

ESTRUTURA DE DADOS

LISTAS LINEARES ENCADEADAS

Prof. Dr. Daniel Caetano

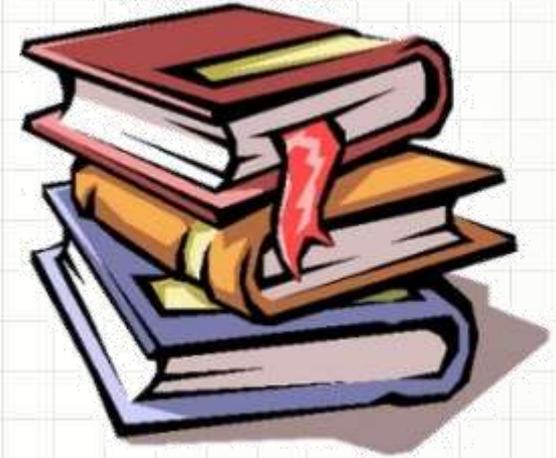
2012 - 2

Objetivos

- Conceituar Listas Lineares Encadeadas
- Compreender a alocação dinâmica e memória
- Representar Listas Simplesmente Encadeadas
- Implementar operações com as listas simplesmente encadeadas
- **Atividade Estruturada!**



Material de Estudo



Material

Acesso ao Material

Apresentação

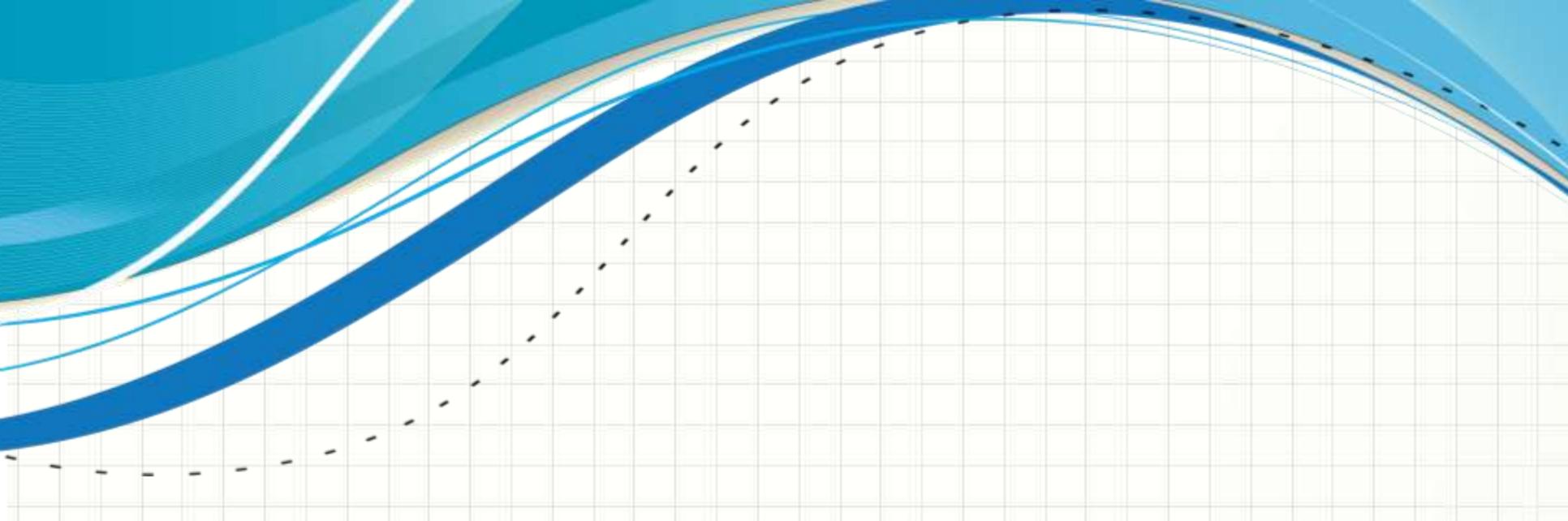
<http://www.caetano.eng.br/>
(Aula 10)

Material Didático

Estruturas de Dados (Parte 1) – Páginas 88 a 98

Online

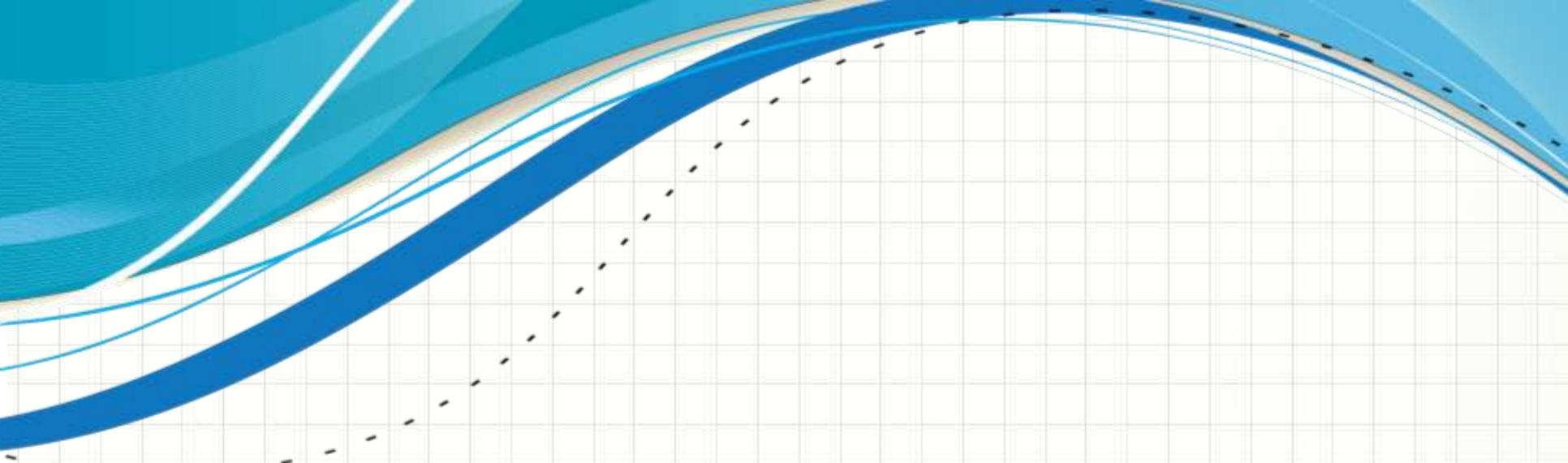
C Completo e Total – Páginas 540 a 550



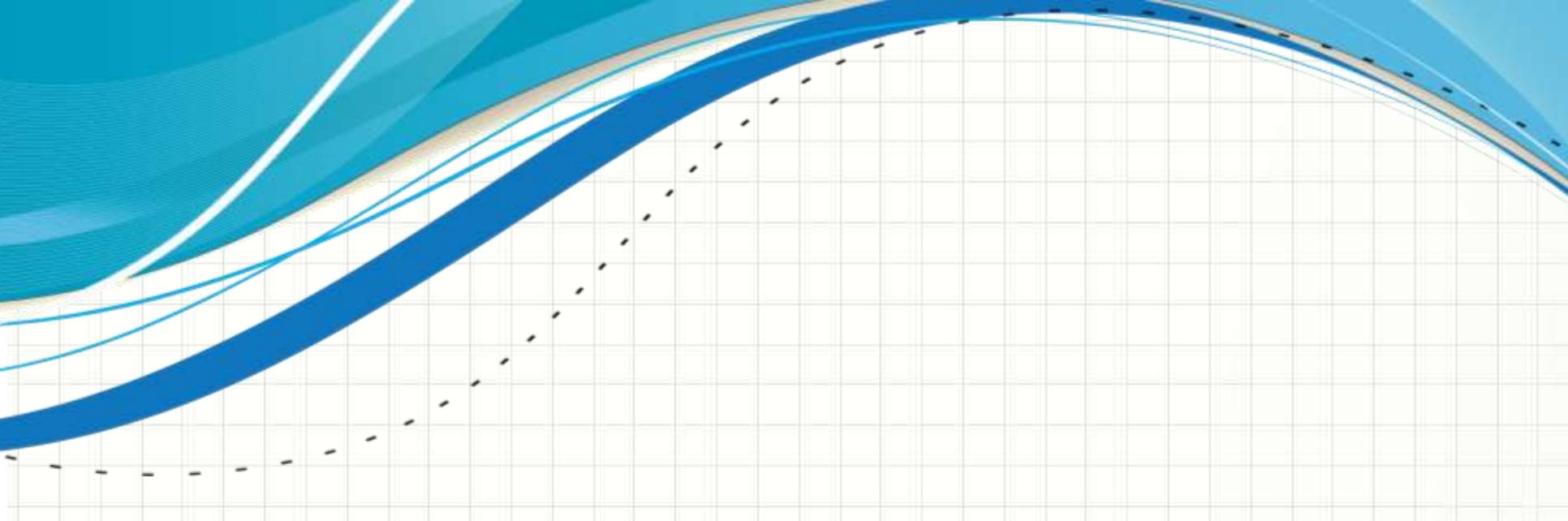
RECORDANDO...

