Paradigmas de Linguagens de Programação em Python

Tipos de Dados Avançados II

Prof. Dr. Daniel Caetano

2022 - 1

Compreendendo o problema

- **Situação:** sistema da universidade: matrícula e média
  - Porém... Não sabemos o número de alunos da universidade

Aluno

- Matrícula (int)
- Média (float)

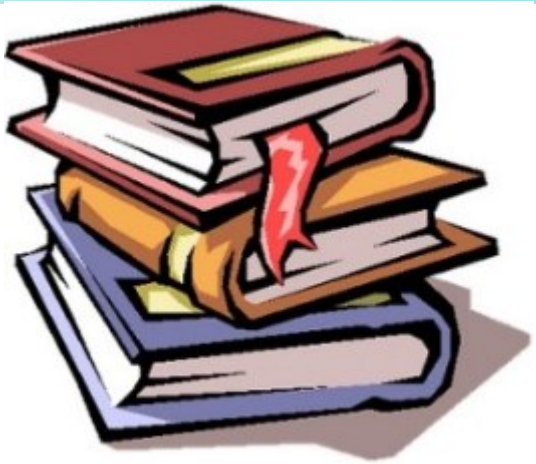
  - Em C/C++... Como declarar?

https://www.menti.com/

Objetivos

- Compreender o conceito de variável-ponteiro
- Compreender o uso de alocação de memória
- Compreender diferenças das listas dinâmicas

Bibliografia da Aula



Material	Acesso ao Material
Apresentação	https://www.caetano.eng.br/aulas/2022a/ara0066.php (Paradigmas de Programação – Aula 09)
Livro Texto	Capítulo 6, páginas 273 a 287
Aprenda Mais!	• Vídeo: Listas Encadeadas (ative legenda traduzida!) https://youtu.be/njTh_OwMljA

ENDEREÇOS

https://www.menti.com/

Recordando: Endereços

- Onde fica armazenado o valor da variável?
  - Memória!
- Como é a memória do computador?
  - Um monte de “gavetas”: **posição de memória**
  - Cada gaveta tem um número único: **endereço**
- Cada nome de variável → endereço
  - Declarar variável =?
  - Armazenar valor em variável = ?

Recordando: Endereços

- Cada nome de variável → endereço

Endereço	0	1	2	3	4	5	6	7
Identificador								
Valor	??	??	??	??	??	??	??	??

- char letra;

Recordando: Endereços

- Cada nome de variável → endereço

Endereço	0	1	2	3	4	5	6	7
Identificador	letra							
Valor	??	??	??	??	??	??	??	??

- char letra;
- int idade;

Recordando: Endereços

- Cada nome de variável → endereço

Endereço	0	1	2	3	4	5	6	7
Identificador	letra	idade						
Valor	??	??	??	??	??	??	??	??

- char letra;
- int idade;
- char a[3];

Recordando: Endereços

- Cada nome de variável → endereço

Endereço	0	1	2	3	4	5	6	7
Identificador	letra	idade				a		
Valor	??	??	??	??	??	??	??	??

- char letra;
- int idade;
- char a[3];
- letra = ‘c’;

Recordando: Endereços

- Cada nome de variável → endereço

Endereço	0	1	2	3	4	5	6	7
Identificador	letra	idade				a		
Valor	99	??	??	??	??	??	??	??

- char letra;
- int idade;
- char a[3];
- letra = ‘c’;
- A[1] = ‘d’;

Recordando: Endereços

- Cada nome de variável → endereço

Endereço	0	1	2	3	4	5	6	7
Identificador	letra	idade				a		
Valor	99	??	??	??	??	??	100	??

- char letra;
- int idade;
- char a[3];
- letra = ‘c’;
- A[1] = ‘d’;
- idade = 256;

Recordando: Endereços

- Cada nome de variável → endereço

Endereço	0	1	2	3	4	5	6	7
Identificador	letra	idade				a		
Valor	99	0	1	0	0	??	100	??

- char letra;
- int idade;
- char a[3];
- letra = ‘c’;
- A[1] = ‘d’;
- idade = 256;

Como saber o endereço de uma variável?

Prefixo &

Em Python, usamos a função id()

Recordando: Endereços

- **Exemplo 1**

```
int a=1, b=2;
cout << "Val.A: " << a << endl;
cout << "End.A: " << &a << endl;
cout << "Val.B: " << b << endl;
cout << "End.B: " << &b << endl;
```

Recordando: Endereços

- Podemos guardar endereço em uma variável?

```
int a=1, b;
b = &a;
cout << "End.A: " << &a << endl;
cout << "Val.B: " << b << endl;
```

Exemplo 2

Recordando: Endereços

- Podemos guardar endereço em uma variável?

```
int a=1, b;
b = &a;
cout << "End.A: " << &a << endl;
cout << "Val.B: " << b << endl;
```

Exemplo 2

- Funcionou?
- Precisamos de um tipo novo de variável...

PONTEIROS

Ponteiros

- Ponteiro: serve para armazenar um endereço
  - Indicado por um * na frente do nome da variável

Exemplo 3

```
int a=1, *b;
b = &a;
cout << "End.A: " << &a << endl;
cout << "Val.B: " << b << endl;
```

Ponteiros

- Ponteiro: serve para armazenar um endereço
  - Indicado por um * na frente do nome da variável

```
int a=1, *b;
b = &a;
cout << "End.A: " << &a << endl;
cout << "Val.B: " << b << endl;
```

- E se quisermos saber o valor da variável "**apontada**"?
  - Usamos o ***** na frente do nome do ponteiro

Ponteiros

- **Exemplo 4**: Lendo valor apontado

```
int a=1, *b;
b = &a;
cout << "Val.A: " << a <<endl;
cout << "End.A: " << &a <<endl;
cout << "Val.B: " << b <<endl;
cout << "Val.*B: " << *b <<endl;
```

Ponteiros

- Acompanhe o **exemplo 5**:
  - Conhecendo o básico sobre ponteiros.

Ponteiros

- Acompanhe o **exemplo 6**:
  - Mudando valores usando ponteiros.

Ponteiros

- Acompanhe o **exemplo 7**:
  - Ponteiros nulos: NULL.

Ponteiros

- Acompanhe o **exemplo 8**:
  - Ponteiros para estruturas.

Ponteiros

- Acompanhe o **exemplo 9**:
  - Ponteiros para estruturas e o referenciador **->** .

