Paradigmas de Linguagens de Programação em Python

Tipos de Dados Avançados Parte I e II

Prof. Dr. Daniel Caetano

2020 - 2

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?

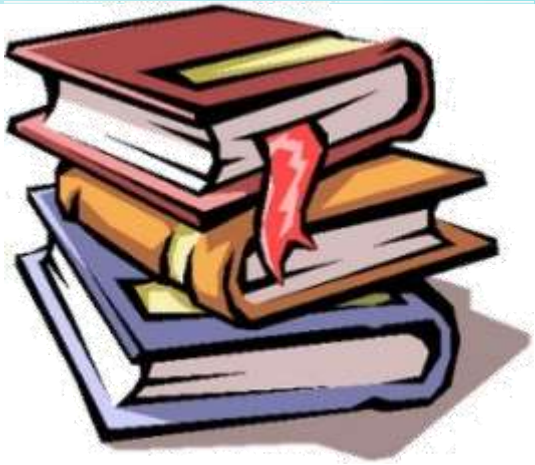
https://www.menti.com/

Objetivos

- Conhecer funções com parâmetros e retorno
- Compreender a lógica das listas estáticas
- Compreender o conceito de variável-ponteiro
- Compreender o uso de alocação de memória
- Compreender diferenças das listas dinâmicas

- **Desafio Aula 08 – Parte I / II**

Bibliografia da Aula



Material	Acesso ao Material
Apresentação	https://www.caetano.eng.br/ (Paradigmas de Programação – Aula 8)
Livro Texto	Capítulo 6, páginas 273 a 287
Aprenda Mais!	• Vídeo: Listas Encadeadas (ative legenda traduzida!) https://youtu.be/njTh_OwMljA

Parte I
Funções e Listas Estáticas

FUNÇÕES

https://www.menti.com/

Subprogramas em Python

- Já vimos antes...

```
def assina():
    print("Atenciosamente,")
    print("Prof. Daniel Caetano")
    print("prof@caetano.eng.br")

assina()
```

- Será que em C/C++ é tão simples?
  - Quase a mesma coisa...

Subprogramas em C/C++

Exemplo 01

```
#include <iostream>
using namespace std;

void assina() {
    cout << "Atenciosamente," << endl;
    cout << "Prof. Daniel Caetano" << endl;
    cout << "prof@caetano.eng.br" << endl;
}

int main() {
    assina();
}
```

Por que chamamos alguns subprogramas de funções?

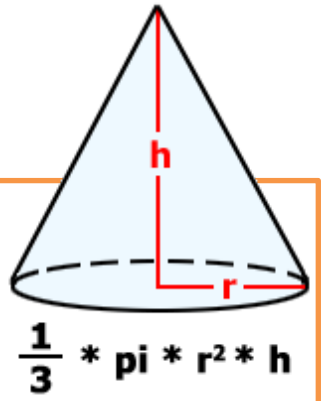
Momento Lúdico

- Como fazer um misto quente?

- Como fazer um sanduiche com um “recheio” genérico?

Funções: subprogramas + parâmetros

- Calcular o volume de um cone

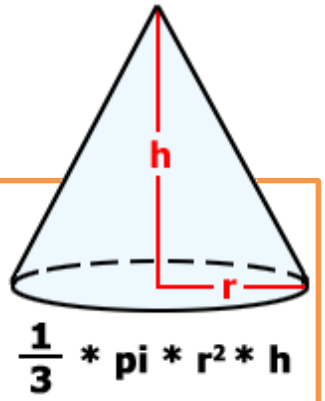


Exemplo 02

```
#include <iostream>
using namespace std;

void volumeCone(float r, float h) {
    float res = (1.0/3.0)*3.14*r*r*h;
    cout << res;
}

main() {
    volumeCone(10,2);
}
```

Funções: subprogramas + parâmetros

- Calcular o volume de um cone



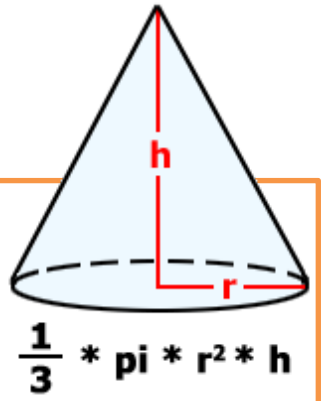
Exemplo 02

```
#include <iostream>
using namespace std;

void volumeCone(float r, float h) {
    float res = (1.0/3.0)*3.14*r*r*h;
    cout << res;
}

main() {
    volumeCone(10,2);
}
```

Mude esses valores e teste!