Recordando...

- Vimos...
 - Listas Contíguas
- Problema:
 - Reservar espaço na memória
 - Conhecer tamanho máximo da lista
- Precisa ser desse jeito?
- Vamos ver um jeito diferente?



**MOMENTO LÚDICO:
ENTENDENDO UMA LISTA
ENCADEADA**



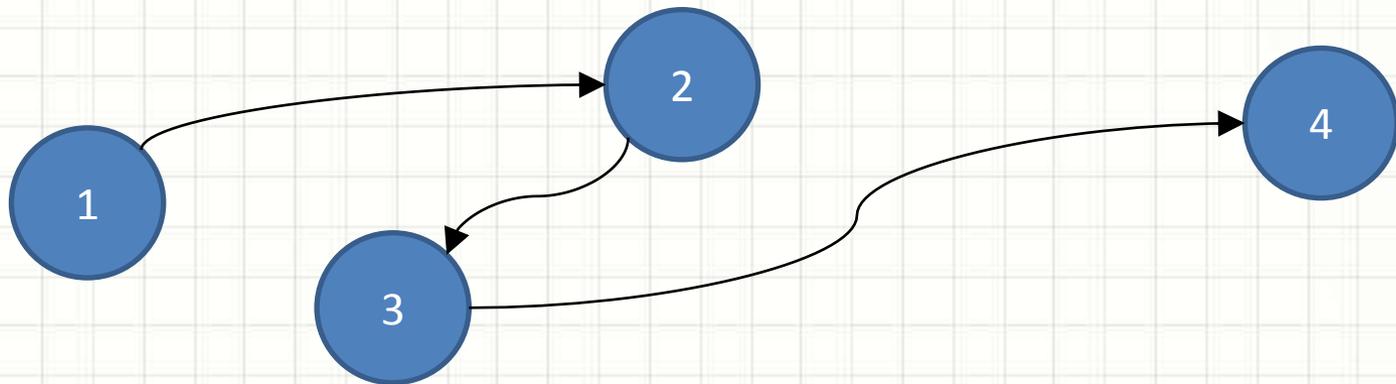
DISCUSSÃO SOBRE A DINÂMICA

Discussão sobre Listas Encadeadas

- Os itens estavam contíguos ou espalhados?
- É possível saber até onde esta lista pode crescer?
- Seria fácil acrescentar outro objeto na sequência?
 - Se for necessário mais um papel, o que haveria nesse papel?
 - Será necessário mudar alguma coisa nos outros papéis?

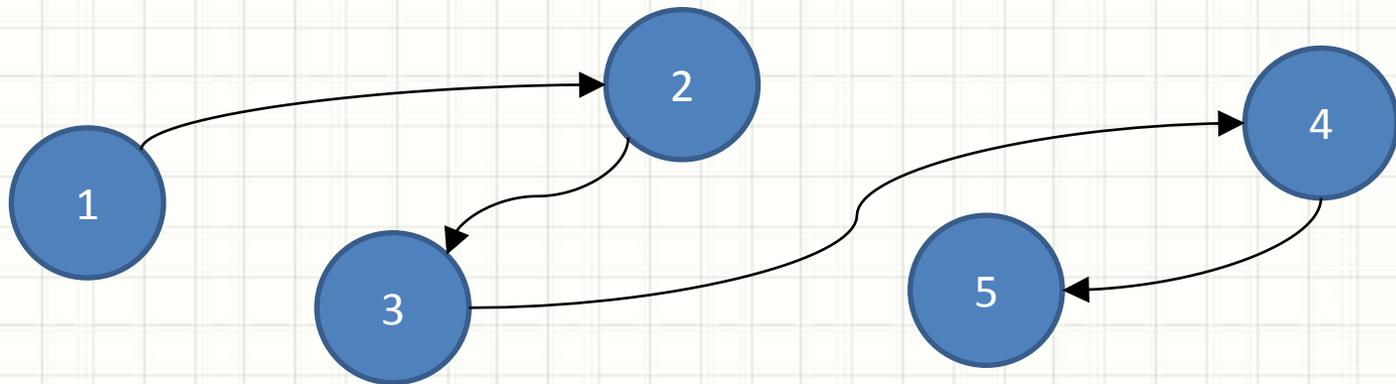
Discussão sobre Listas Encadeadas

- Lista Encadeada: lista de elementos interligados, como em uma corrente
 - O primeiro elemento aponta para o segundo
 - O segundo aponta para o terceiro
 - O terceiro aponta para o quarto
 - ...



Discussão sobre Listas Encadeadas

- Lista Encadeada: lista de elementos interligados, como em uma corrente
- Para acrescentar outro elemento no fim?
- Criar o novo elemento
- Associá-lo à lista (encadeá-lo)

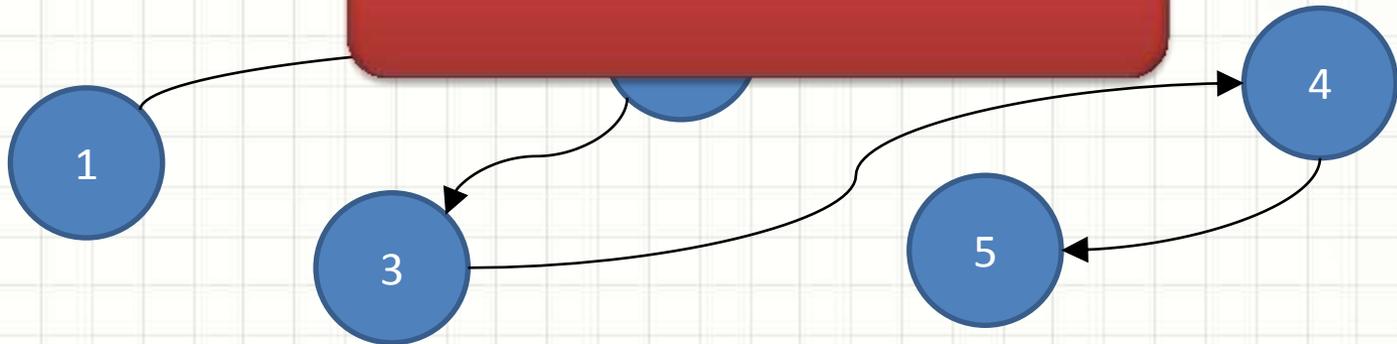


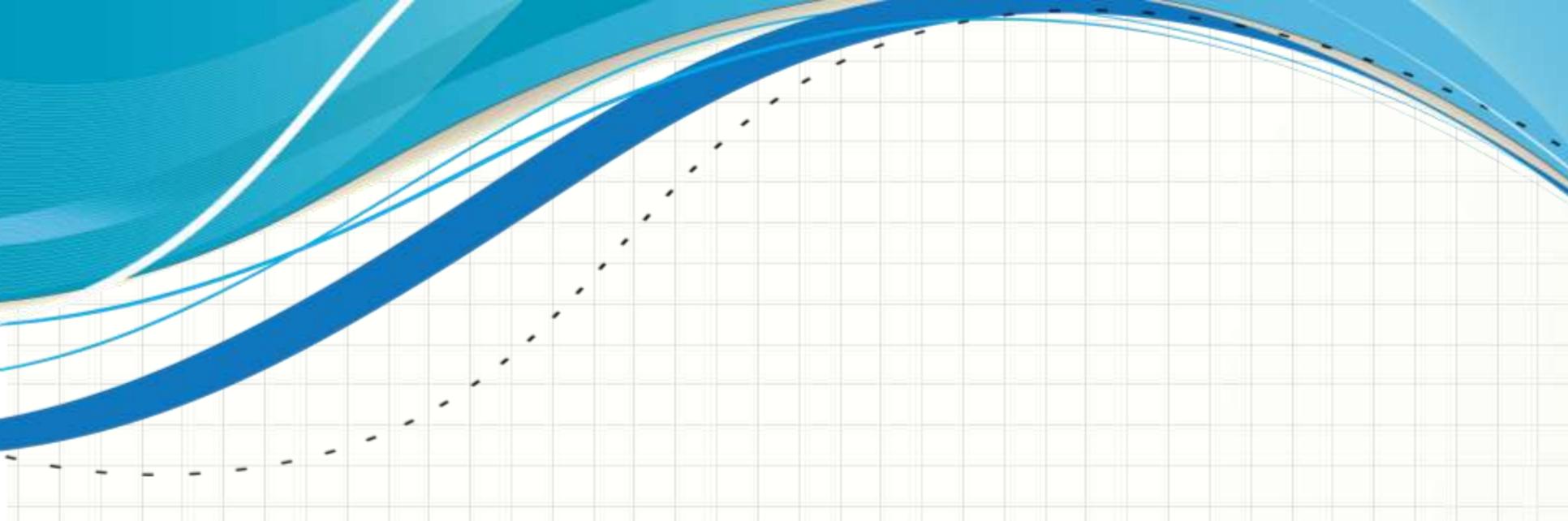
Discussão sobre Listas Encadeadas

- Lista Encadeada interligado
- Para acrescentar no fim?
- Criar o novo elemento
- Associá-lo

Como criar novos elementos?

