



ESTRUTURA DE DADOS

PILHAS DINÂMICAS E EXERCÍCIOS COM LISTAS ENCADEADAS

Prof. Dr. Daniel Caetano

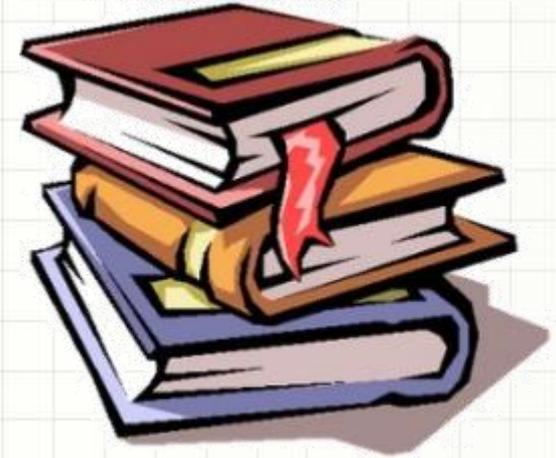
2014 - 2

Objetivos

- Compreender como usar uma lista ligada como uma pilha
- Usar pilhas dinâmicas para aplicações
- Treinar construção de operações com listas ligadas
- **Atividade Estruturada!**



Material de Estudo



Material

Acesso ao Material

Apresentação

<http://www.caetano.eng.br/>
(Aula 13)

Material Didático

Estruturas de Dados (Parte 2) – Páginas 132 a 136

Online

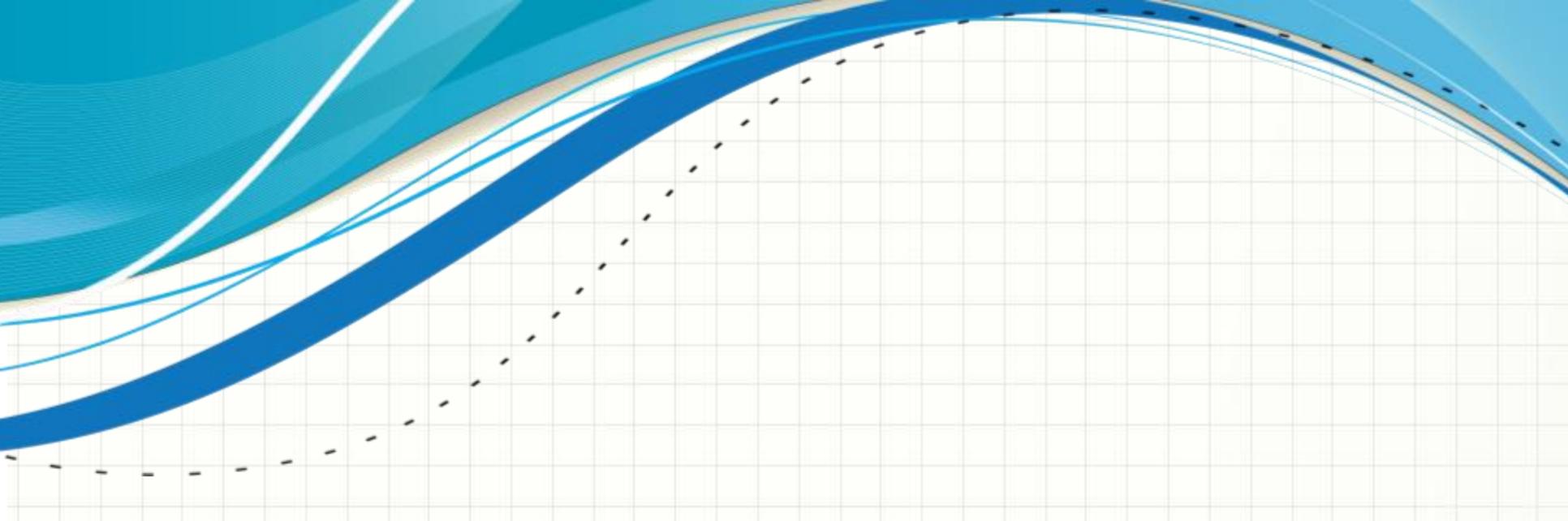
C Completo e Total – Páginas 540 a 550

The image features a decorative header with several overlapping, curved lines in shades of blue and white. Below this, the background is a light gray grid. The text "RECORDANDO..." is positioned in the lower right area of the grid.

RECORDANDO...

Recordando...

- Vimos Listas Encadeadas
- Uso:
 - Inserção no Início
 - Remoção do Início
- E se chamarmos o início de “topo”?
- Virou uma pilha?



PILHA ENCADEADA OU PILHA DINÂMICA

Pilhas Encadeadas

- Sempre que usarmos uma lista
 - Inserirmos / Removermos do início... Ou
 - Inserirmos / Removermos do fim...
- Pilha Dinâmica ou Encadeada
- Exemplo

Pilha: Inicializando

pilha

NULL

Quando criamos uma pilha, ela está vazia

```
no *pilha = NULL;
```

Pilha: Inserção

pilha

NULL

Como criamos um elemento?

```
no *tmp;  
tmp = new no;  
tmp->valor = 1;  
tmp->prox = NULL
```

tmp

1

NULL

Pilha: Inserção

ilha

NULL

Como inseri-lo na
ilha?

```
tmp->prox = ilha;  
ilha = tmp;
```

tmp

1

NULL

Pilha: Inserção

ilha

NULL

Como inseri-lo na
ilha?

```
tmp->prox = ilha;  
ilha = tmp;
```

tmp

1

NULL

Pilha: Inserção

ilha

NULL

Como inseri-lo na
ilha?

```
tmp->prox = ilha;  
ilha = tmp;
```

tmp

1

NULL

Pilha: Inserção



Como inseri-lo na
ilha?

```
tmp->prox = ilha;  
ilha = tmp;
```



Pilha: Inserção

pilha



Inserindo outro nó

```
no *tmp;  
tmp = new no;  
tmp->valor = 2;  
tmp->prox = pilha;  
pilha = tmp;
```