Alocação Dinâmica de Memória

Prof. Dr. Daniel Caetano

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;
```

Exemplo 10

- Vejamos!

Exemplo Alocação Dinâmica

- Crie uma estrutura que represente um **produto** em estoque, contendo:
  - um número **identificador**
  - uma **quantidade** disponível
  - **preço** atual do produto.

- **Exemplo 11**: crie uma função que **imprime** um produto.

Exemplo Alocação Dinâmica

- **Exemplo 12**: faça um programa que crie um produto “em tempo de execução” com o **new** e, depois de imprimi-lo, remova-o com o **delete**.

Exemplo Alocação Dinâmica

- **Exemplo 13**: Modifique o programa anterior para que ele crie **3** produtos em tempo de execução, imprima os **3** e, posteriormente, remova-os da memória.
- Modifique o programa anterior para que, além de imprimir os conteúdos de cada produto, **também imprima o endereço na memória** de cada produto.

Momento Lúdico:

Entendendo uma Lista Encadeada

Discussão sobre a Dinâmica

Discussão sobre Listas Encadeadas

- Os itens estavam contíguos (lado a lado) ou disperso

Mentimeter

https://www.menti.com/

Discussão sobre Listas Encadeadas

- Dá pra saber até onde esta lista pode crescer?

Mentimeter

https://www.menti.com/

Discussão sobre Listas Encadeadas