Já vimos isso?





ALOCAÇÃO DINÂMICA DE MEMÓRIA

Alocação Dinâmica de Memória

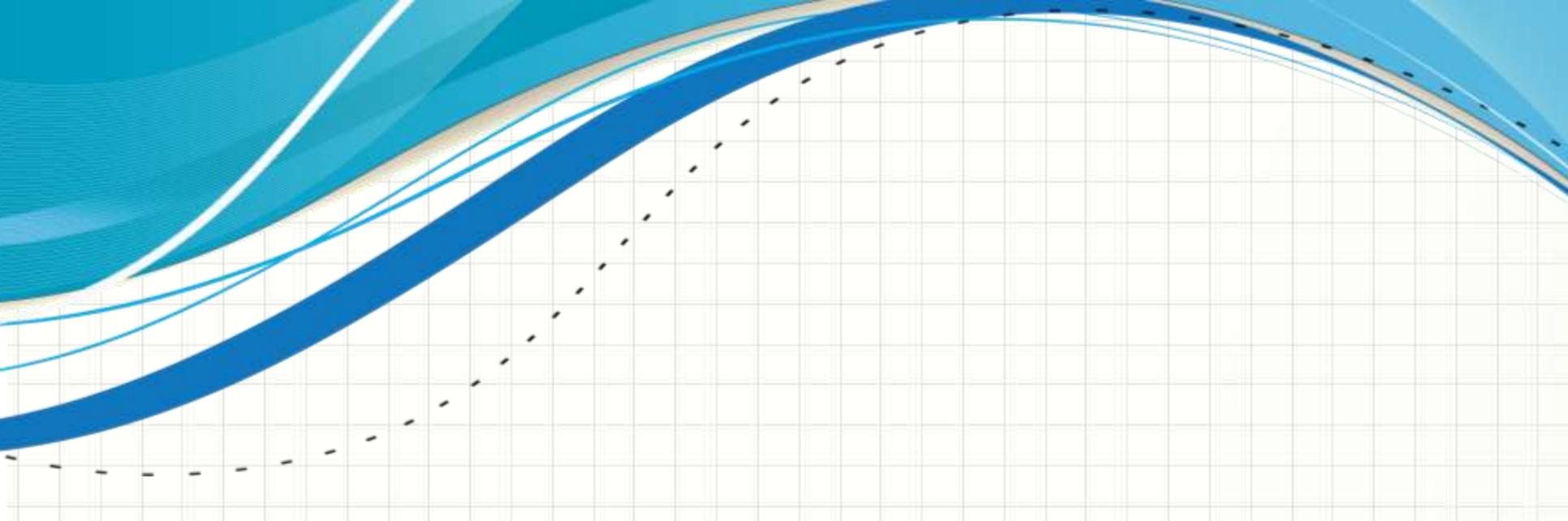
- Alocar memória estaticamente: declarar
 - int, float, char etc.
- Alocação dinâmica?
 - Reservar no momento necessário
 - Liberar quando não for mais necessário
- Reservar: **new**
- Liberar: **delete**

Alocação Dinâmica de Memória

- Modo de Usar: **new** e **delete**

```
int *p;  
p = new int;  
*p = 10;  
cout << p << endl;  
cout << *p << endl;  
delete p;
```

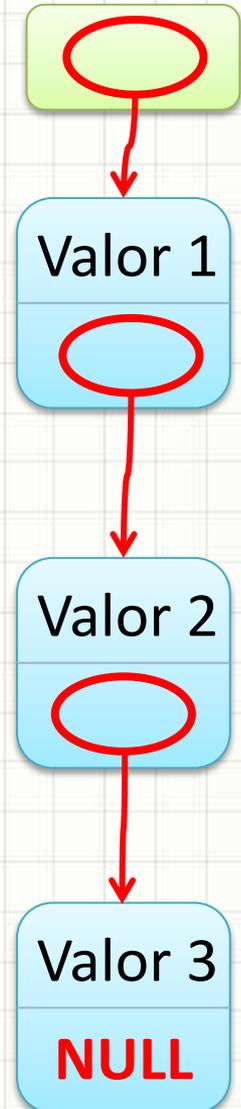
- Acompanhe os exemplos



REPRESENTANDO LISTAS ENCADEADAS

Representando Listas Encadeadas

- Como vimos na dinâmica:
 - Lista encadeada é composta por **nós**
- Um nó tem a função de...
 - Guardar um elemento da lista
- Que outra informação tem o nó?
 - Um ponteiro que indique o próximo nó
- E quando não há um próximo nó?
 - Ajustamos o ponteiro para valer NULL
- Onde começa uma lista encadeada?
 - Precisamos de um ponteiro para isso!



Representando Listas Encadeadas



Cada elemento da lista encadeada é uma **struct**

Criaremos uma struct chamada **no**

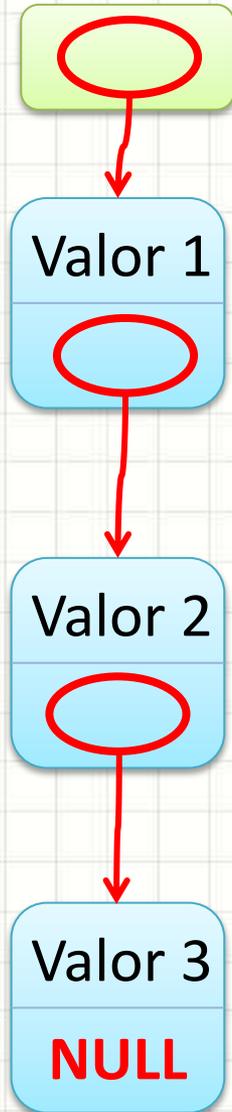
Representando Listas Encadeadas



Cada "no" é composto de uma **informação**

E de um **ponteiro** para outro "no"

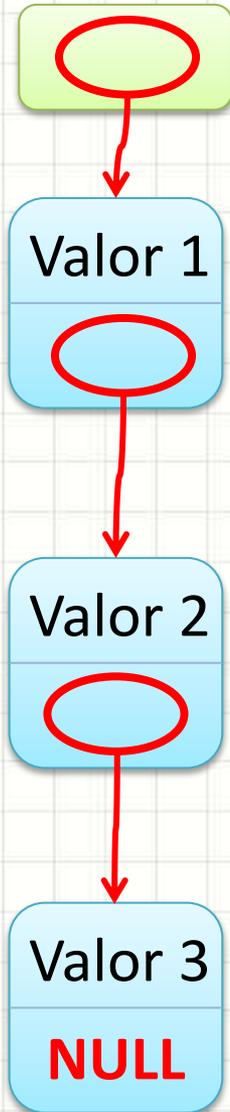
Representando Listas Encadeadas



```
struct no {  
    int valor;  
    no * ptr;  
};
```