Pilha: Inserção

pilha



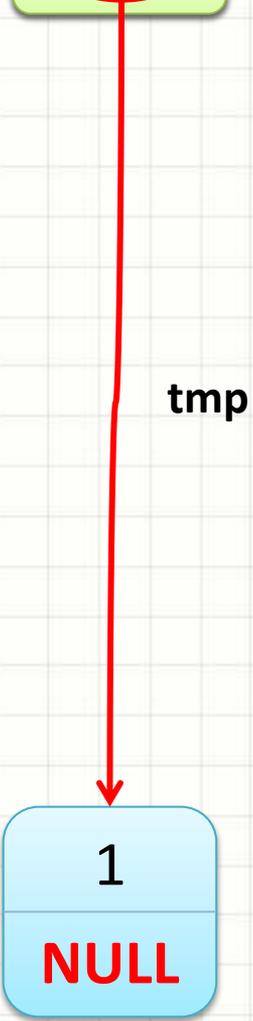
Inserindo outro nó

```
no *tmp;  
tmp = new no;  
tmp->valor = 2;  
tmp->prox = pilha;  
pilha = tmp;
```

Pilha: Inserção



Inserindo outro nó



```
no *tmp;  
tmp = new no;  
tmp->valor = 2;  
tmp->prox = pilha;  
pilha = tmp;
```

Pilha: Inserção



tmp



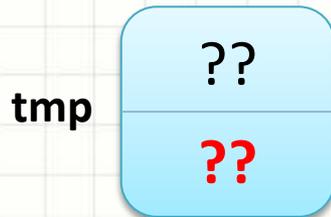
Inserindo outro nó

```
no *tmp;  
tmp = new no;  
tmp->valor = 2;  
tmp->prox = pilha;  
pilha = tmp;
```

Pilha: Inserção



Inserindo outro nó

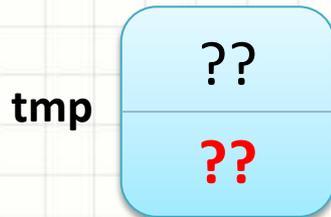


```
no *tmp;  
tmp = new no;  
tmp->valor = 2;  
tmp->prox = pilha;  
pilha = tmp;
```

Pilha: Inserção



Inserindo outro nó



```
no *tmp;  
tmp = new no;  
tmp->valor = 2;  
tmp->prox = pilha;  
pilha = tmp;
```

Pilha: Inserção

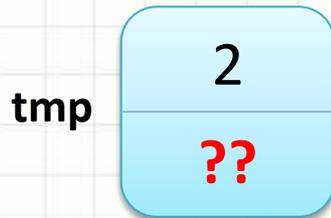


Inserindo outro nó



```
no *tmp;  
tmp = new no;  
tmp->valor = 2;  
tmp->prox = pilha;  
pilha = tmp;
```

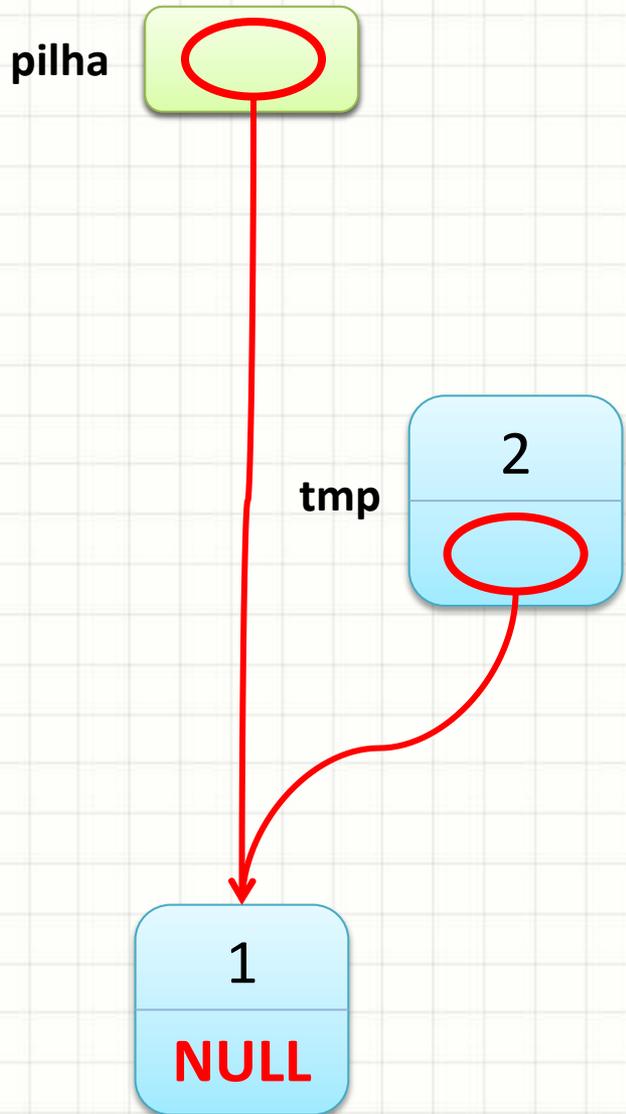
Pilha: Inserção



Inserindo outro nó

```
no *tmp;  
tmp = new no;  
tmp->valor = 2;  
tmp->prox = pilha;  
pilha = tmp;
```

