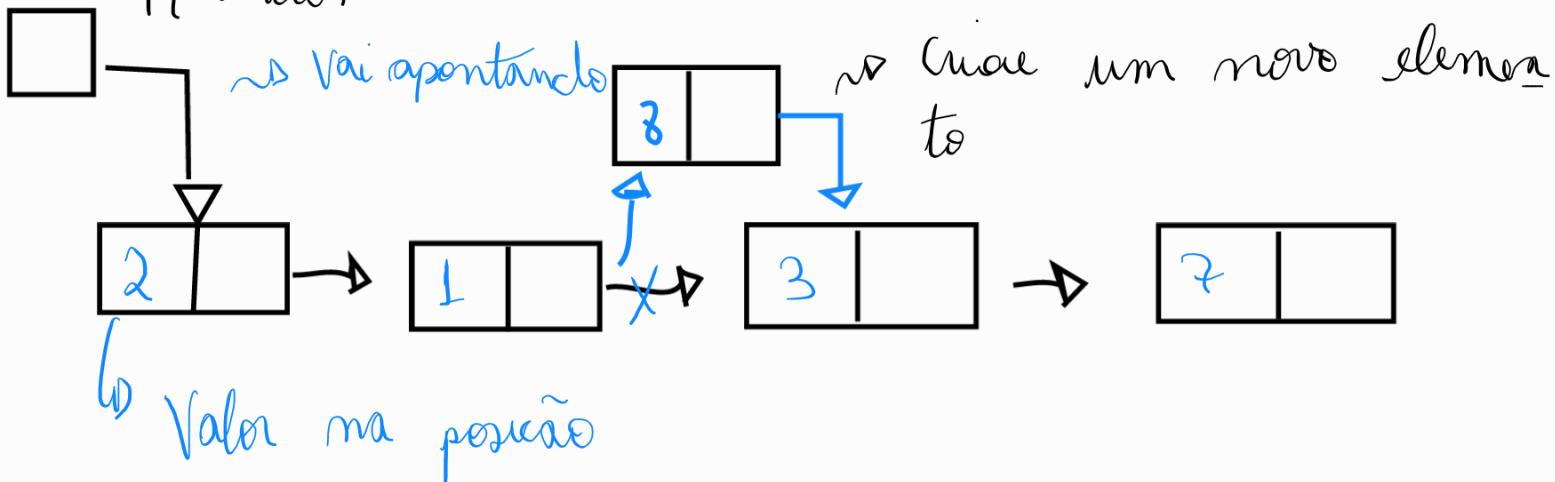
- Vetores
- O problema tá em inserir, remover e alterar tem haver com deslocamento ($i+1$) ($i-1$)

Listas ligadas

- Um ponteiro que aponta pra primeira e endereço (não entendi muito bem mas acho que o ponteiro vai apontando)
- Listas ligadas são vetores que podem crescer ou diminuir conforme as nossas necessidades.
- Eliminar ou inserir sem alterações em posições

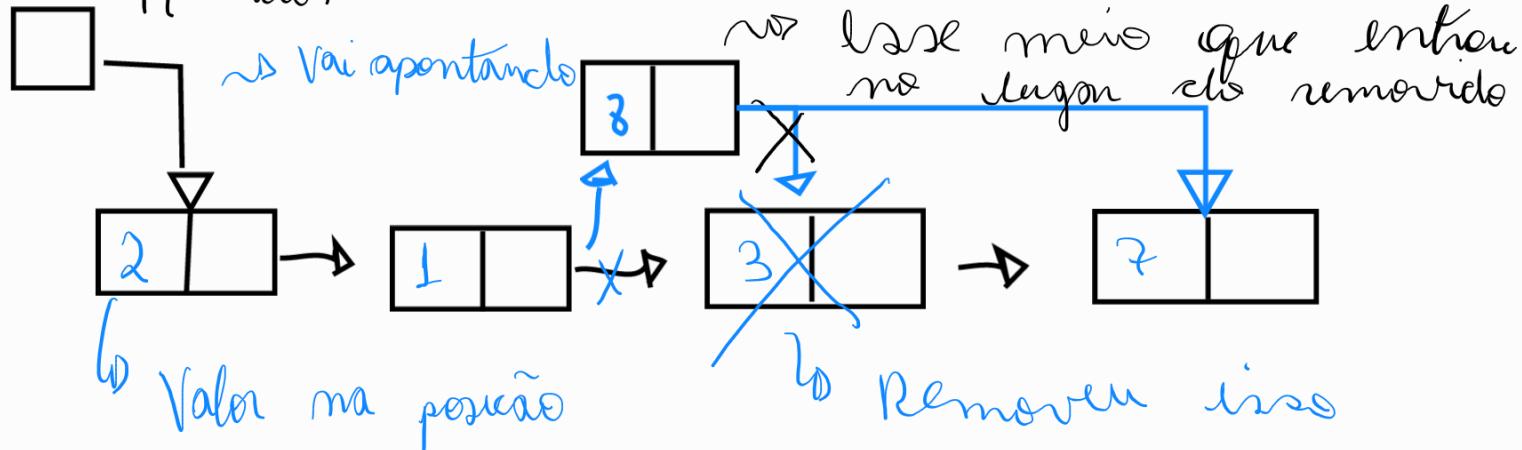
Como Funciona? (Ao adicionar)