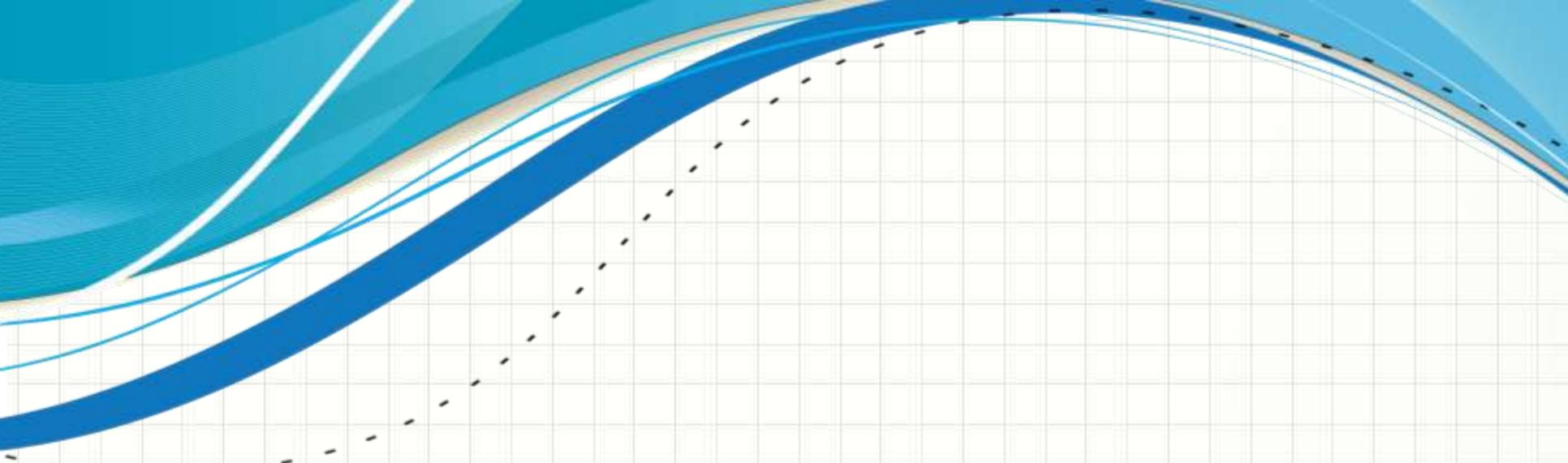
O primeiro nó é apontado por um simples ponteiro

Representando Listas Encadeadas



```
struct no {  
    int valor;  
    no *ptr;  
};
```

```
no *lista;
```



**INICIALIZANDO
LISTAS ENCADEADAS E
INSERINDO NÓS**

Listas Encadeadas: Inicializando

lista

NULL

Quando criamos uma lista, ela está vazia

```
no *lista = NULL;
```

Listas Encadeadas: Inserção

lista 

tmp 

Como criamos um elemento?

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

Listas Encadeadas: Inserção

lista 

tmp 

Como colocar o elemento na lista?

```
if (lista == NULL) {  
    lista = tmp;  
}
```

Listas Encadeadas: Inserção

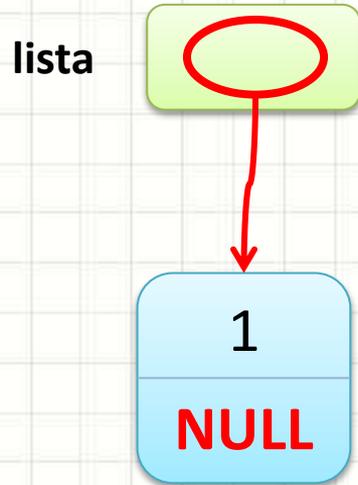


Como colocar o elemento na lista?

```
if (lista == NULL) {  
    lista = tmp;  
}
```



Listas Encadeadas: Inserção

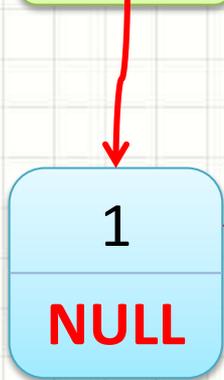


```
if (lista == NULL) {  
    lista = tmp;  
}
```



E para acrescentar um outro elemento?

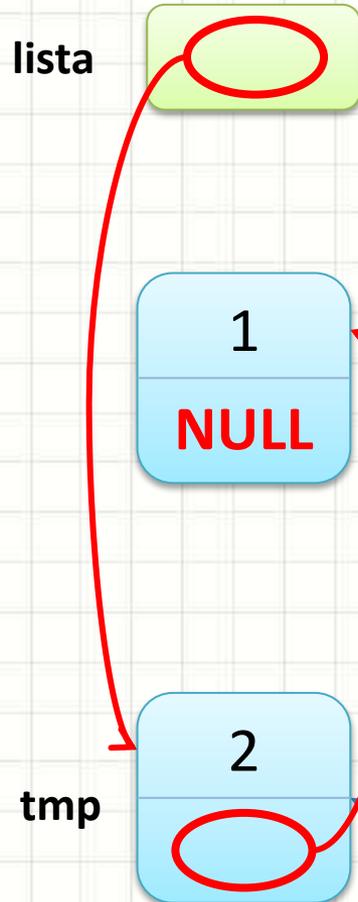
Listas Encadeadas: Inserção



```
if (lista == NULL) {  
    lista = tmp;  
} else {  
    tmp->ptr = lista  
}
```

E para acrescentar um outro elemento?

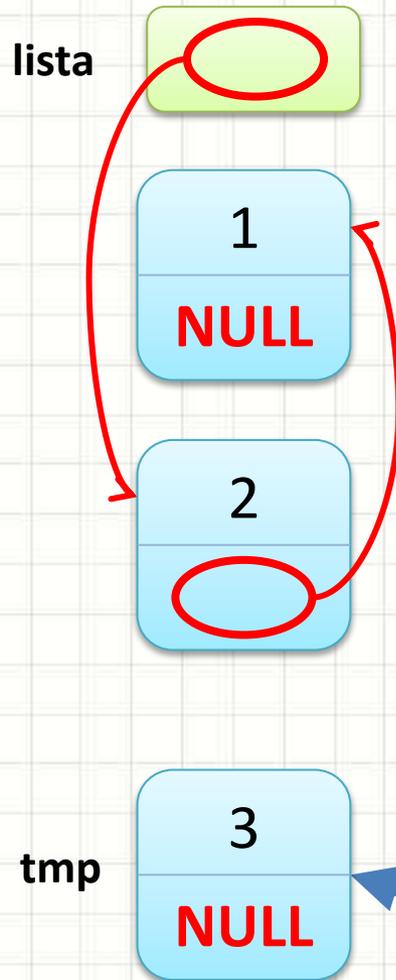
Listas Encadeadas: Inserção



```
if (lista == NULL) {  
    lista = tmp;  
} else {  
    tmp->ptr = lista  
    lista = tmp;  
}
```

E para acrescentar um outro elemento?

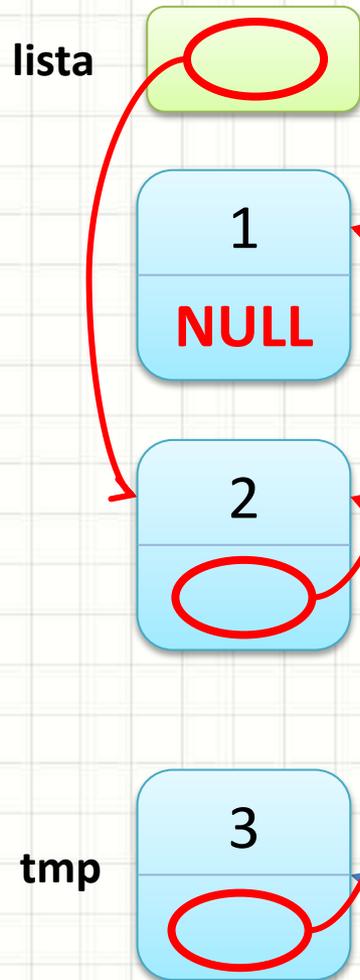
Listas Encadeadas: Inserção



```
if (lista == NULL) {  
    lista = tmp;  
} else {  
    tmp->ptr = lista  
    lista = tmp;  
}
```

E para acrescentar um
outro elemento?

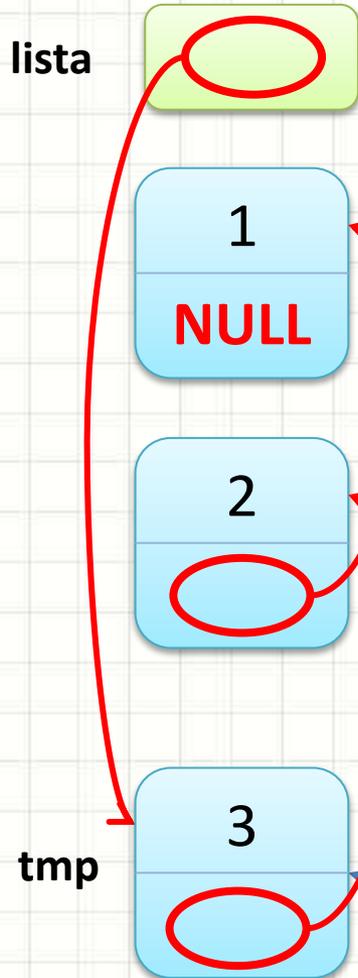
Listas Encadeadas: Inserção



```
if (lista == NULL) {  
    lista = tmp;  
} else {  
    tmp->ptr = lista  
    lista = tmp;  
}
```

E para acrescentar um
outro elemento?