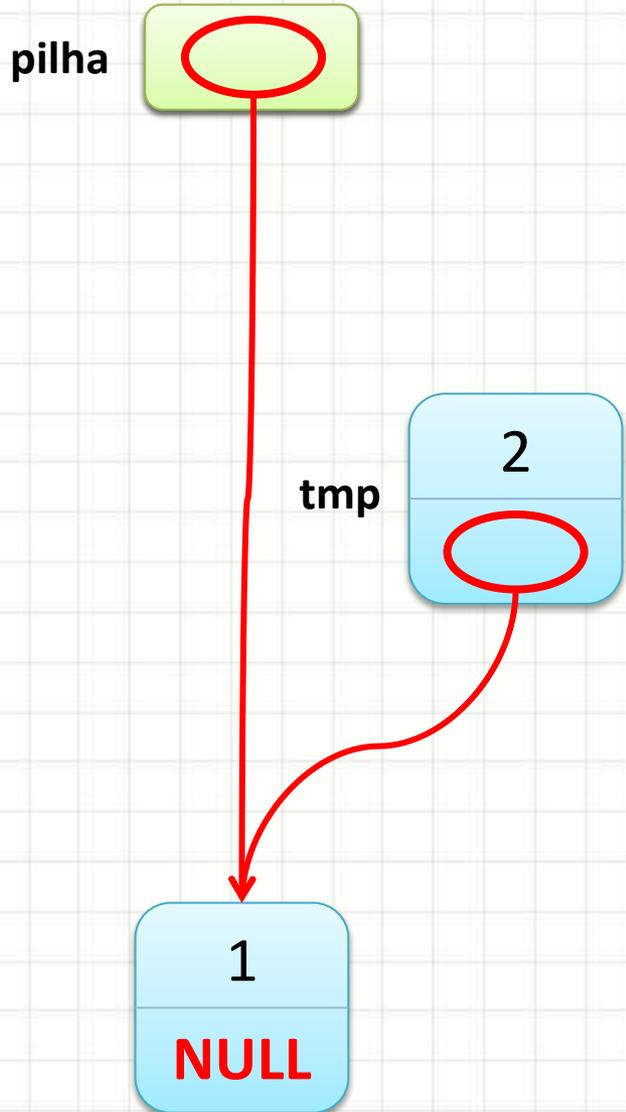
Pilha: Inserção



Inserindo outro nó

```
no *tmp;  
tmp = new no;  
tmp->valor = 2;  
tmp->prox = pilha;  
pilha = tmp;
```

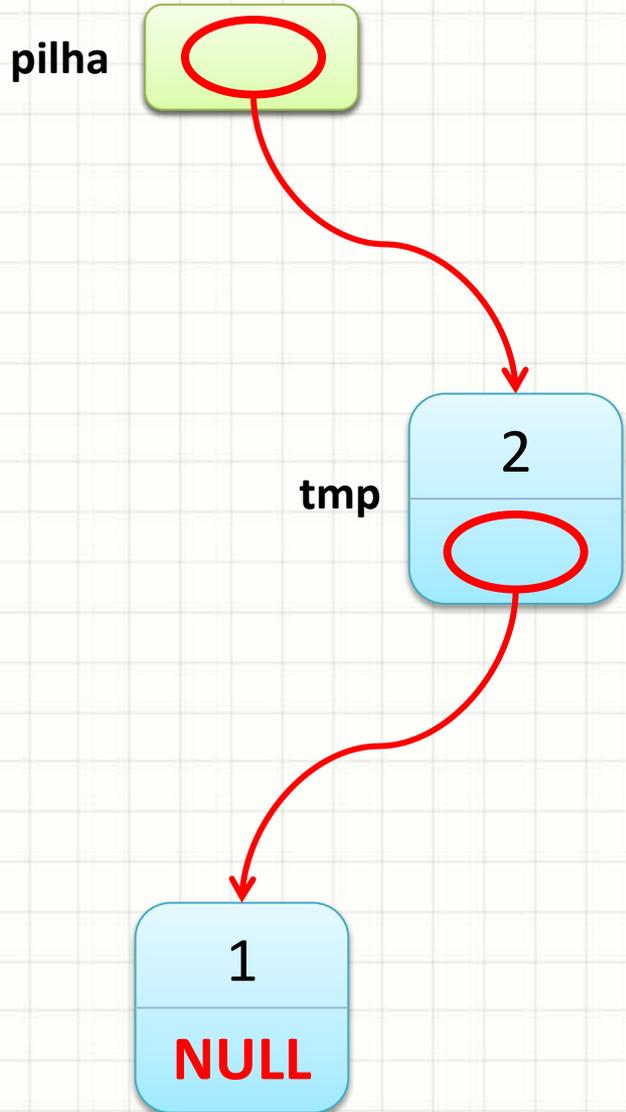
Pilha: Inserção



Inserindo outro nó

```
no *tmp;  
tmp = new no;  
tmp->valor = 2;  
tmp->prox = pilha;  
pilha = tmp;
```