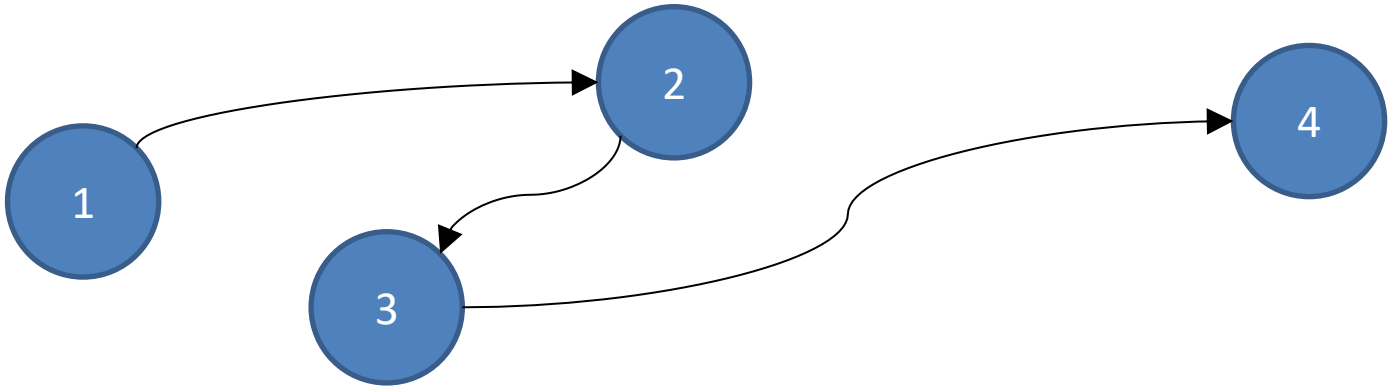
- É simples adicionar outro item na sequência?
  - Ao adicionar outro item, o que haveria nele?
  - Precisa mudar alguma coisa nos outros itens?

Mentimeter

https://www.menti.com/

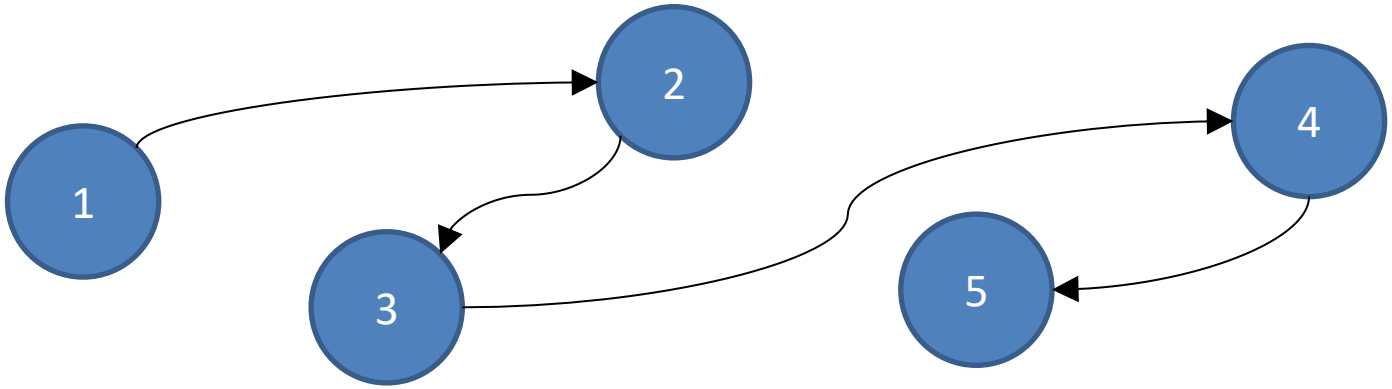
Discussão sobre Listas Encadeadas

- 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)



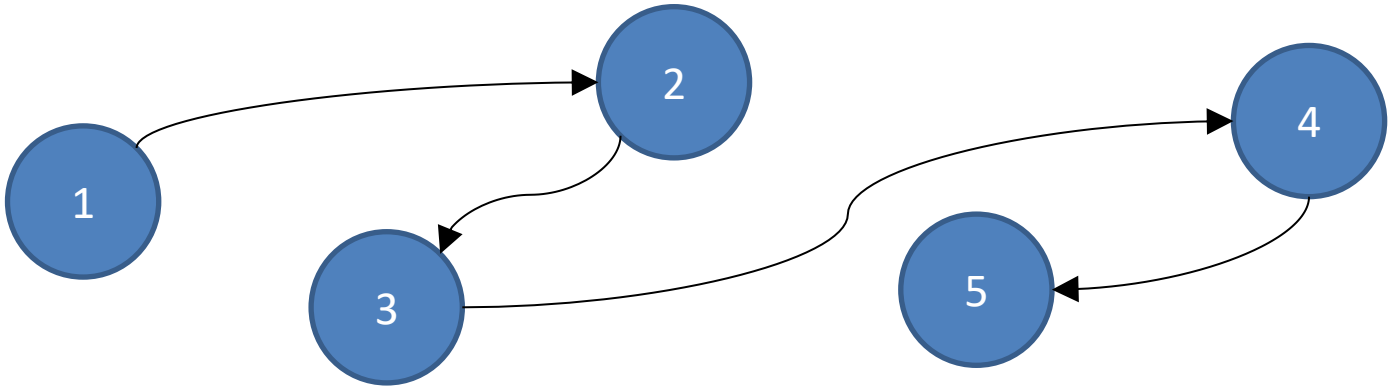
Discussão sobre Listas Encadeadas

Como **criar** novos elementos?

- Li
  in
- Para acres no fim?
- Criar o no

Já vimos isso?

- Associá-lo à lista (encadeá-lo)



Alocação Dinâmica de Memória

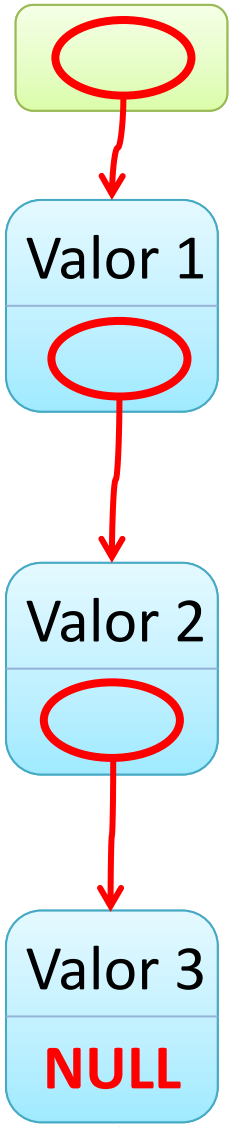
- Alocação dinâmica
  - Reservar no momento necessário
  - Liberar quando não for mais necessário
- Reservar: **new**
- Liberar: **delete**

Representando Listas Encadeadas

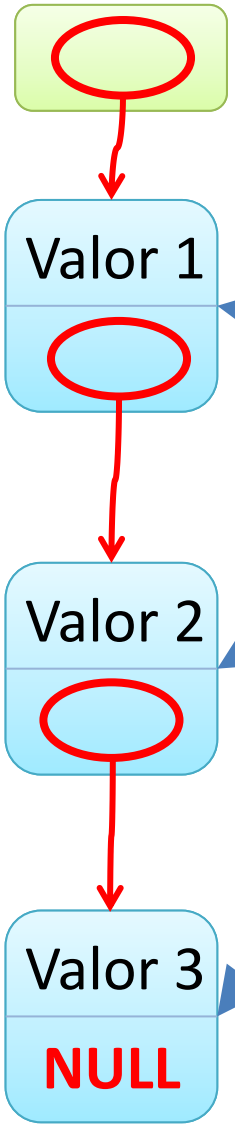
Prof. Dr. Daniel Caetano

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!



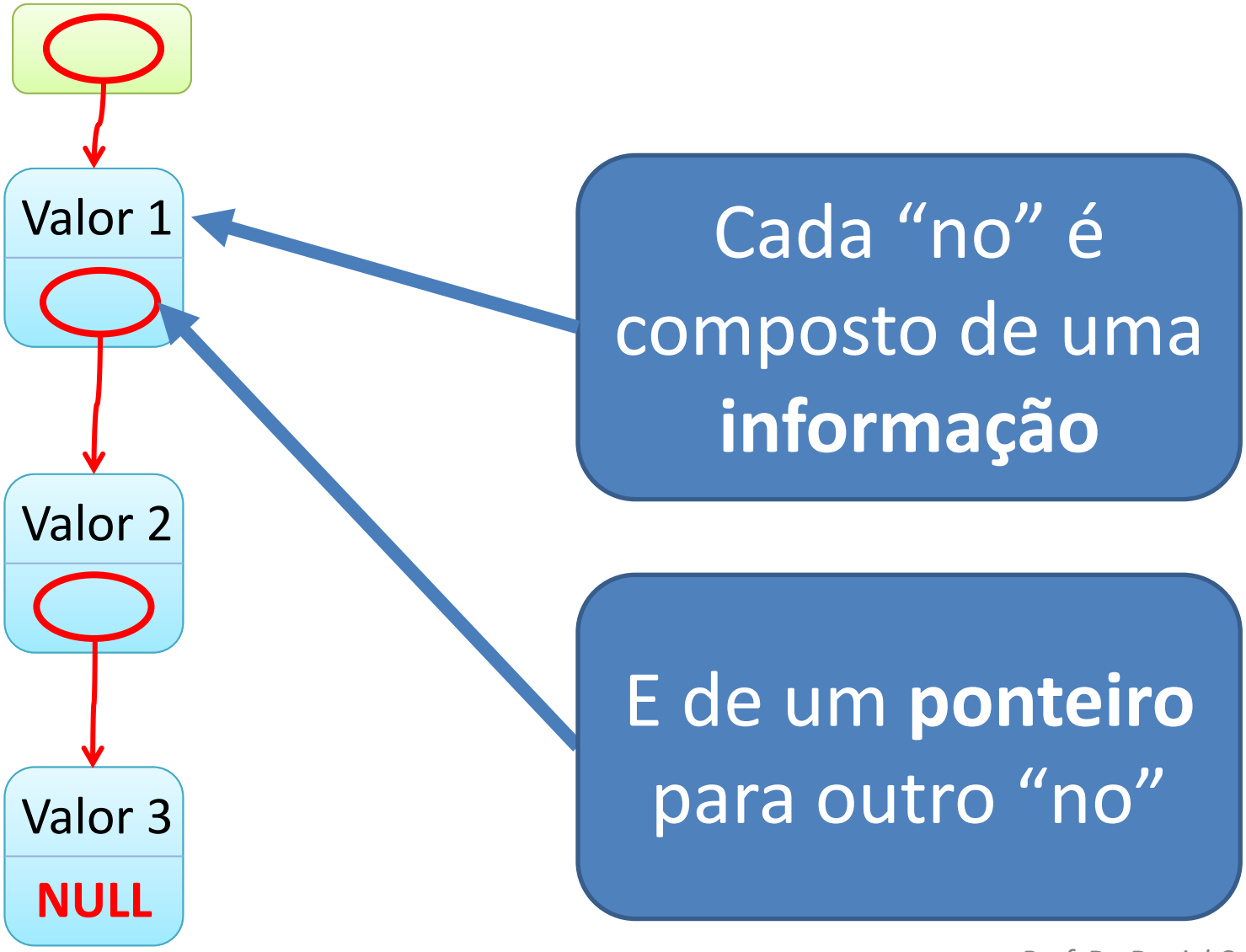
Representando Listas Encadeadas



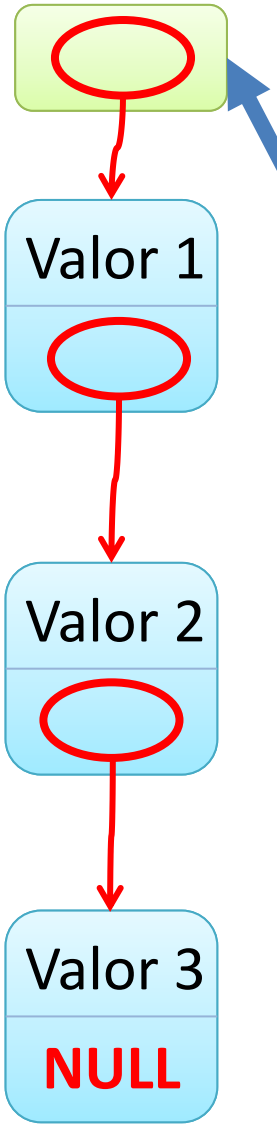
Cada elemento da lista encadeada é uma **struct**

Criaremos uma struct chamada **no**

Representando Listas Encadeadas



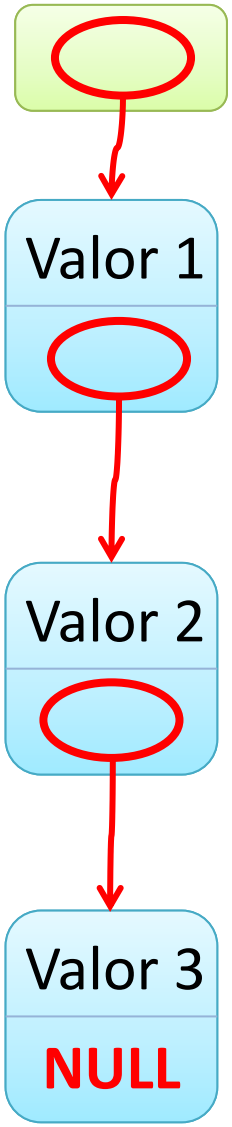
Representando Listas Encadeadas