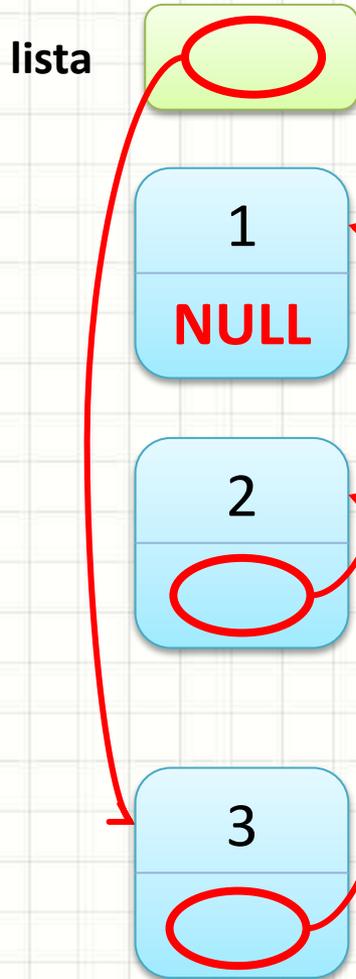
Listas Encadeadas: Inserção



```
if (lista == NULL) {  
    lista = tmp;  
} else {  
    tmp->ptr = lista  
    lista = tmp;  
}
```

E para acrescentar um
outro elemento?

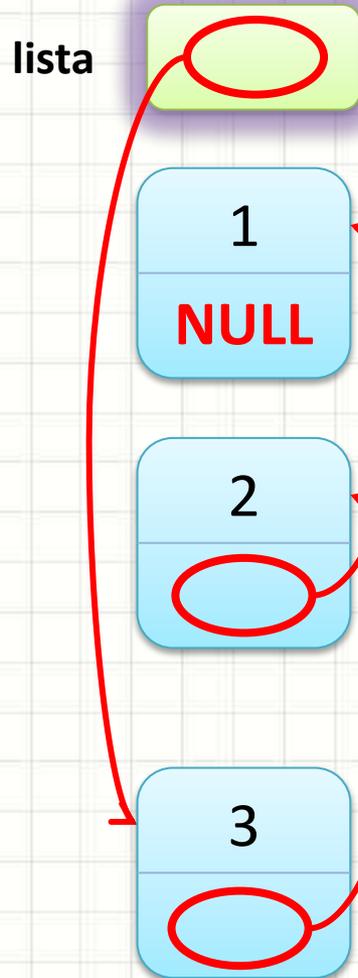
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

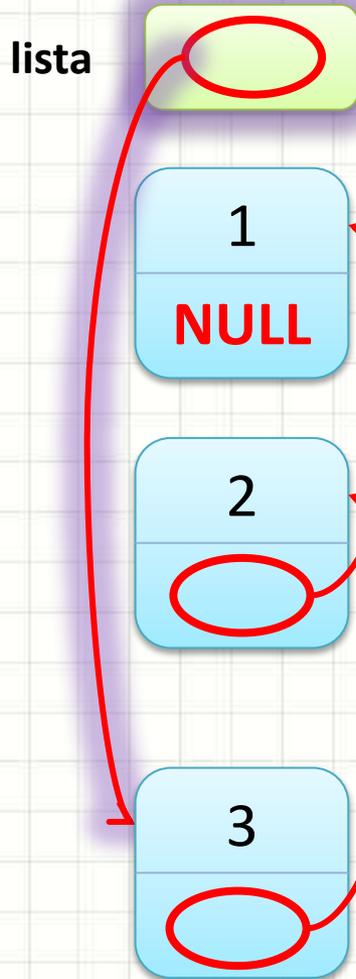
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

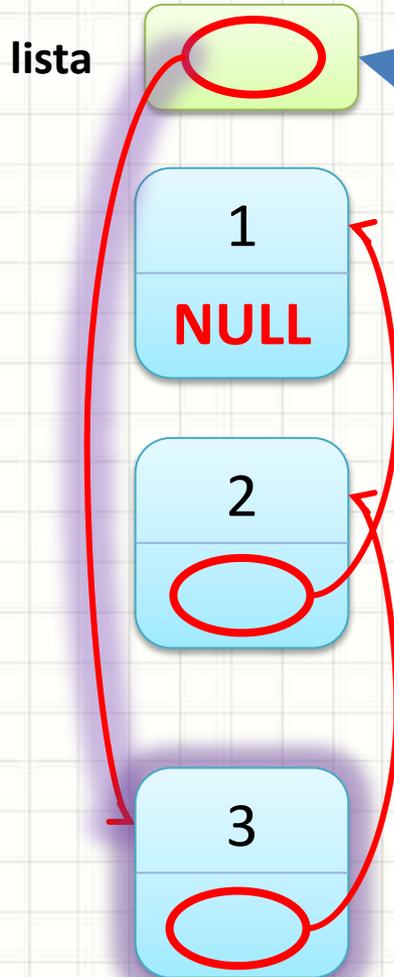
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

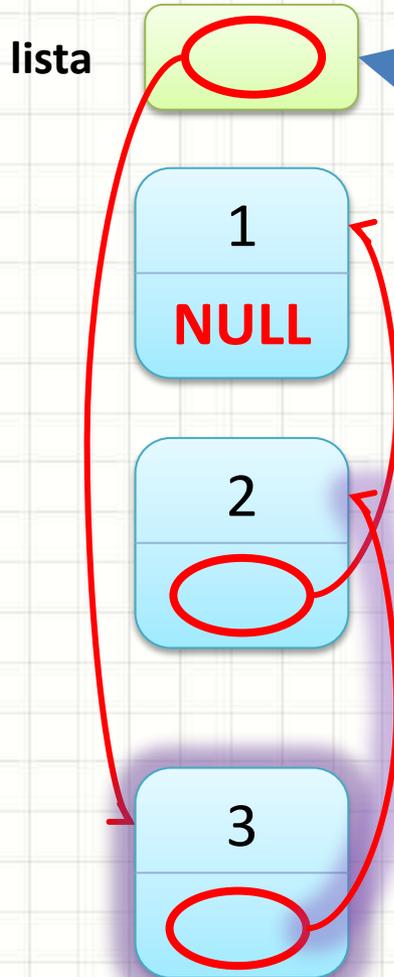
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

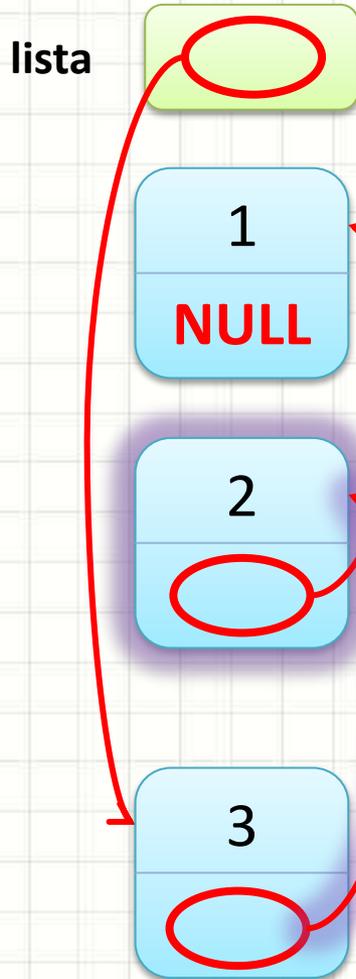
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