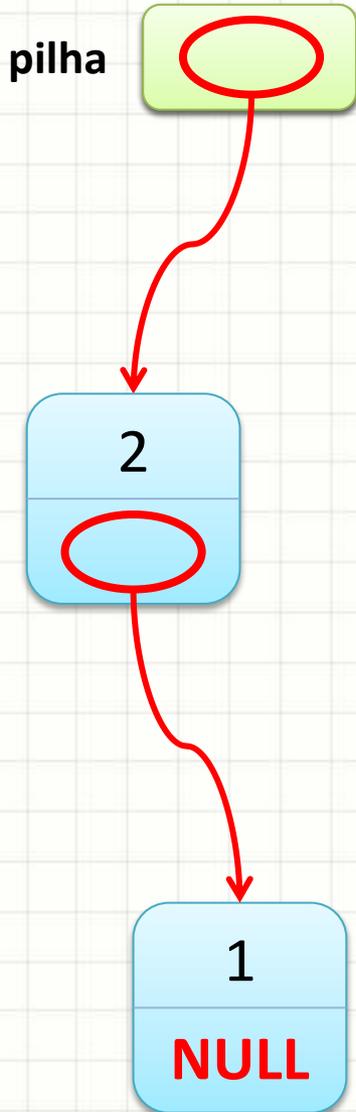
Pilha: Inserção



Inserindo outro nó

```
no *tmp;  
tmp = new no;  
tmp->valor = 2;  
tmp->prox = pilha;  
pilha = tmp;
```

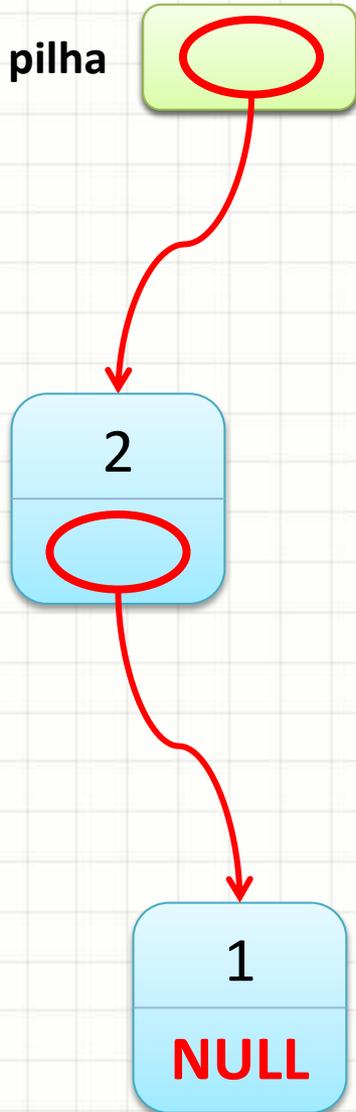
Pilha: Remoção



Removendo um nó

```
no *tmp;  
tmp = pilha;  
pilha = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

Pilha: Remoção



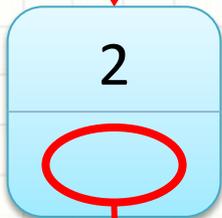
Removendo um nó

```
no *tmp;  
tmp = pilha;  
pilha = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

Pilha: Remoção



Removendo um nó

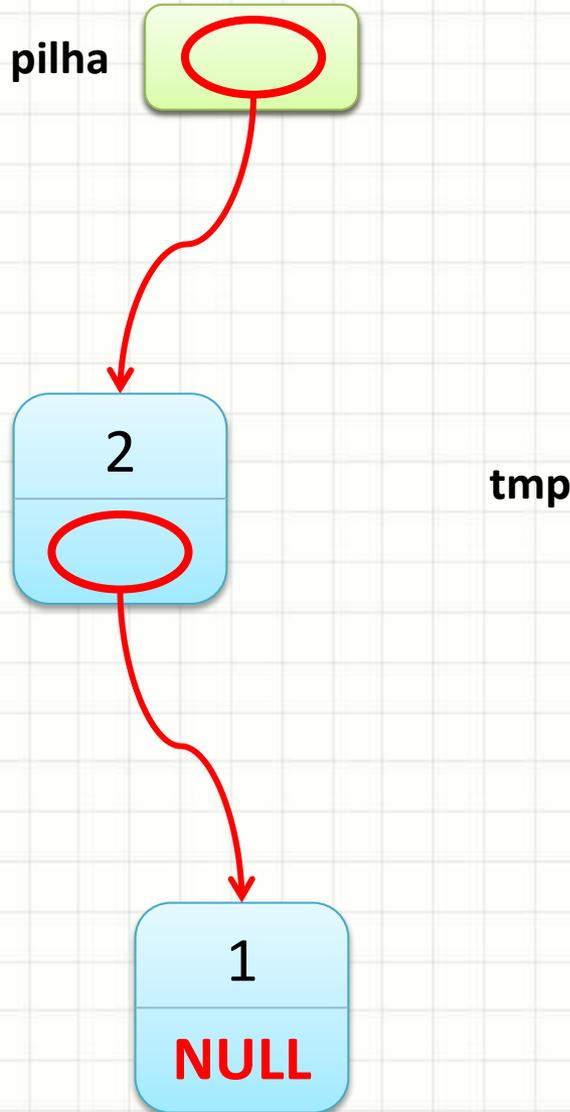


tmp



```
no *tmp;  
tmp = pilha;  
pilha = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

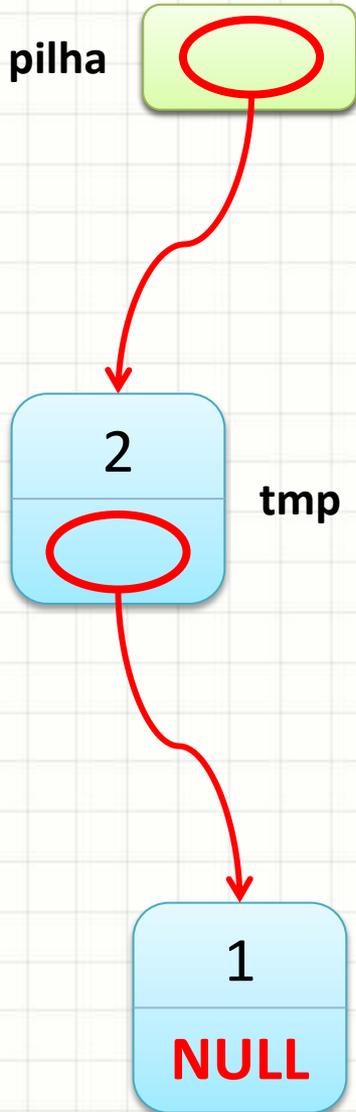
Pilha: Remoção



Removendo um nó