Funções: subs + parâms + retorno

- Calcular o volume de um cone

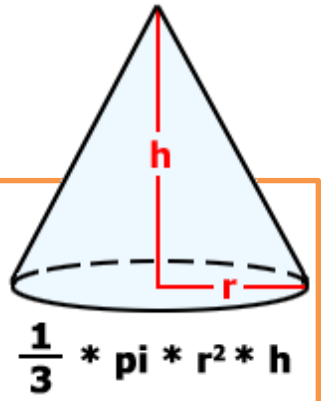


Exemplo 02b

```
#include <iostream>
using namespace std;

void volumeCone(float r, float h) {
    float res = (1.0/3.0)*3.14*r*r*h;
    return res;
}

main() {
    cout << volumeCone(10,2);
}
```

Funções: subs + parâms + retorno

- Calcular o volume de um cone



Exemplo 02b

```
#include <iostream>
using namespace std;

void volumeCone(float r, float h) {
    float res = (1.0/3.0)*3.14*r*r*h;
    return res;
}

main() {
    cout << volumeCone(10,2);
}
```

Flexibilizar
Organizar
Reaproveitar

Funções: Modificando Parâmetros

- Função que atualiza salário (+10%)

Exemplo 03

```
#include <iostream>
using namespace std;

void atualizaSalario(float sal) {
  sal = sal * 1.10;
}

int main() {
   float s=100.00 ;
   atualizaSalario(s);
   cout << s;
}
```

Funções: Modificando Parâmetros

- Função que atualiza salário (+10%)

Exemplo 03

```
#include <iostream>
using namespace std;

void atualizaSalario(float sal) {
  sal = sal * 1.10;
}

int main() {
   float s=100.00 ;
   atualizaSalario(s);
   cout << s;
}
```

Não funciona!

Funções: Modificando Parâmetros

- Função que atualiza salário (+10%)

Exemplo 03

```
#include <iostream>
using namespace std;

void atualizaSalario(float &sal) {
  sal = sal * 1.10;
}

int main() {
   float s=100.00 ;
   atualizaSalario(s);
   cout << s;
}
```

Passagem por **referência**

E se colocar um **número** aqui?

FUNÇÕES: ATIVIDADES

Atividade 01

Crie (e use) uma função que imprima o texto

Eu estou funcionando!

https://www.tutorialspoint.com/compile_cpp_online.php

Atividade 02

Crie (e use) uma função que some dois valores inteiros e retorne o resultado.

Atividade 03

Crie uma função em C/C++ que calcule a área de um retângulo e tenha a seguinte estrutura:

double calculaArea(double base, double altura)

Depois construa a **main** que usa essa função

Listas Lineares Estáticas

https://www.menti.com/

Listas Lineares Estáticas

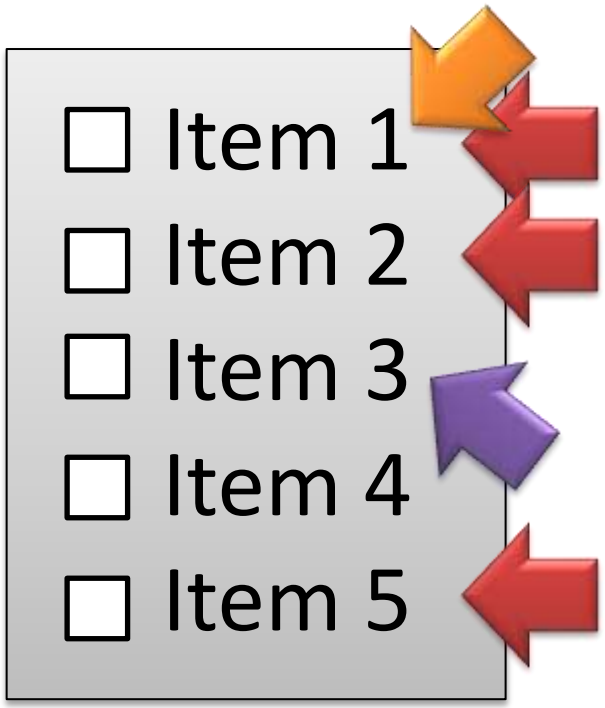
- Como representar a bibliografia do curso?
- Como representar os contatos telefônicos?
- Como representar o conjunto de notas dos alunos?