Listas Encadeadas: Listar/Percorrer

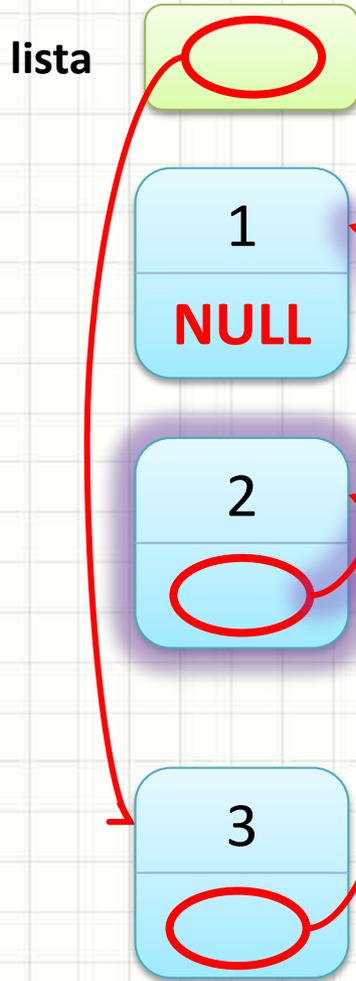


Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

Listas Encadeadas: Listar/Percorrer

lista



1
NULL

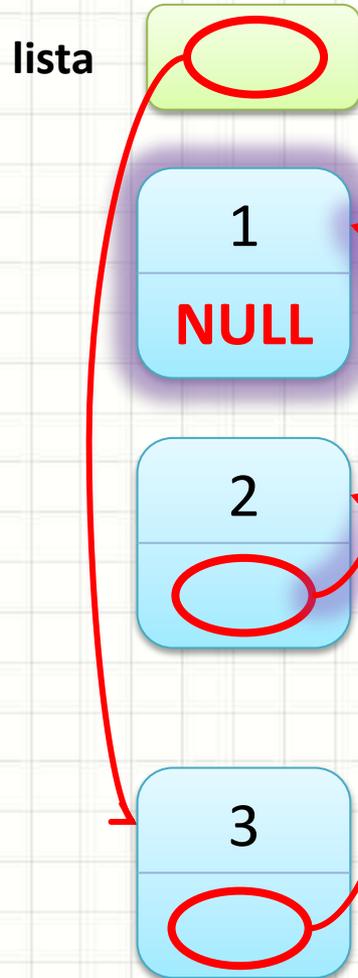
2

3

Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

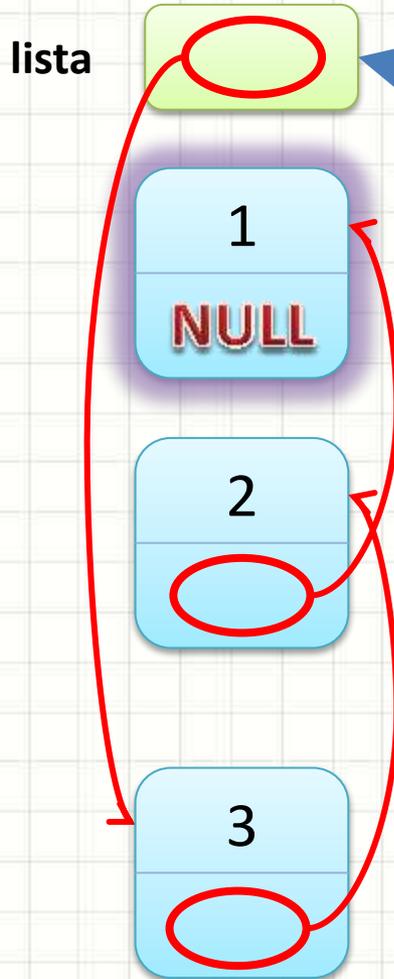
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

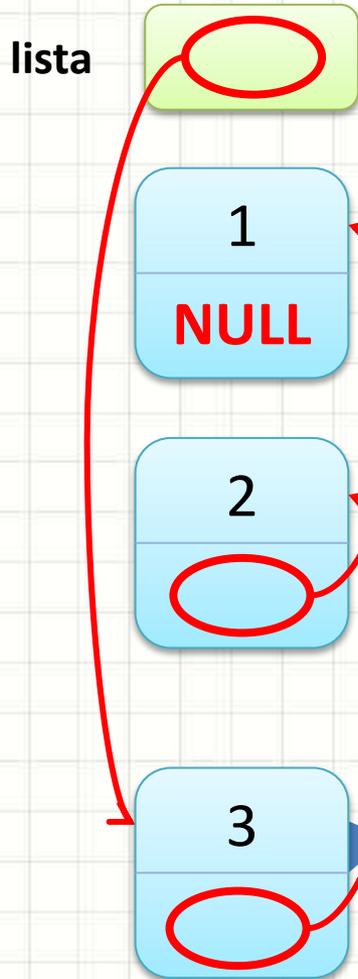
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

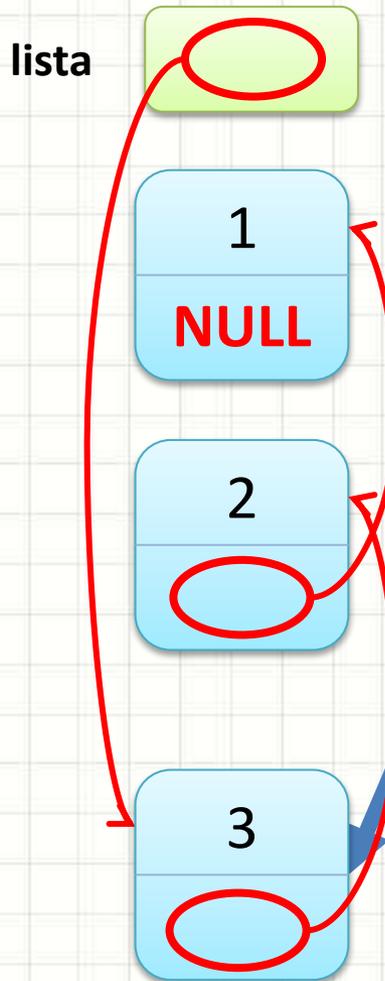
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

```
if (lista != NULL) {  
    tmp = lista->ptr;  
    lista->ptr = tmp->ptr;  
    delete tmp;  
}
```

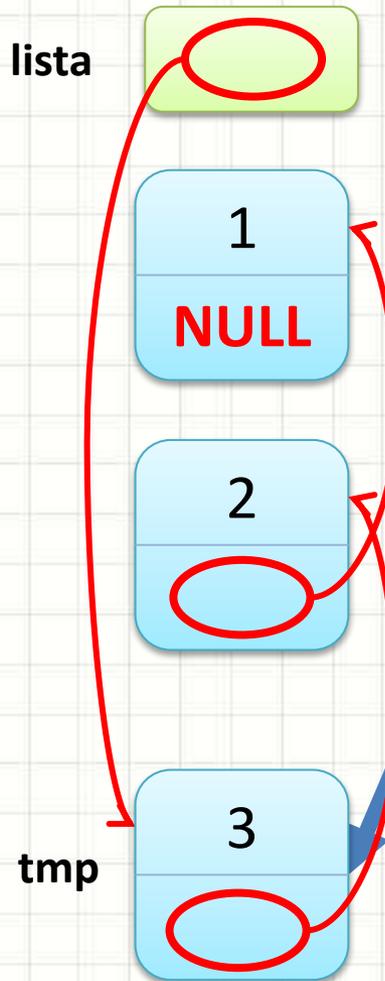
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

```
if (lista != NULL) {  
    tmp = lista->ptr;  
    lista->ptr = tmp->ptr;  
    delete tmp;  
}
```

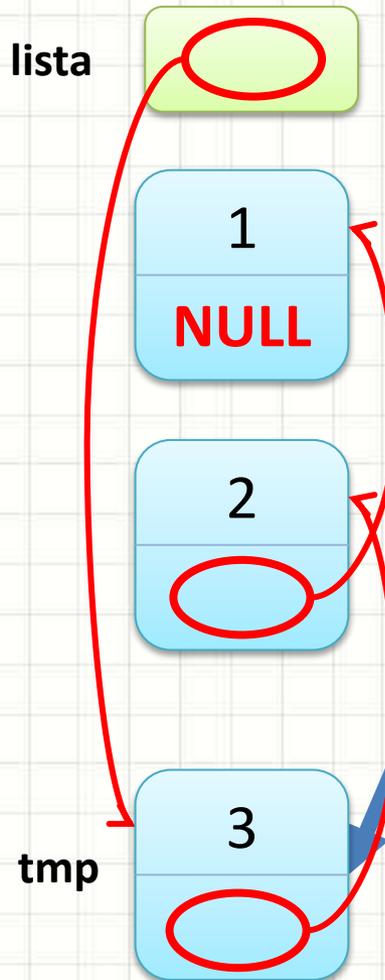
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

```
if (lista != NULL) {  
    tmp = lista->ptr;  
    lista->ptr = tmp->ptr;  
    delete tmp;  
}
```

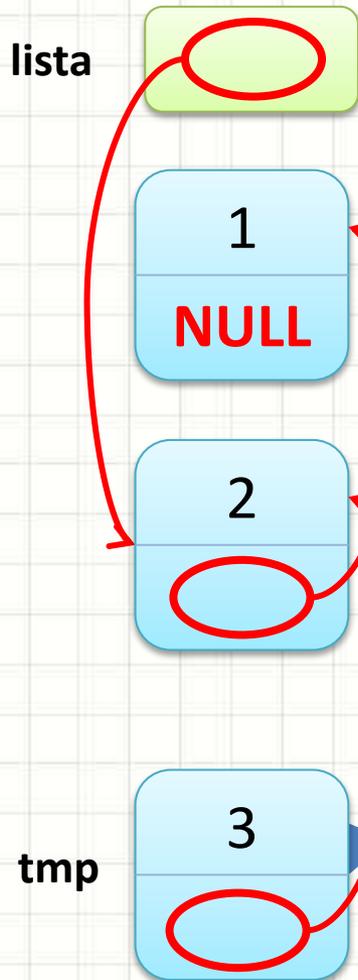
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

```
if (lista != NULL) {  
    tmp = lista->ptr;  
    lista->ptr = tmp->ptr;  
    delete tmp;  
}
```