```
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;
```

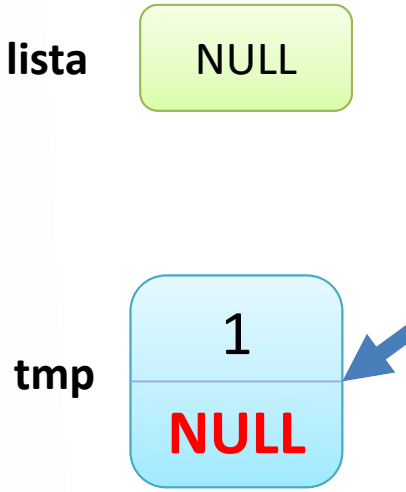
Inicializando Listas Encadeadas e Inserindo nós

Prof. Dr. Daniel Caetano

Listas Encadeadas: Inicializando

no *lista = NULL;

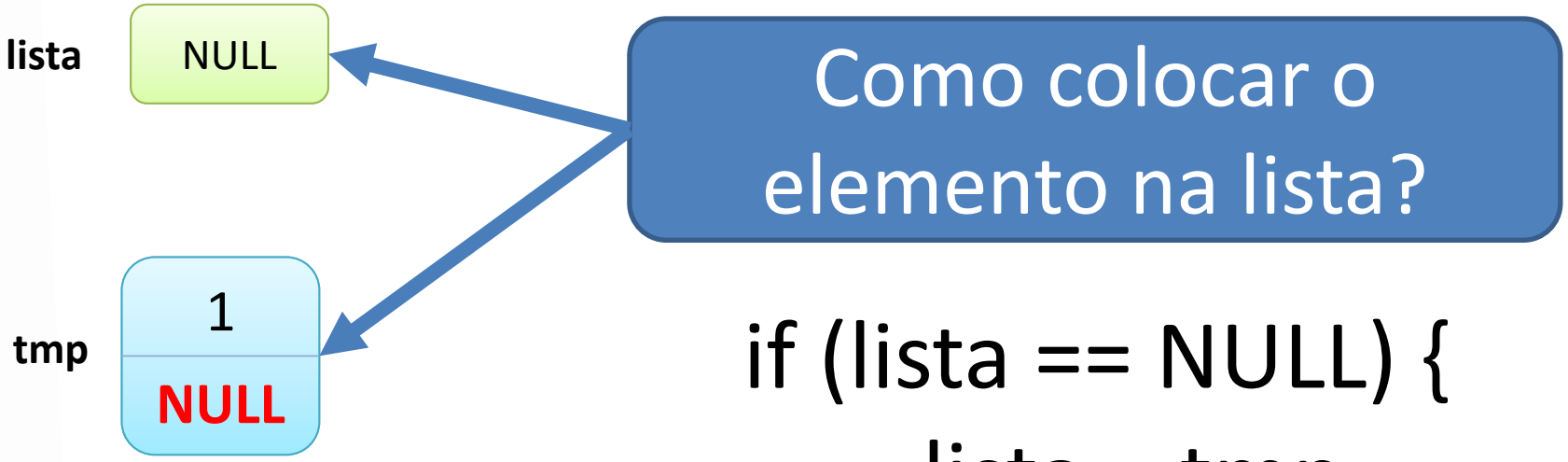
Listas Encadeadas: Inserção



Como criamos um elemento?

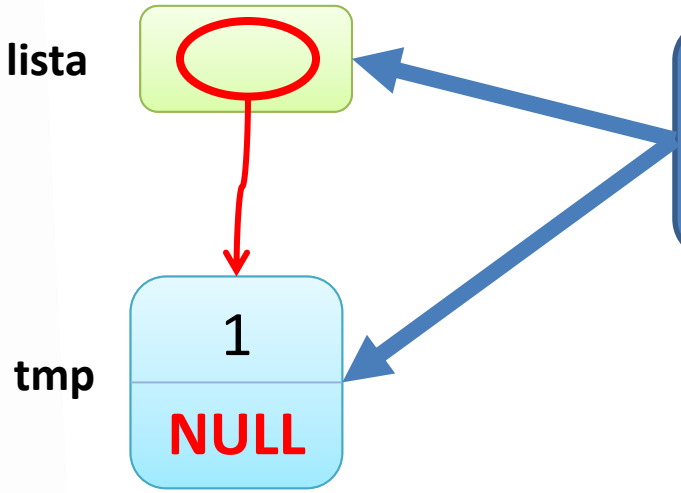
```
no *tmp;
tmp = new no;
tmp->valor = 1;
tmp->ptr = NULL
```

Listas Encadeadas: Inserção




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

Listas Encadeadas: Inserção



Como colocar o elemento na lista?

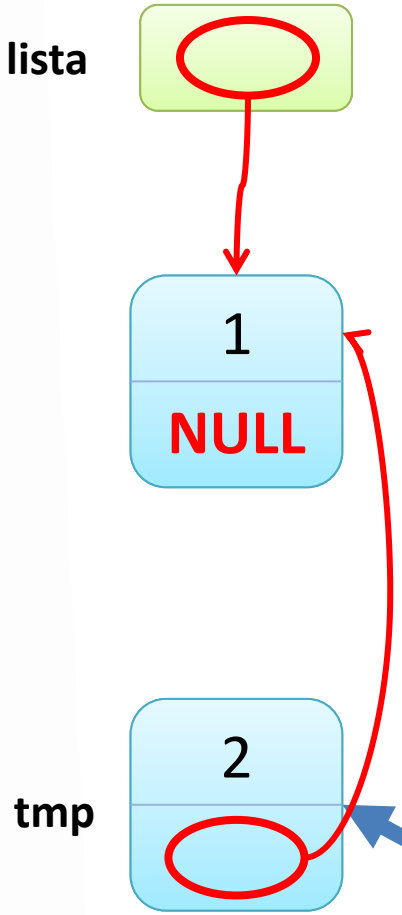
```
if (lista == NULL) {
        lista = tmp;
}
```

Listas Encadeadas: Inserção


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

E para acrescentar um outro elemento?