```
no *tmp;  
tmp = pilha;  
pilha = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

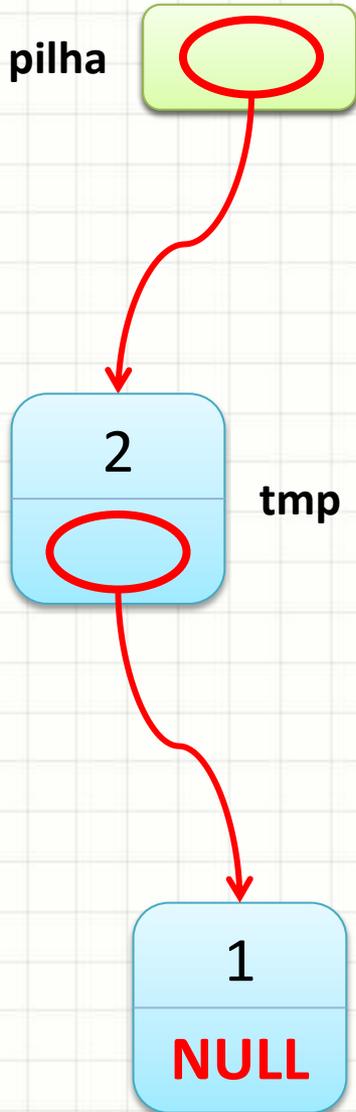
Pilha: Remoção



Removendo um nó

```
no *tmp;  
tmp = pilha;  
pilha = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

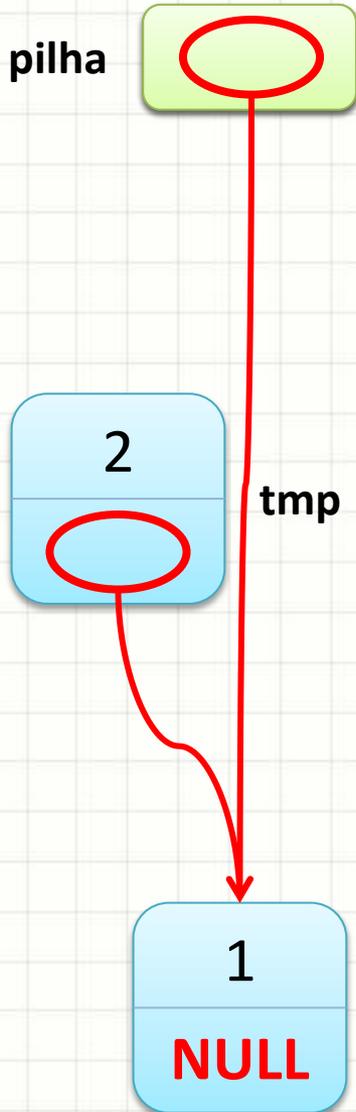
Pilha: Remoção



Removendo um nó

```
no *tmp;  
tmp = pilha;  
pilha = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

Pilha: Remoção



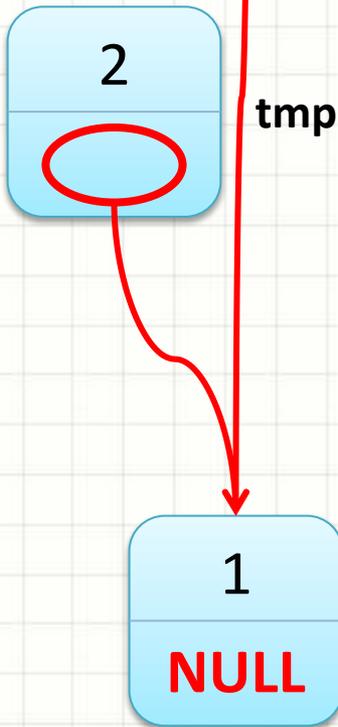
Removendo um nó

```
no *tmp;  
tmp = pilha;  
pilha = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

Pilha: Remoção



Removendo um nó

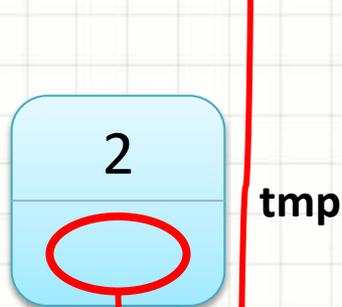


```
no *tmp;  
tmp = pilha;  
pilha = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