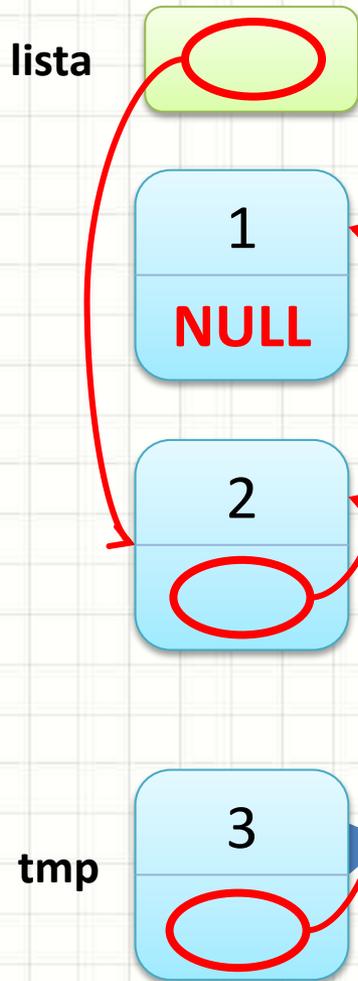
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

```
if (lista != NULL) {  
    tmp = lista->ptr;  
    lista->ptr = tmp->ptr;  
    delete tmp;  
}
```

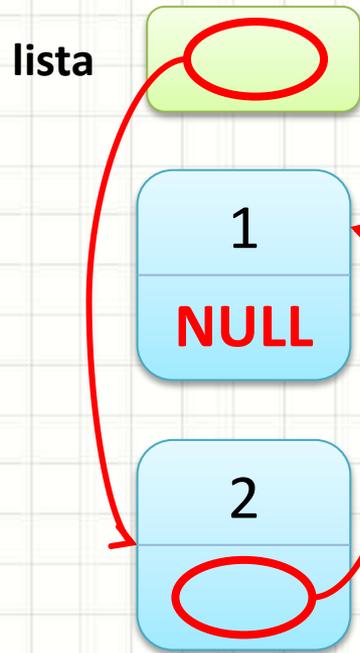
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

```
if (lista != NULL) {  
    tmp = lista->ptr;  
    lista->ptr = tmp->ptr;  
    delete tmp;  
}
```

Listas Encadeadas: Remoção

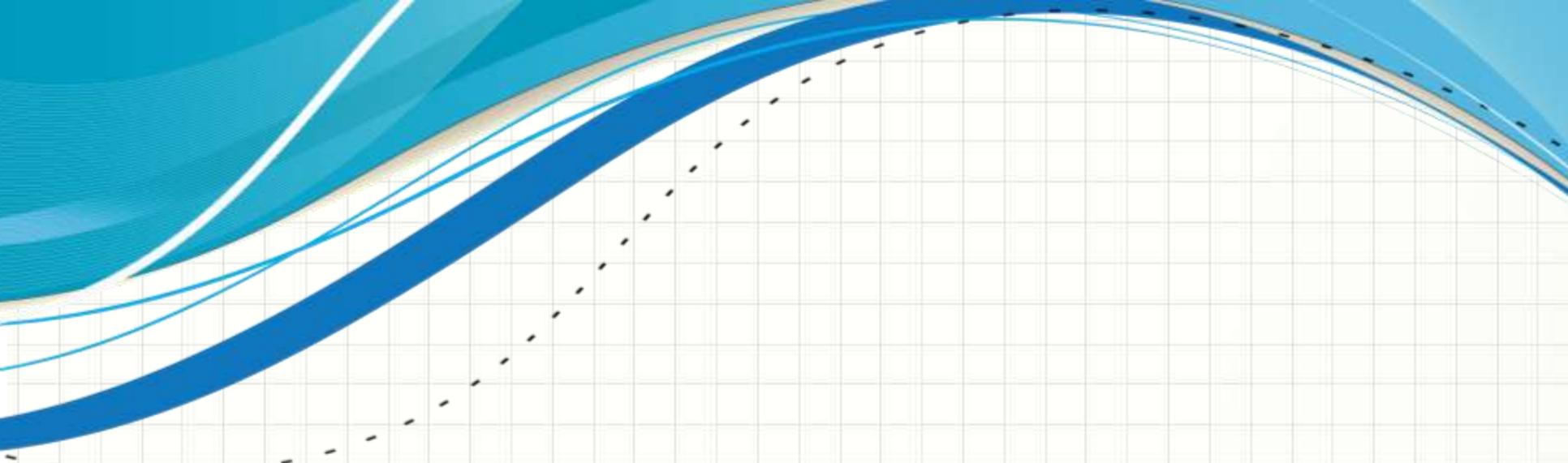


Como remover o primeiro elemento?

```
if (lista != NULL) {  
    tmp = lista->ptr;  
    lista->ptr = tmp->ptr;  
    delete tmp;  
}
```

Operações Com Listas Encadeadas

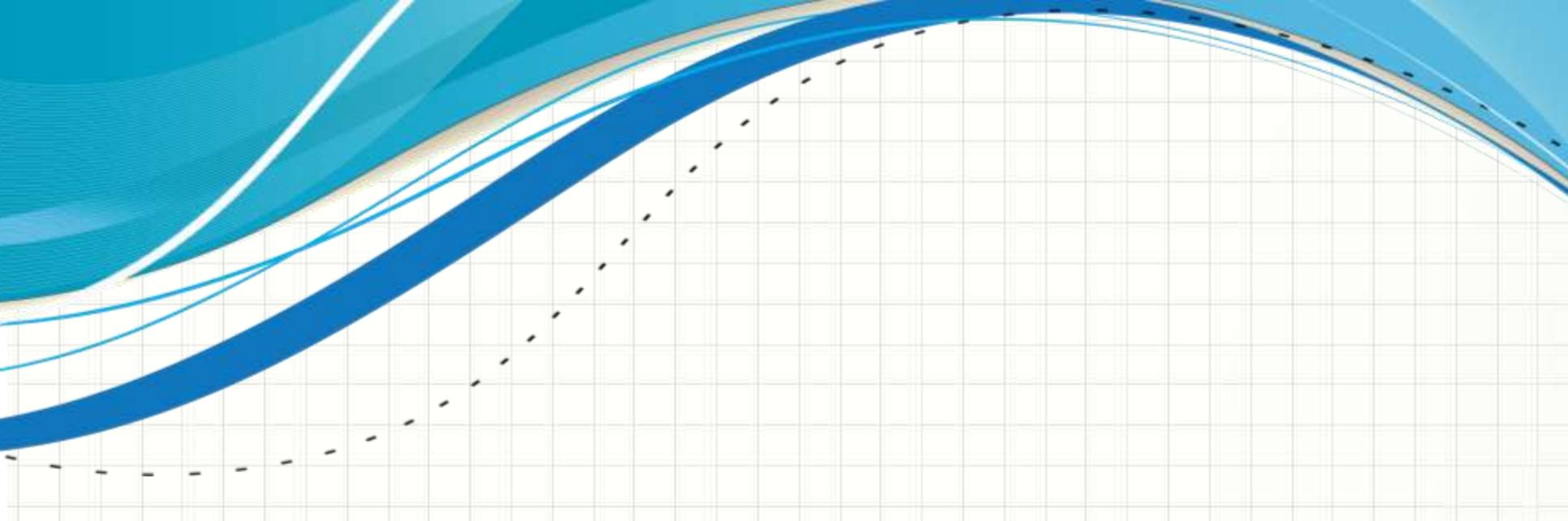
- Vistas
 - Inicialização
 - Inserção no Início
 - Listar / Percorrer
 - Remover
- Para pensar:
 - Buscar
 - Inserir no final
 - Inserir no meio
 - Remover do meio