P(_{início})



Como Funciona? (ao remover)

P(_{início})



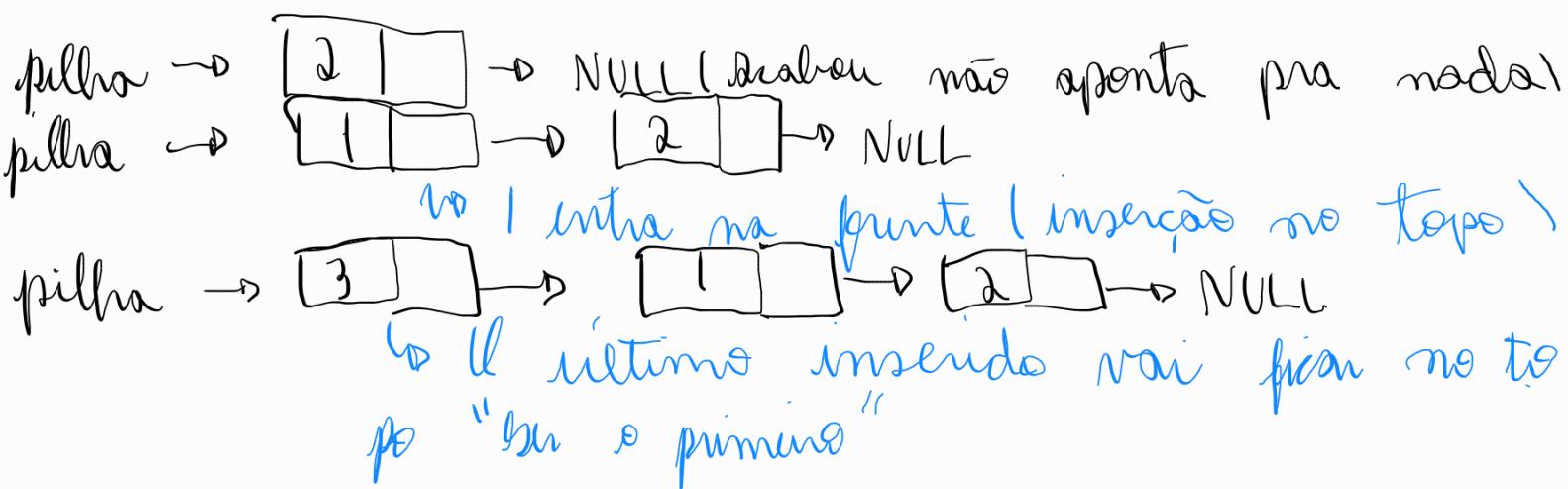
1º tím. informação de estrutura de dados

informações da Segunda, a terceira
do terceiro, etc

Pilhas: (LIFO)

- Tem. haver com pilha de livros
- As inserções e remoções são realizadas no topo (topo)

Exemplo: Se inserir 2, 1, 3 (nessa ordem)



- Fica 3, 1, 2
- Se fosse para remover o primeiro a ser removido seria o 3, o segundo 1, o terceiro

Código (partes importantes)

- list < int > pilha; // ponteiro para a lista
↳ é uma lista de inteiros
- Tem que usar essa biblioteca: #include < list >
- Inserir na pilha:
pilha.push_front // inserir os elementos na frente da fila [front !!]
- Remover os dados da pilha e ir mostrando:
 $x = *pilha \cdot begin();$ // Aponta para o primeiro da lista (begin = inicio)
cout << x;
- pilha.pop_front(); // Isso remove o primeiro da