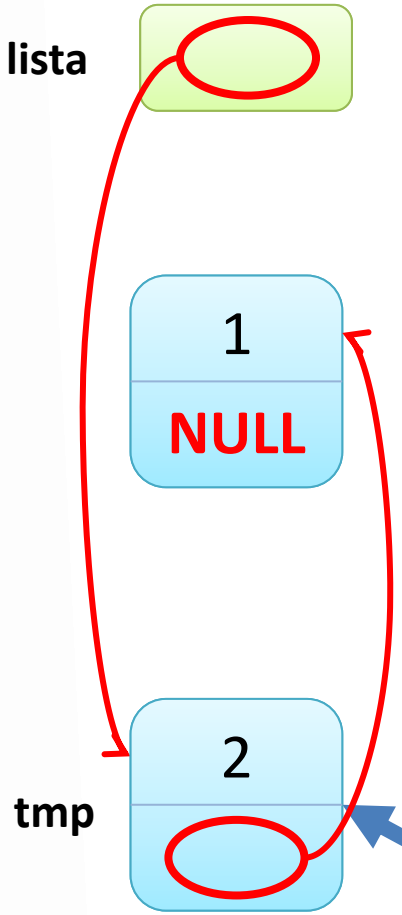
Listas Encadeadas: Inserção




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

E para acrescentar um outro elemento?

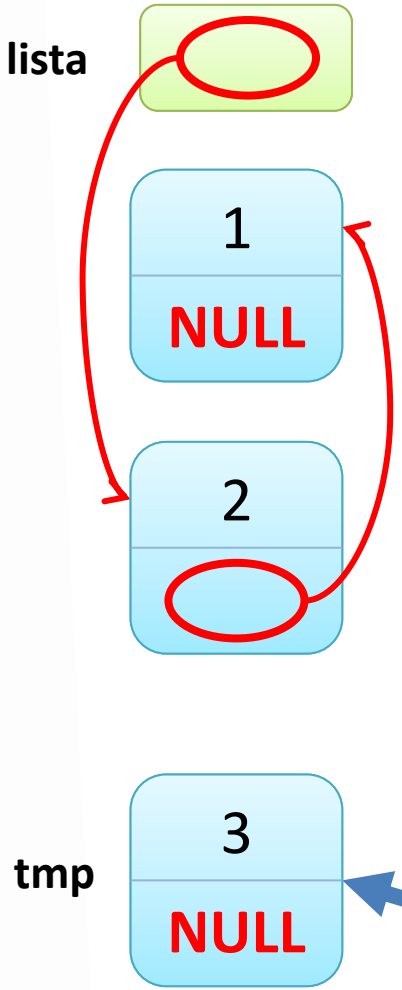
Listas Encadeadas: Inserção




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

E para acrescentar um outro elemento?

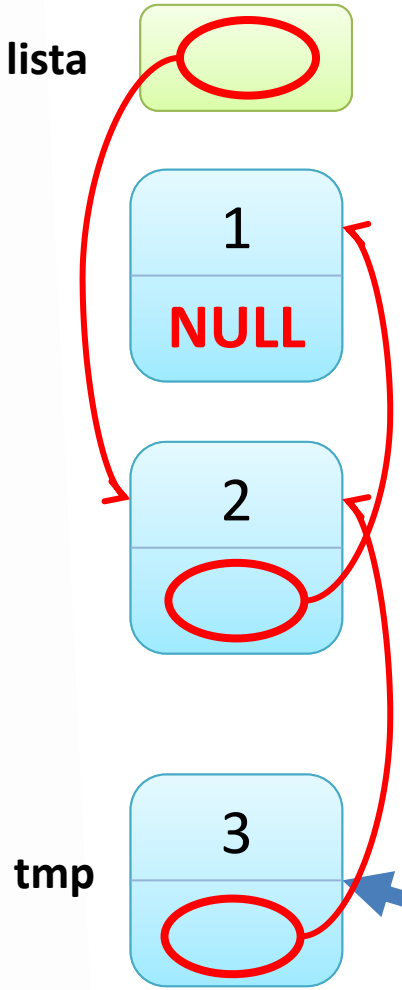
Listas Encadeadas: Inserção




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

E para acrescentar um outro elemento?

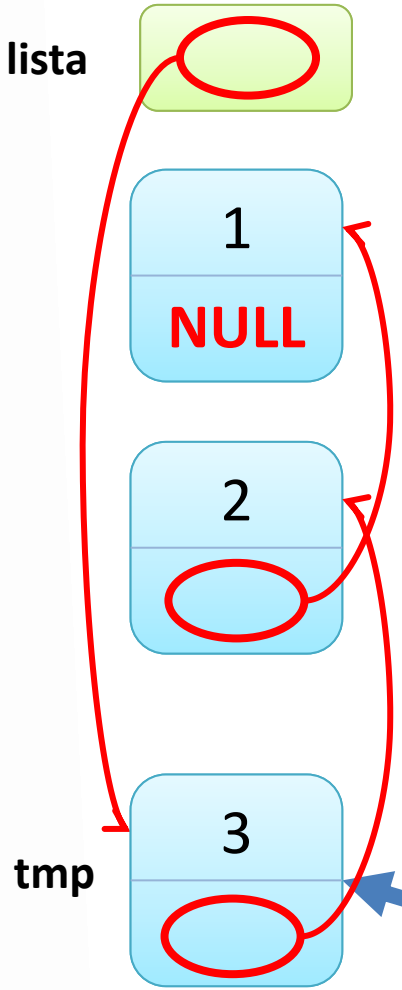
Listas Encadeadas: Inserção




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

E para acrescentar um outro elemento?

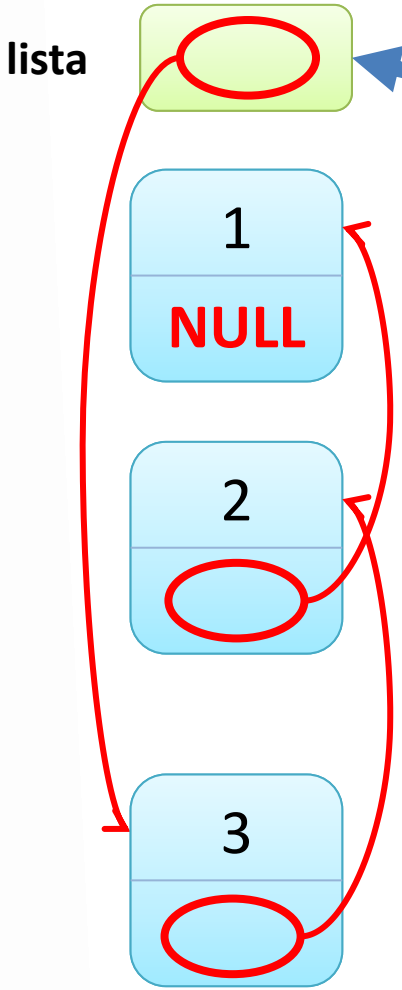
Listas Encadeadas: Inserção




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

E para acrescentar um outro elemento?