Lista

Só que esse não é tipo primitivo do C/C++!

Entendendo Detalhes da Lista

- Uma lista é um conjunto de elementos
  - Usualmente de um mesmo tipo
  - Possui uma ordem
    - Primeiro elemento
    - Último elemento
  - Elementos Intermediários
    - Antecessor
    - Sucessor



Listas: Detalhes de Implementação

- Se sabemos o tamanho máximo da lista...
  - Podemos alocar todo o espaço...
  - Espaço “contíguo” na memória
  - Podemos usar um vetor!
- Exemplo: armazenar **até** 10 notas de alunos

float notas[10];

Listas: Detalhes de Implementação

- Por que podemos usar vetor?
  - Tamanho máximo da lista: **10**
  - Dados todos do mesmo tipo: **float**
- A lista vai estar sempre cheia?
  - Se houver só 7 notas, quantas imprimir?
  - Mas como vamos saber que são 7?
  - Variável de controle de *quantidade*

float notas[10];

int quantidade;

Listas: Detalhes de Implementação

- Note: estrutura composta por **duas** variáveis

```
float notas[10];
int quantidade;
```

Listas: Detalhes de Implementação

- Vamos começar um programa e declarar nossa lista dentro do main

float notas[40];

int quantidade;

nota:

0	1	2	3	4	5	6	7	8	9
?	?	?	?	?	?	?	?	?	?

quantidade: ???

Listas: Detalhes de Implementação

- Que operações podemos implementar?
  - *Inicializar*
  - *Inserir*
  - *Listar*
  - *Buscar*
  - *Remover*
  - *Substituir*
  - *Ordenar*
  - *...*

Listas: Detalhes de Implementação

- Que operações podemos implementar?
  - ***Inicializar***
  - ***Inserir***
  - ***Listar***
  - ***Buscar***
  - ***Remover***
  - *Substituir*
  - *Ordenar*
  - *...*

Listas: Lógica de Criação

- Inicializar?
  - Definir o “status” inicial
  - Prepará-la para o uso
  - O que caracteriza uma lista que não recebeu dados?
- Vamos implementar a inicialização?

nota:

0	1	2	3	4	5	6	7	8	9
?	?	?	?	?	?	?	?	?	?

quantidade: 0

Exemplo 04

Listas: Lógica de Inserção

- Inserir
  - Vamos considerar uma lista sem ordenação
  - Vamos considerar que pode haver repetição
  - Acrescentar valor no fim do vetor
  - Verificar se a lista está cheia!

nota:

0	1	2	3	4	5	6	7	8	9
?	?	?	?	?	?	?	?	?	?

quantidade: 0

  - Depois de inserir dado, incrementar *quantidade*

Listas: Lógica de Inserção

- Inserir
  - Vamos implementar a função inserir?
  - Parâmetros
    - Vetor
    - Dado a inserir
    - Posição (= quantidade)
    - Tamanho máximo da lista

por referência!