Pilha: Remoção



Removendo um nó

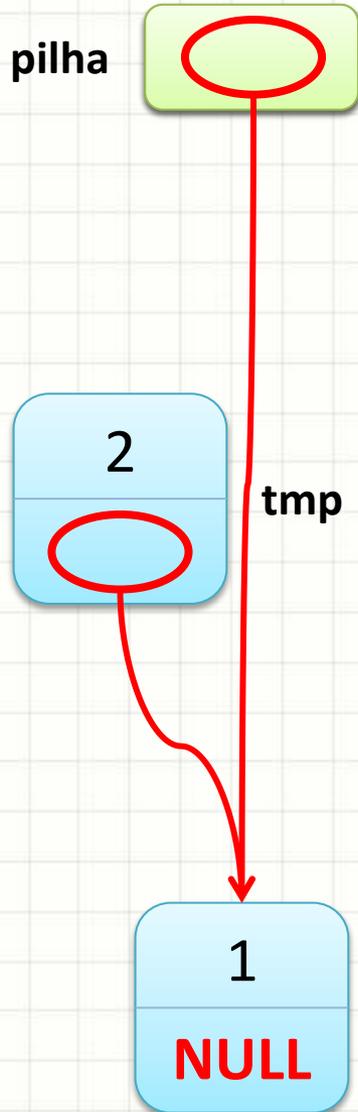


2

```
no *tmp;  
tmp = pilha;  
pilha = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```



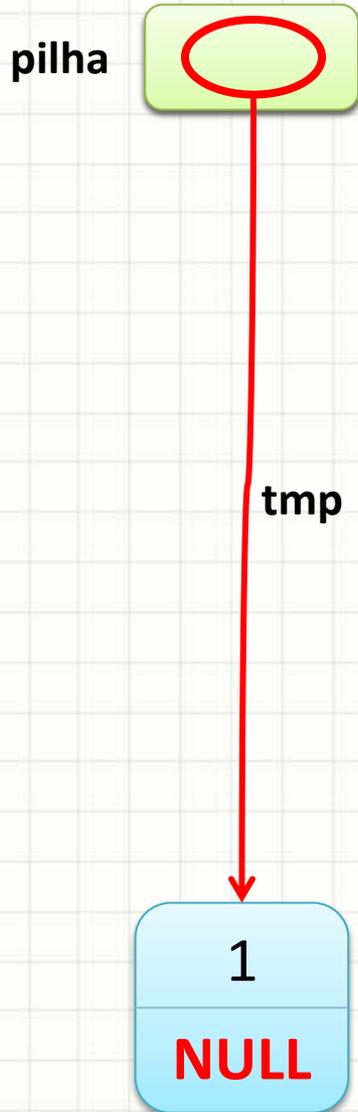
Pilha: Remoção



Removendo um nó

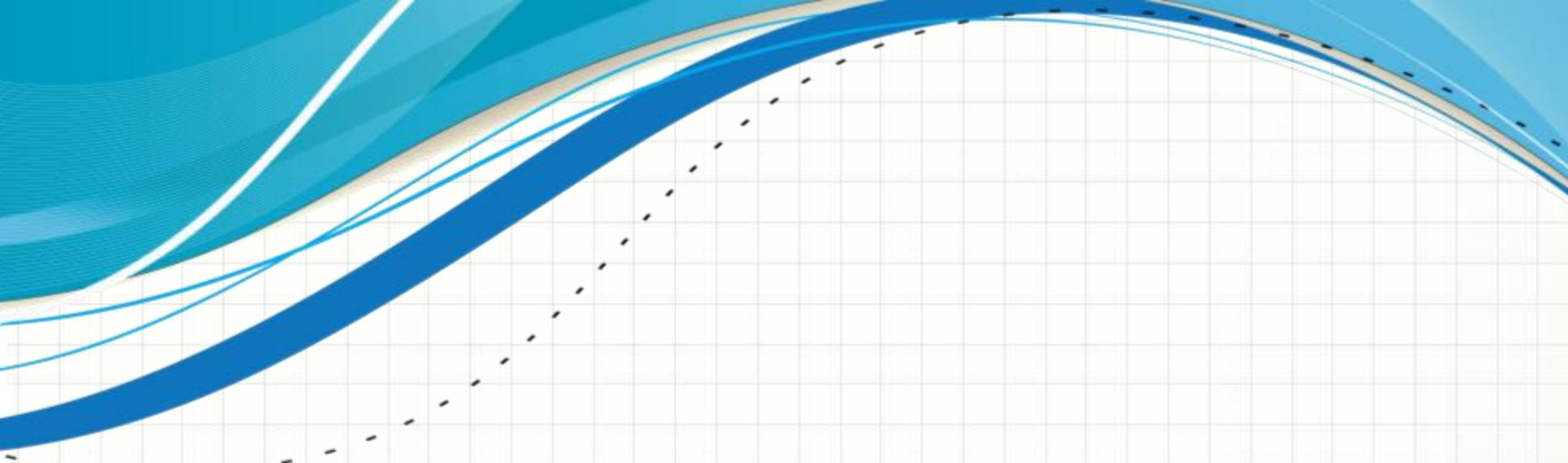
```
no *tmp;  
tmp = pilha;  
pilha = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```

Pilha: Remoção



Removendo um nó

```
no *tmp;  
tmp = pilha;  
pilha = tmp->prox;  
valor = tmp->valor;  
delete tmp;
```



EXERCÍCIO 1:
USANDO UMA PILHA
ENCADEADA

Exercício 1

- Converter um número para binário:
 - Dividir por 2 e anotar o resto (da dir para esq.)
 - Pegar parte inteira...
 - Dividir por 2 e anotar o resto...