lista

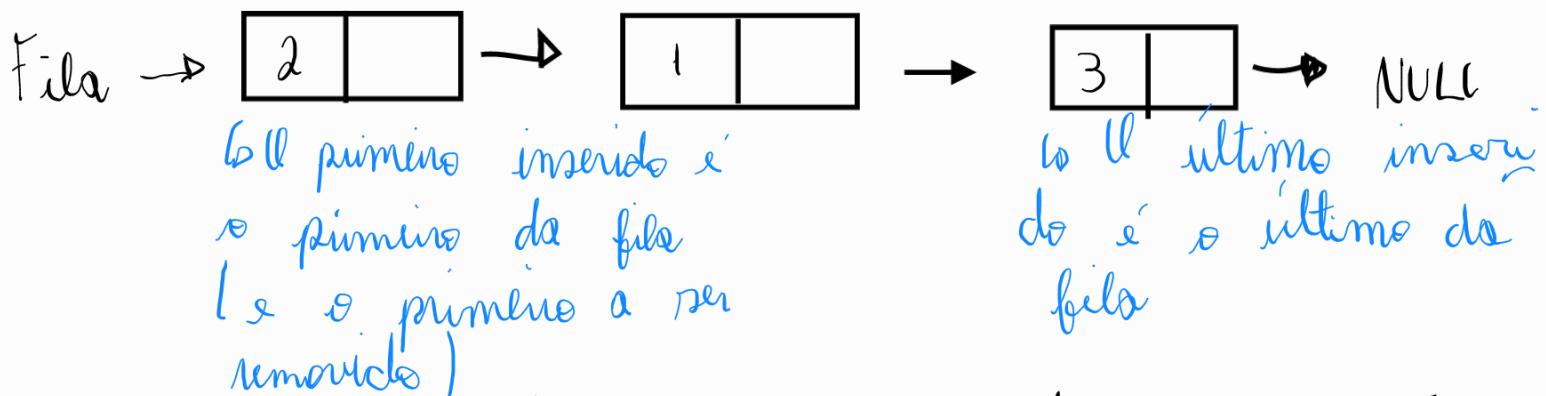
- Quando não sabe o tamanho da lista:
- ```

while (!pilha.empty()) { // enquanto a pilha não
 X = *pilha.begin(); // ficar vazia aponta para
 cout << X << endl; // primeira, imprime ela e
 pilha.pop_front(); // remove
}

```

## Filas:

- Inserção no final da fila e Remoção no inicio
  - O primeiro da fila é o primeiro que sai
- Exemplo:** Se inserir 2, 1, 3 nessa ordem



- Mostra os elementos na mesma ordem que entrou
- Enquanto as pilhas mostram na ordem inversa

## Código (partes importantes):

```

list<int> fila; // declara a fila
fila.push_back(); // Insere os elementos no final da fila (back !!!)
fila.pop_front(); // remove o primeiro
cin >> x; // de um certo valor
fila.push_back(x); // adiciona esse elemento na fila

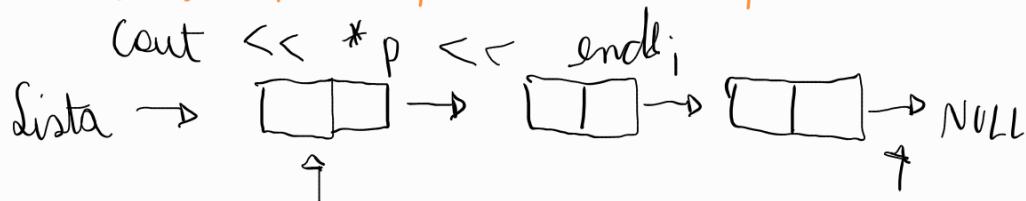
```

## Stacks:

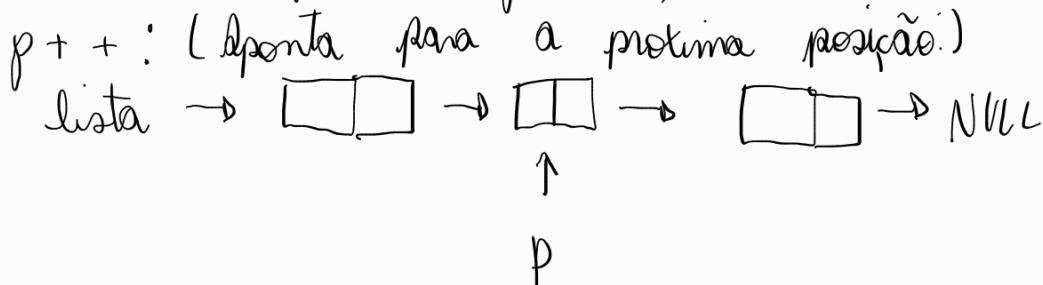
- Pode ser considerados ponteiros para os elementos da lista

Código:

```
list < int > :: iterator p; // declara o iterator
for (p = lista.begin(); p != lista.end(); p++) // lista.begin = 1º nó
 lista.end = aponta pro elemento depois do último
```



p(lista.begin)                            lista.end



Vai varrendo o vetor até chegar no lista.end