- Vamos ler 1 valor na **main** e inseri-lo na lista

Exemplo 05a

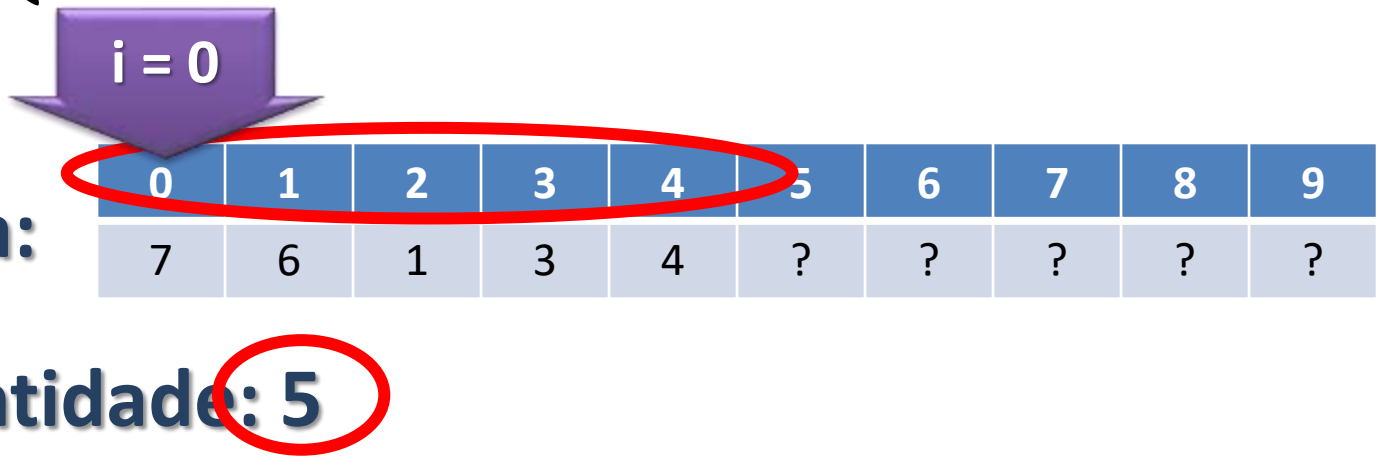
Exemplo de Uso

- Implementar, na main, a leitura de vários valores, até o usuário digitar o valor 0
  - Cada valor lido deve ser inserido na lista
  - Se o valor for zero, ele não deve ser inserido

Exemplo 05b

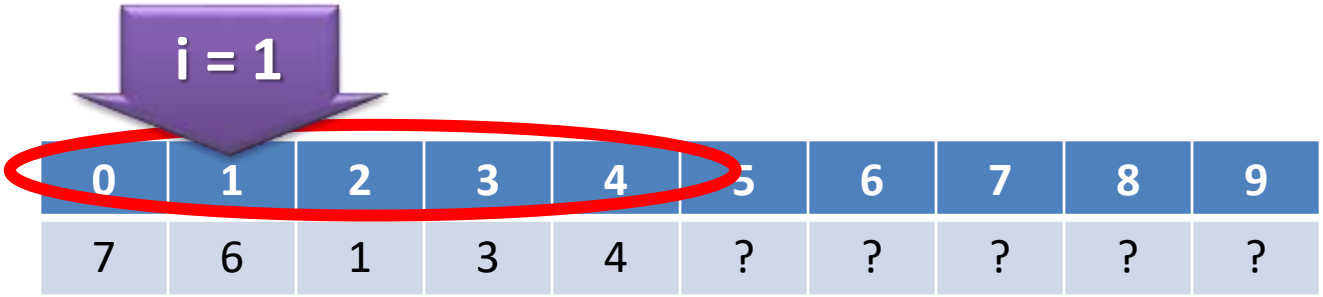
Listas: Lógica de Varredura

- Listar?
  - Imprimir um a um os valores...
  - Até o fim do vetor?
  - Quantidade!