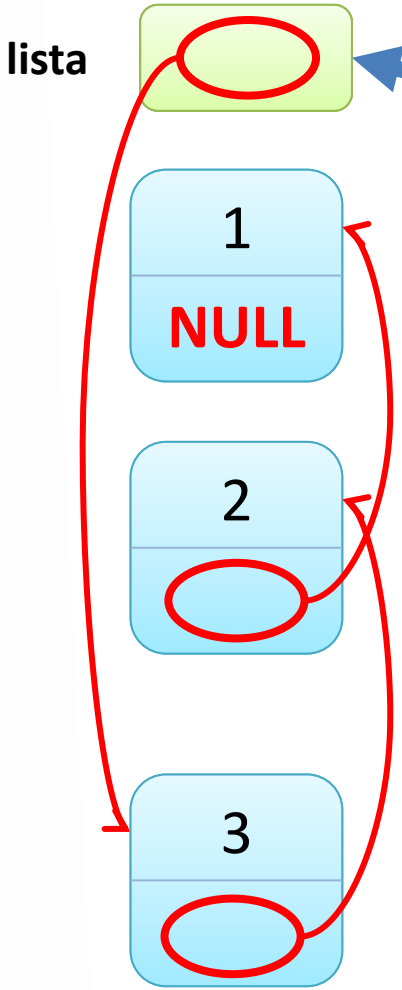
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

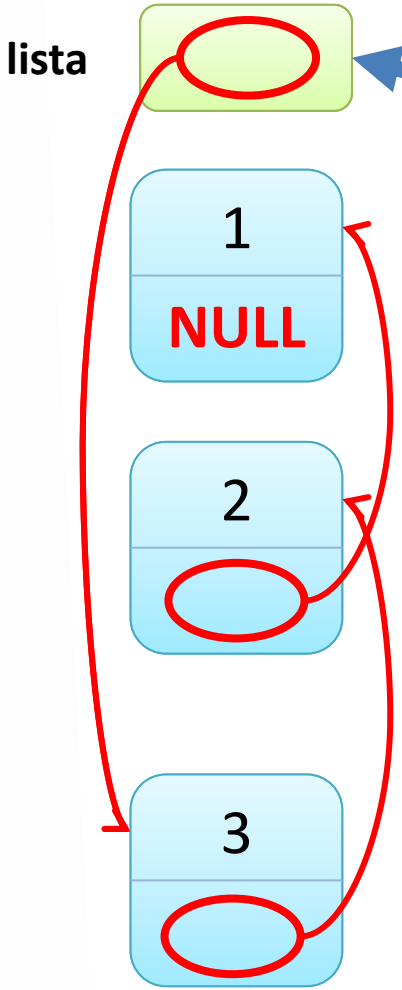
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

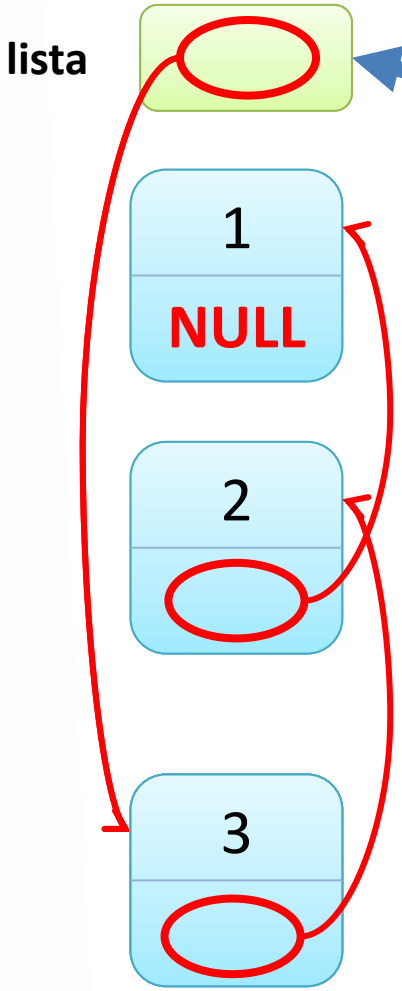
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

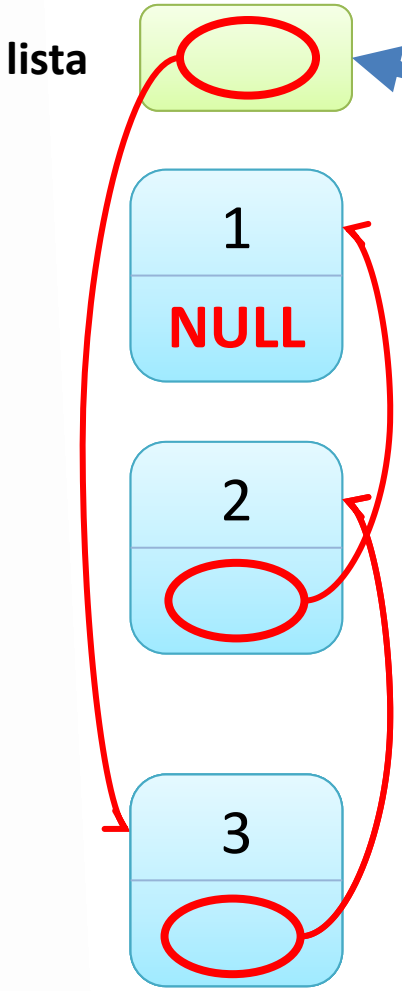
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

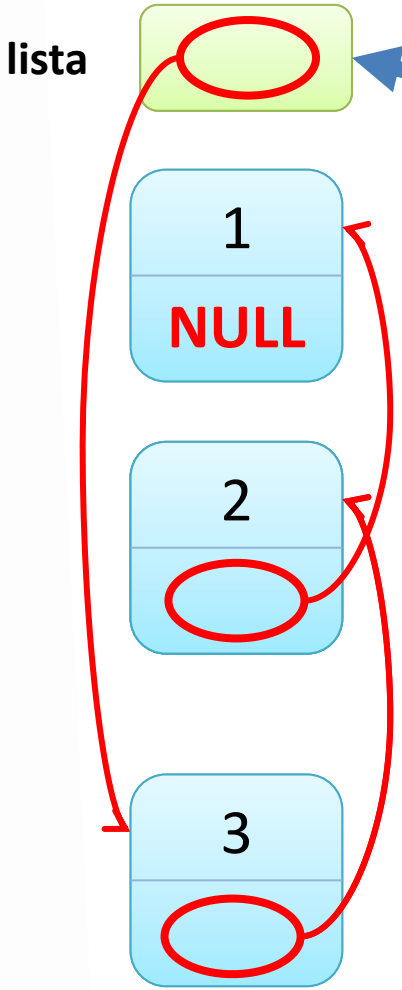
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

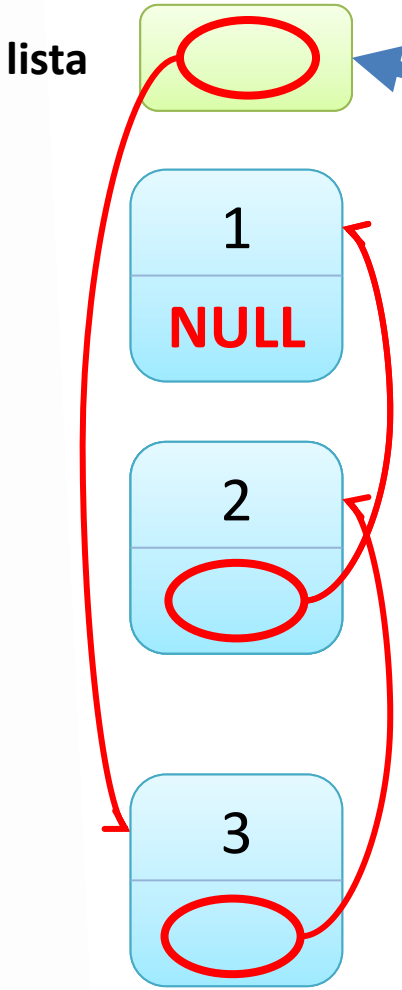
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

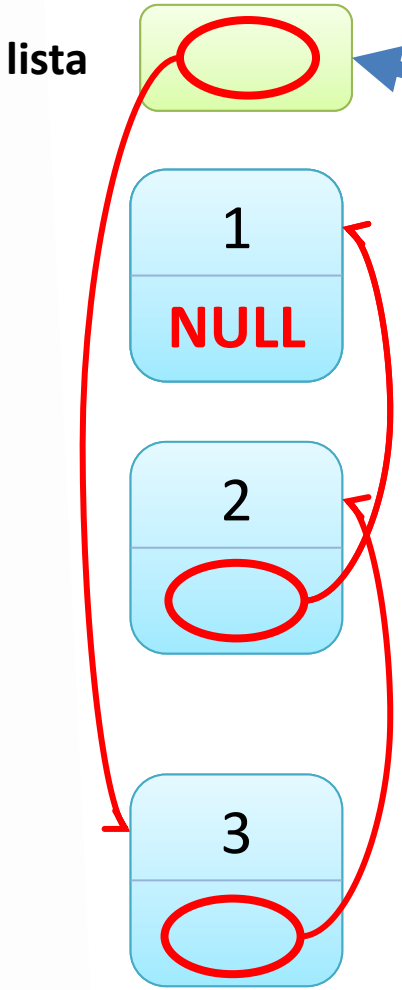
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

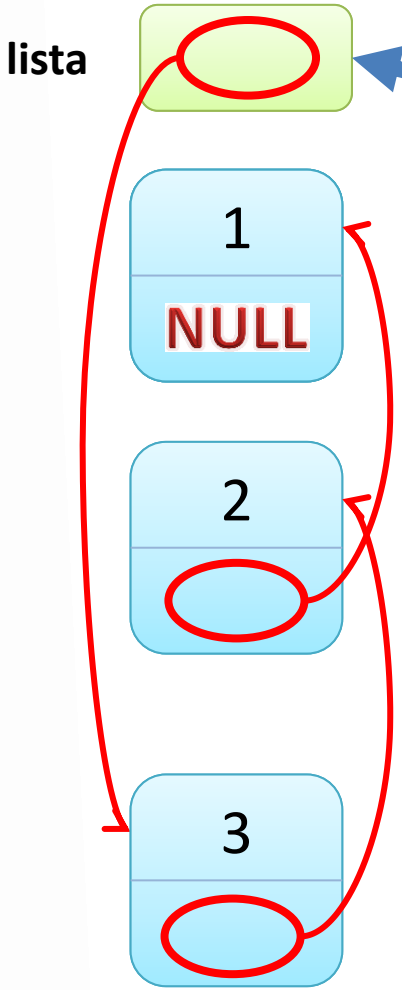
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

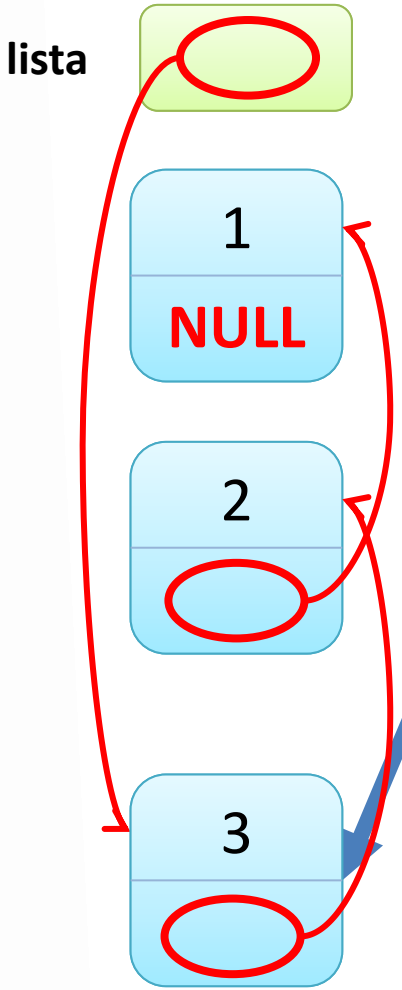