Conversão $D \rightarrow B$

- Regra prática: converter 13 para binário

1b

- $13/2 = 6\dots$ Resto 1

Conversão $D \rightarrow B$

- Regra prática: converter 13 para binário

01b

- $13/2 = 6\dots$ Resto 1
- $6/2 = 3\dots$ Resto 0

Conversão $D \rightarrow B$

- Regra prática: converter 13 para binário

101b

- $13/2 = 6\dots$ Resto 1
- $6/2 = 3\dots$ Resto 0
- $3/2 = 1\dots$ Resto 1

Conversão $D \rightarrow B$

- Regra prática: converter 13 para binário

1101b

- $13/2 = 6\dots$ Resto 1
- $6/2 = 3\dots$ Resto 0
- $3/2 = 1\dots$ Resto 1
- $1/2 = 0\dots$ Resto 1

Conversão D→B

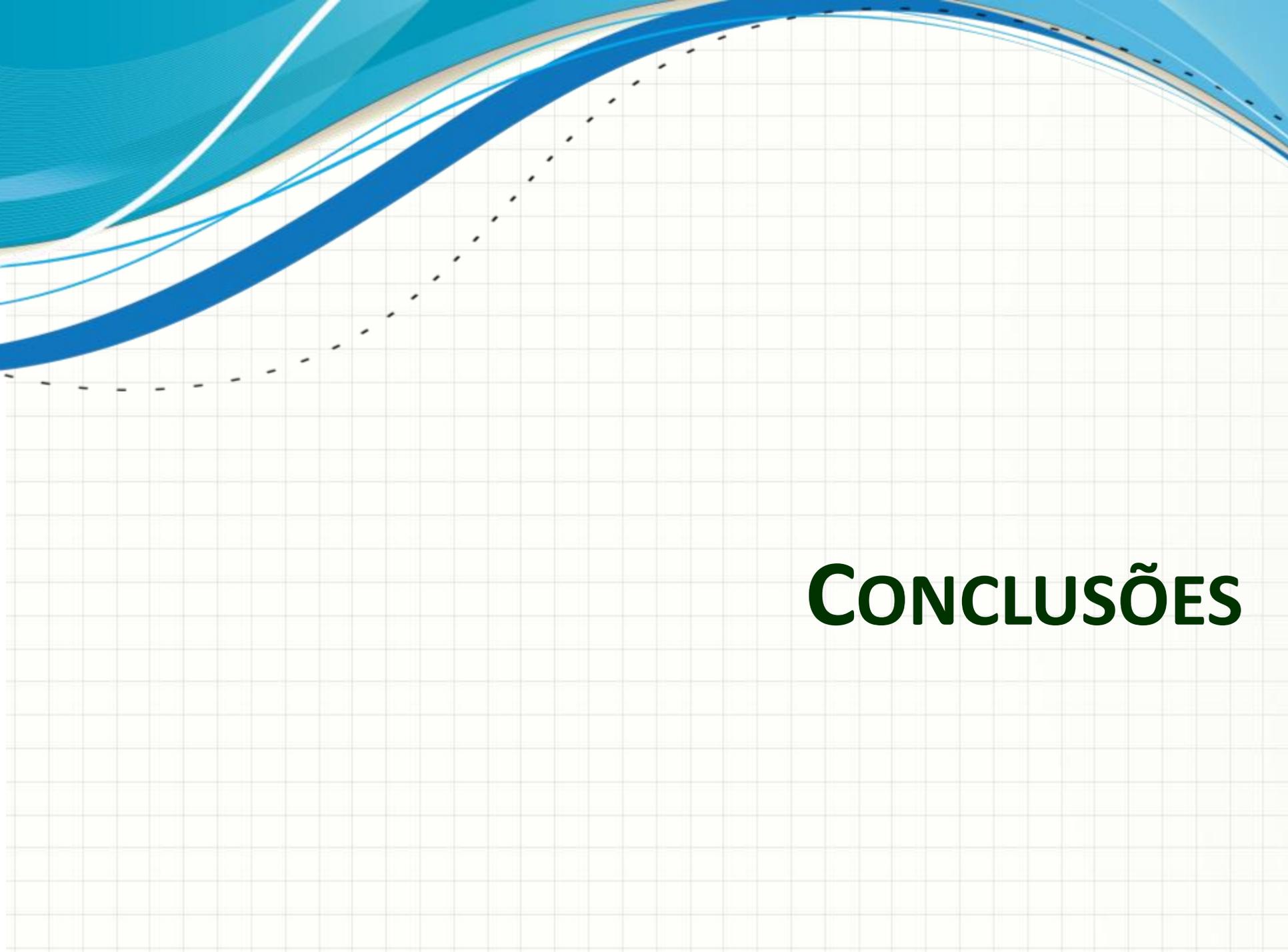
- Regra prática: converter 13 para binário

1101b

- $13/2 = 6...$ Resto 1
- $6/2 = 3...$ Resto 0
- $3/2 = 1...$ Resto 1
- $1/2 = 0...$ Resto 1
- 0 **Fim!**