Listas: Lógica de Varredura

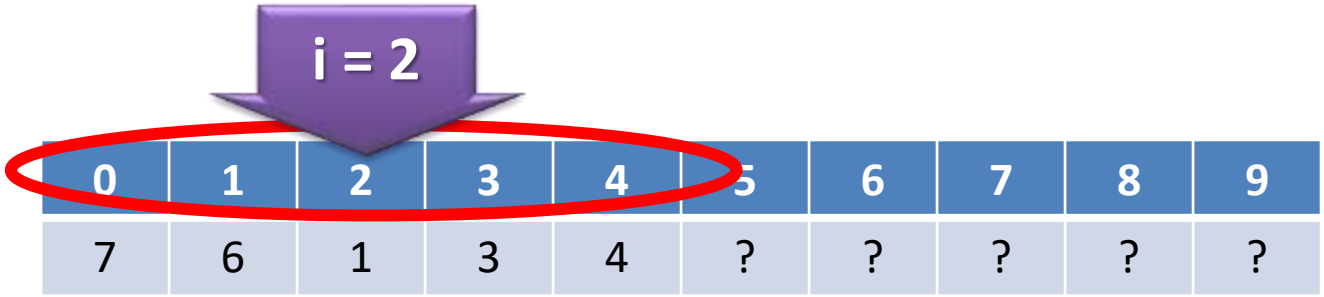
- Listar?
  - Imprimir um a um os valores...
  - Até o fim do vetor?
  - Quantidade!



quantidade: 5

Listas: Lógica de Varredura

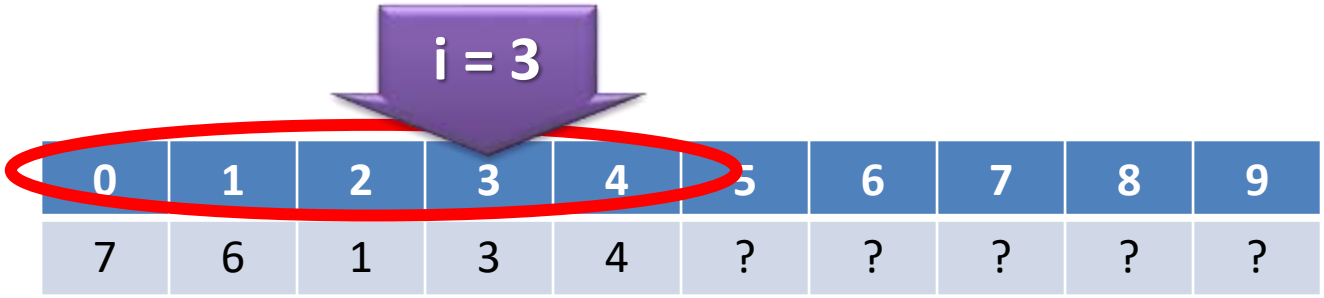
- Listar?
  - Imprimir um a um os valores...
  - Até o fim do vetor?
  - Quantidade!



quantidade: 5

Listas: Lógica de Varredura

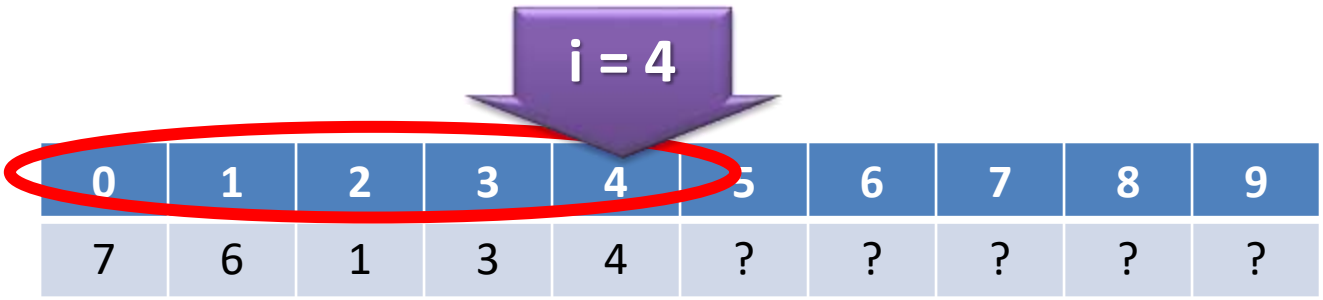
- Listar?
  - Imprimir um a um os valores...
  - Até o fim do vetor?
  - Quantidade!



quantidade: 5

Listas: Lógica de Varredura

- Listar?
  - Imprimir um a um os valores...
  - Até o fim do vetor?
  - Quantidade!



quantidade: 5

  - ***i = 0*** enquanto ***i < quantidade***

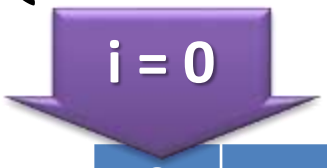
Exemplo de Aplicação

- Listar
  - Vamos implementar a função listar?
  - Parâmetros
    - Vetor
    - Quantidade
- Vamos imprimir o vetor na **main**?