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

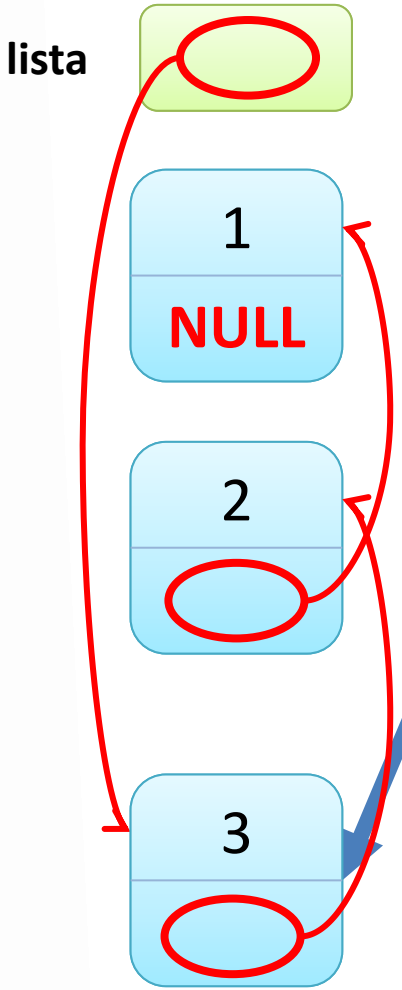
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

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

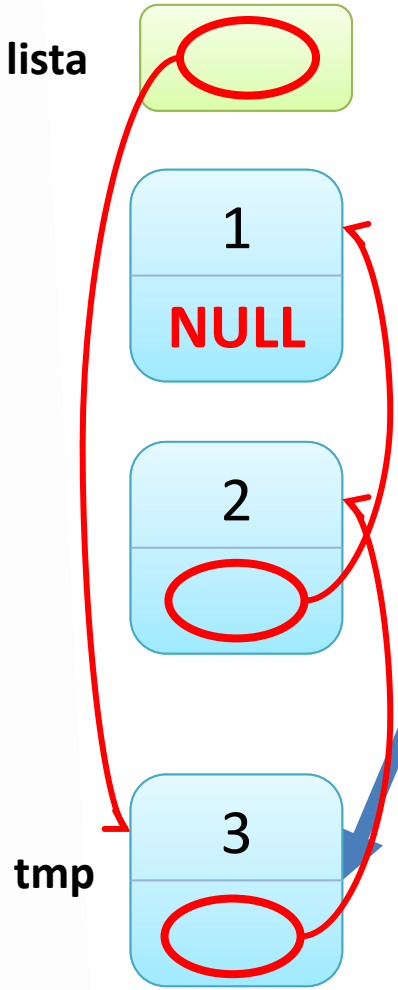
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

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

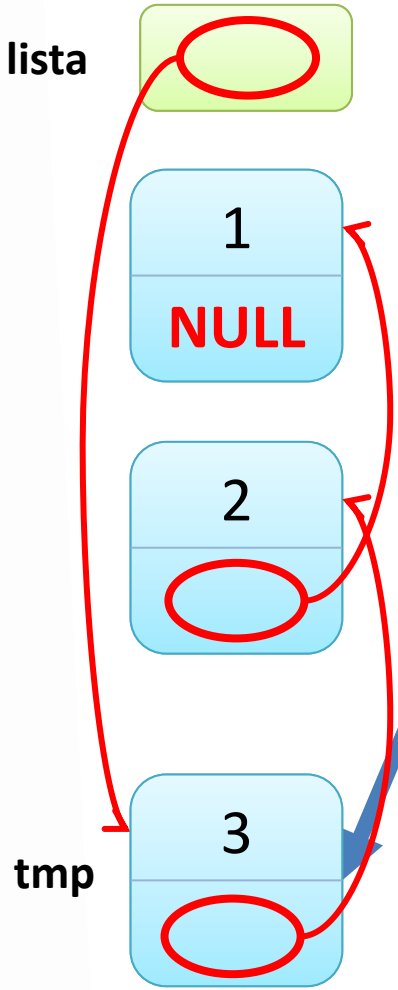
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

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

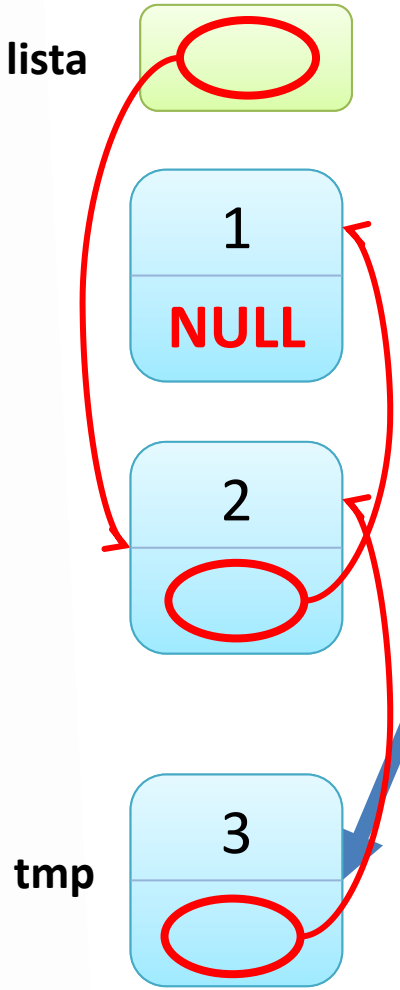
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

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

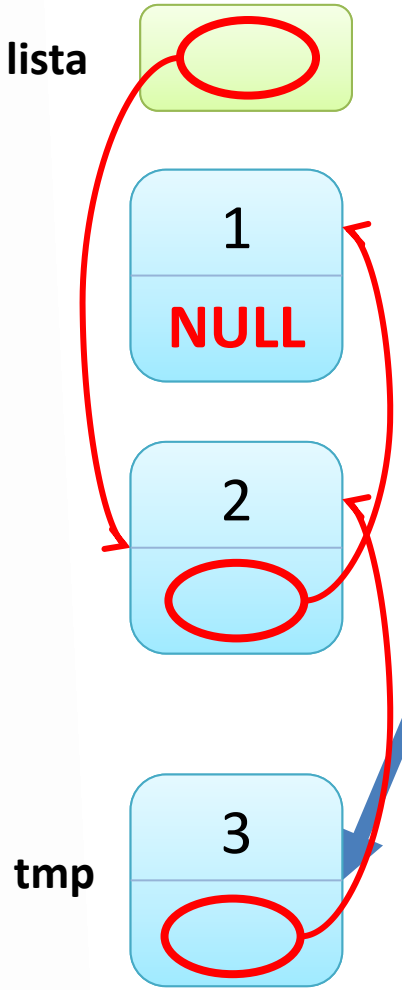
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

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

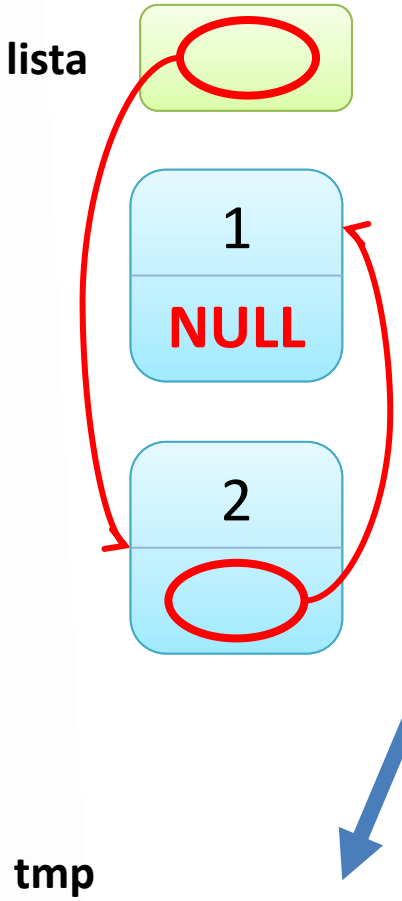
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

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

Listas Encadeadas: Remoção



Como remover o primeiro elemento?

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

Operações Com Listas Encadeadas

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

Na Prática:

Implementando uma Lista Encadeada

Prof. Dr. Daniel Caetano

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

Atividade Avaliativa D

Prof. Dr. Daniel Caetano

ENCERRAMENTO

Resumo e Próximos Passos

- Ponteiros
- Alocação dinâmica de memória
- Listas Dinâmicas
  - E funcionalidades
- **Pós Aula**: Saiba Mais, A Seguir... e Desafio!
  - No mural: https://padlet.com/djcaetano/paradigmas

- Expressões e sentenças de atribuição
  - O que podemos construir?

PERGUNTAS?