NA PRÁTICA:
IMPLEMENTANDO UMA
LISTA ENCADEADA

Implementação de Lista Encadeada

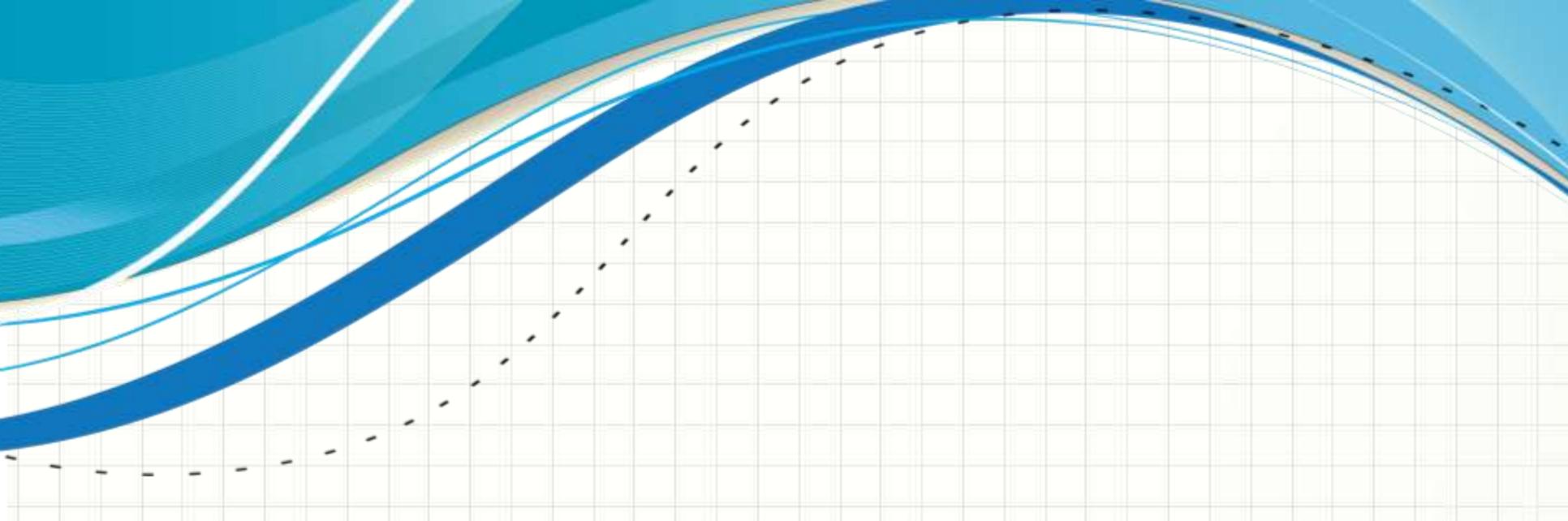
- Acompanhe o professor... Inicialização e inserção!
- Criação de uma lista com “n” nós
- Função Listar/Percorrer
- Função de Substituição de valores
- Função de Remoção de Nó



EXERCÍCIOS DE FIXAÇÃO

Exercício

- Crie uma função para BUSCAR o primeiro nó que possua um determinado valor:
no * buscar(no * lista, int valor)
- Modifique o programa principal para que o usuário possa verificar se um valor específico ocorre na lista
- Após a tentativa de execução, “visite” pelo menos dois colegas e veja como ele implementou a solução; tente compreender a forma do colega de raciocinar.



CONCLUSÕES

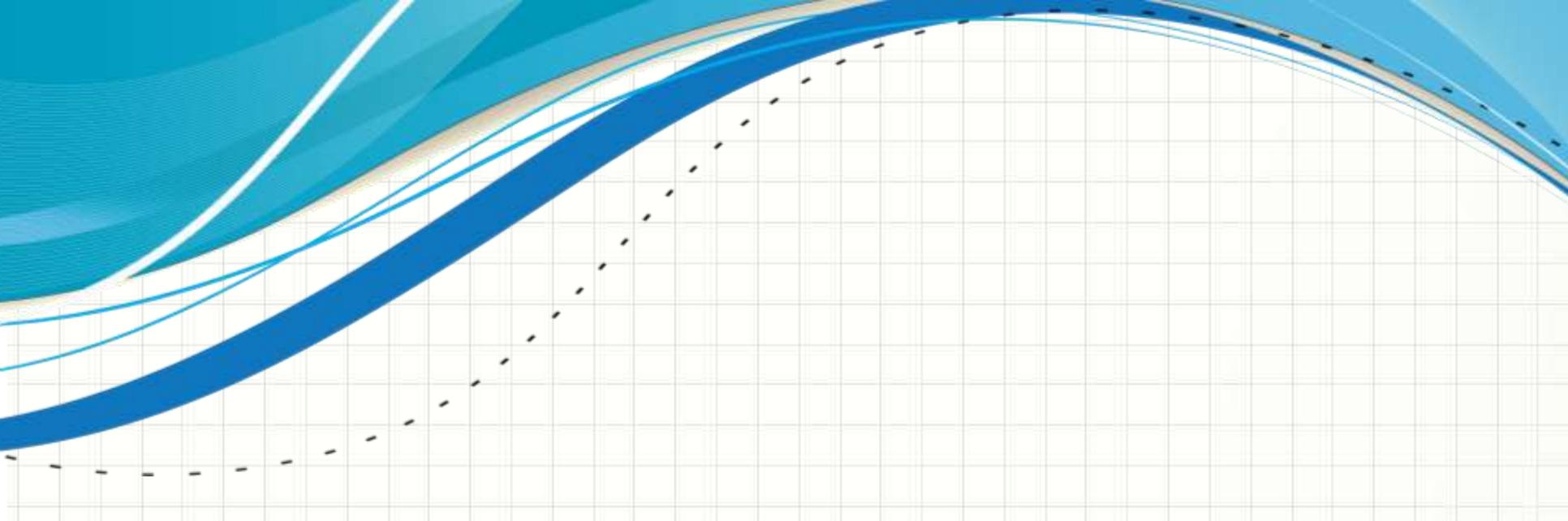
Resumo

- As listas encadeadas permitem construir listas de tamanho arbitrário
- Apenas a quantidade de memória necessária é gasta
- O jeito de programar é diferente, mas as funções são muito parecidas!
- **TAREFA**
 - Atividade Estruturada!

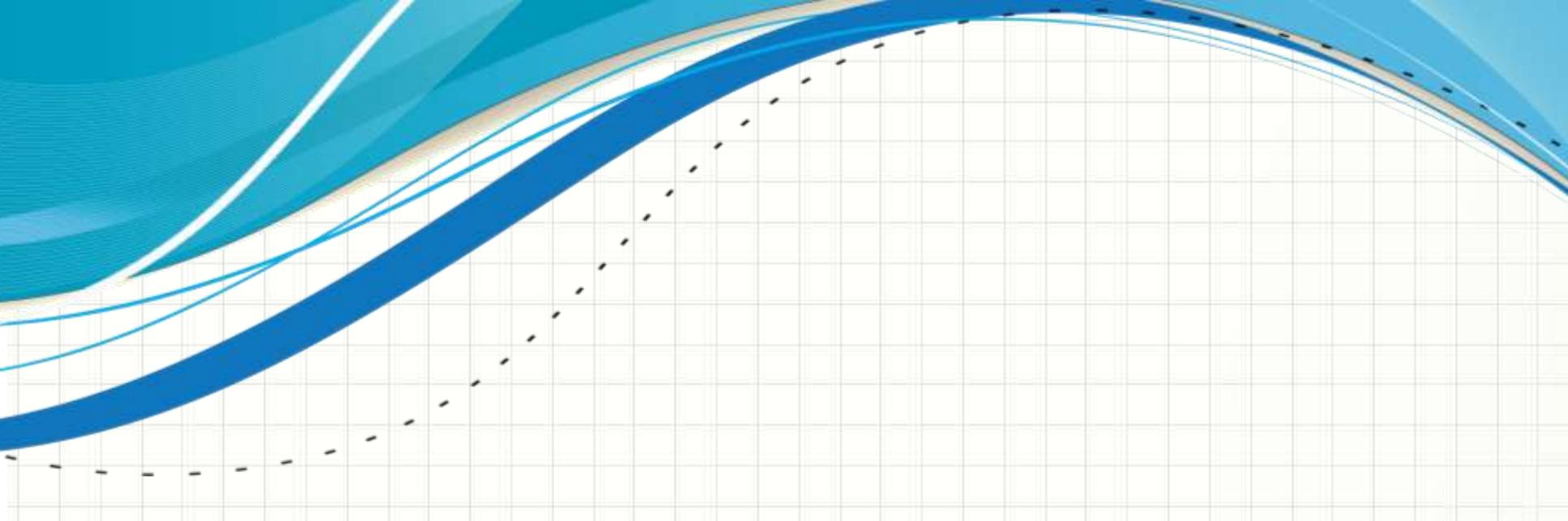
Próxima Aula



- Como usar esse conceito para construir uma pilha?



PERGUNTAS?



**BOM DESCANSO
A TODOS!**