Exemplo 06

Listas: Lógica de Busca

- Buscar?
  - Procurar por um valor
  - Até o fim do vetor?
  - Quantidade!



nota:

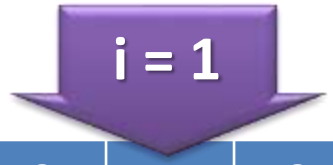
0	1	2	3	4	5	6	7	8	9
7	6	1	3	4	?	?	?	?	?

quantidade: 5

valor: 1

Listas: Lógica de Busca

- Buscar?
  - Procurar por um valor
  - Até o fim do vetor?
  - Quantidade!



nota:

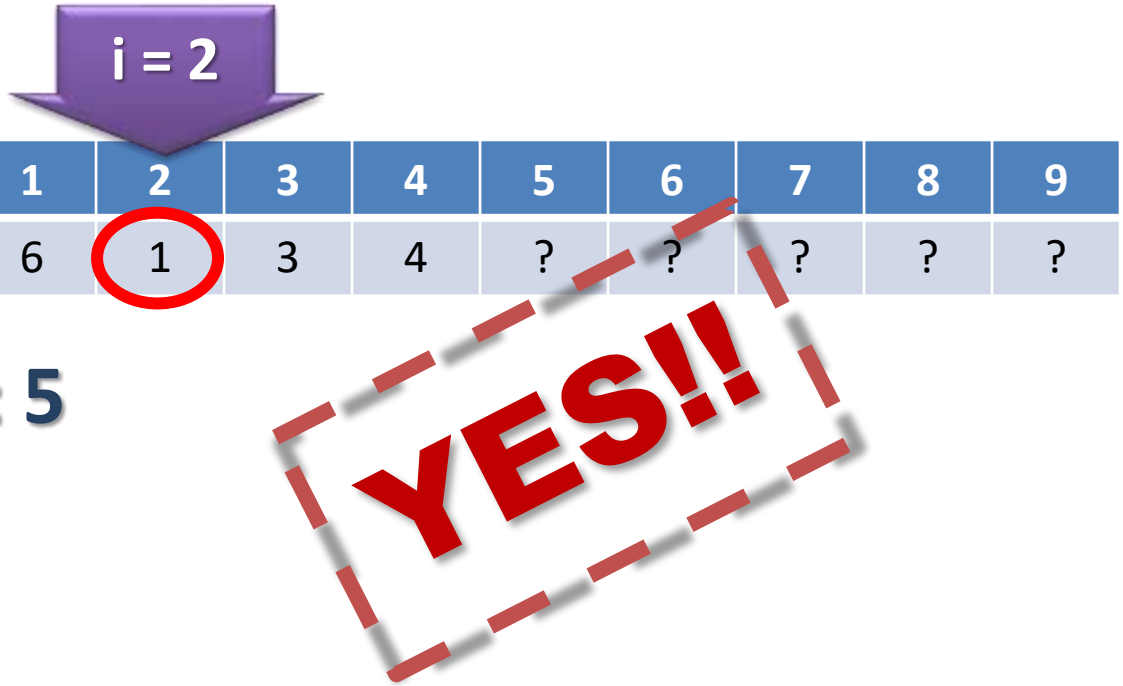
0	1	2	3	4	5	6	7	8	9
7	6	1	3	4	?	?	?	?	?

quantidade: 5

valor: 1

Listas: Lógica de Busca

- Buscar?
  - Procurar por um valor
  - Até o fim do vetor?
  - Quantidade!