Exercício 1

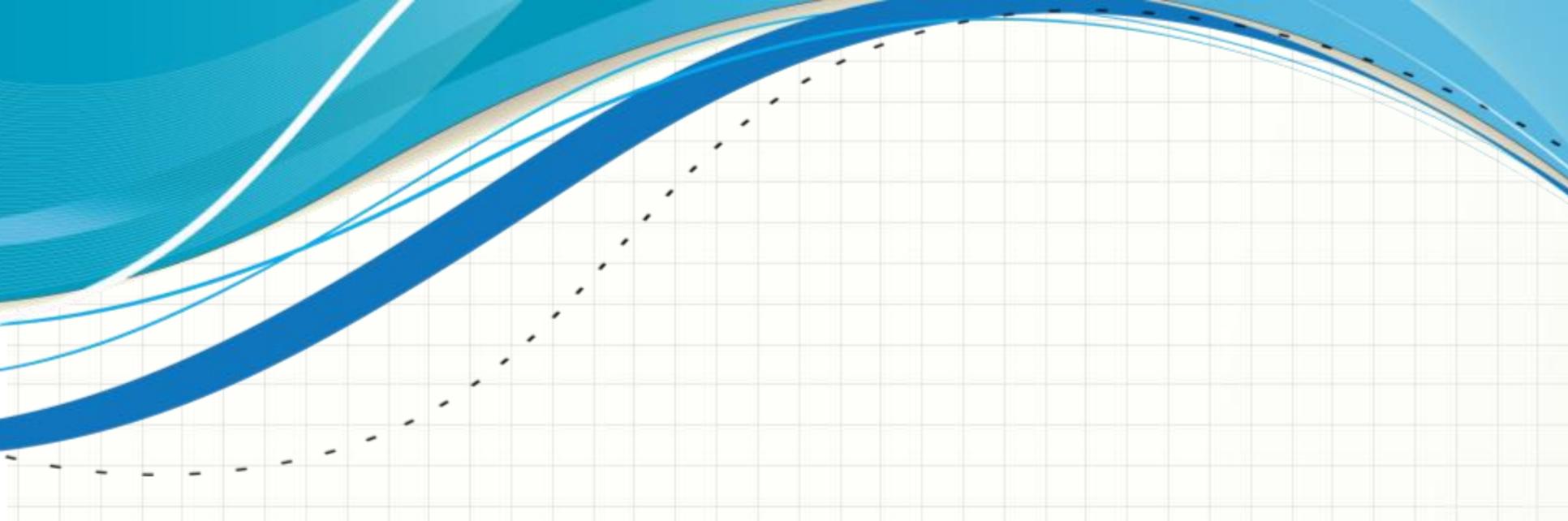
- O arquivo “ex1.cpp” contém a implementação de uma lista encadeada.
- O arquivo “binario.cpp” contém a implementação de conversão decimal -> binário em pilha contígua
- Modifique o “ex1.cpp” para implementar a conversão usando a lista encadeada como uma pilha



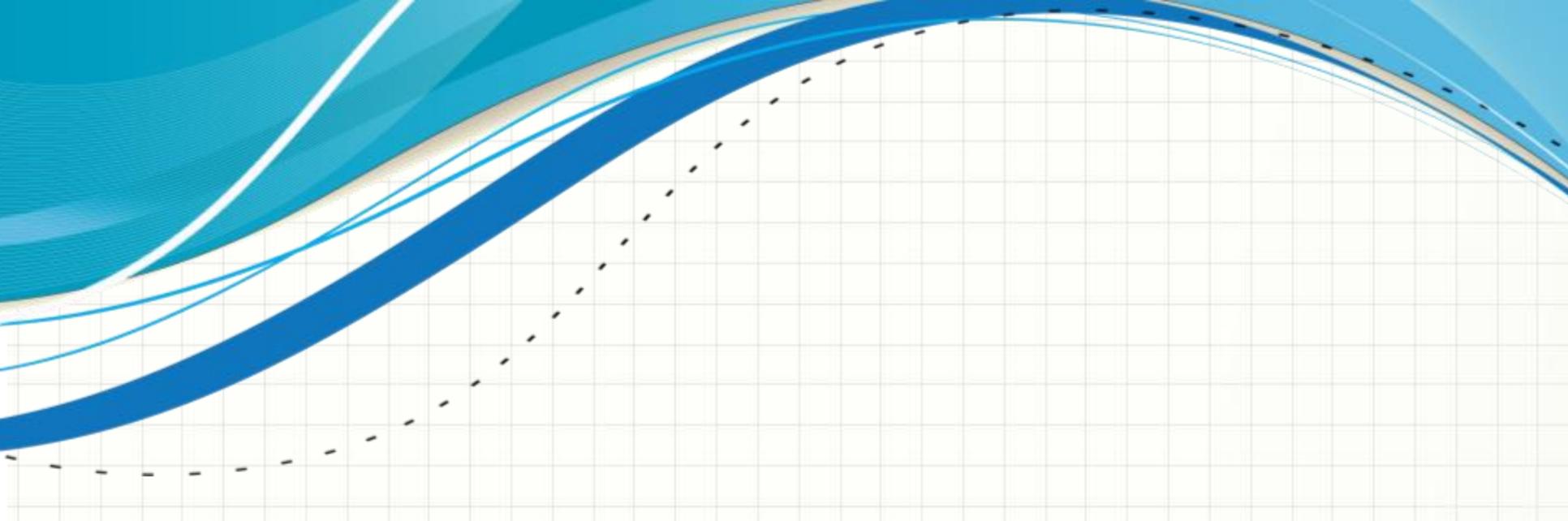
CONCLUSÕES

Resumo

- As listas encadeadas permitem seu uso na forma de pilha
 - Adaptar programas que usam listas e pilhas contíguas para usar listas ligadas é simples
-
- É possível implementar filas dinâmicas?



PERGUNTAS?



EXERCÍCIOS DE FIXAÇÃO

Exercício 2

- Como base em “ex2.cpp” implemente:
 - Modifique-o para que leia valores inteiros positivos... Até que o valor digitado seja ≤ 0
 - Faça o programa principal imprimir todos os dados da lista
 - Faça uma função que imprima apenas os dados ímpares (e use-a!)
 - Faça uma função que imprima apenas os múltiplos de 5 (e use-a!)
 - Faça uma função que conta os nós da lista e use-a!
 - Faça uma função que busque um valor e o substitua pelo seu triplo!

Exercício 3

- Como base em “ex3.cpp” e “calculadora.cpp”, implemente a calculadora usando pilha dinâmica.