Listas: Lógica de Busca

- Buscar
  - Mas e se a busca não encontrar o número?
  - Responder **-1**
  - **Por quê?**
  - Parâmetros
    - Vetor
    - Quantidade
    - Valor

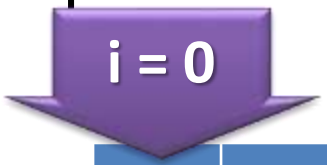
Exemplo de Aplicação

- Vamos implementar a função **buscar**!
- Modificar a **main** para permitir uma busca
  - Se encontrar, deve imprimir a posição
  - Se não encontrar, deve imprimir que não achou

Exemplo 07

Listas: Lógica de Remoção

- Remover?
  - Remover um dado valor
  - Procurar por ele...
  - Copiar o último elemento sobre ele



nota:

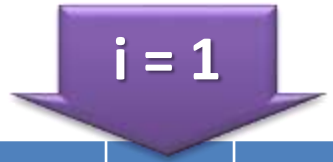
0	1	2	3	4	5	6	7	8	9
7	6	1	3	4	?	?	?	?	?

quantidade: 5

valor: 1

Listas: Lógica de Remoção

- Remover
  - Remover um dado valor
  - Procurar por ele...
  - Copiar o último elemento sobre ele



nota:

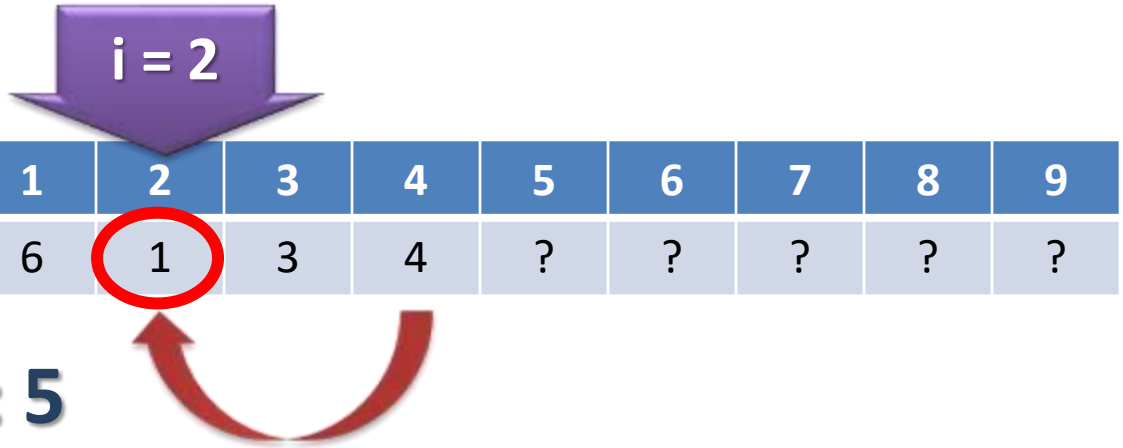
0	1	2	3	4	5	6	7	8	9
7	6	1	3	4	?	?	?	?	?

quantidade: 5

valor: 1

Listas: Lógica de Remoção

- Remover
  - Remover um dado valor
  - Procurar por ele...
  - Copiar o último elemento sobre ele

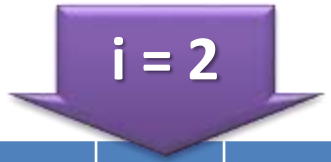


quantidade: 5

valor: 1

Listas: Lógica de Remoção

- Remover
  - Remover um dado valor
  - Procurar por ele...
  - Copiar o último elemento sobre ele



nota:

0	1	2	3	4	5	6	7	8	9
7	6	4	3	4	?	?	?	?	?

quantidade: 4

valor: 1

Listas: Lógica de Remoção

- Remover
  - Deve imprimir erro se
    - Lista vazia
    - Elemento não encontrado
  - Parâmetros
    - Vetor
    - Quantidade **por referência!**
    - Valor

Exemplo de Aplicação

- Vamos implementar a função **remover**!
- Modifique a **main** para permitir que um elemento seja removido
  - Imprimir a lista após remoção

Exemplo 08

Listas Estáticas: Atividades

Atividade 04 – 10 minutos

- Faça um programa para receber uma lista de até 50 **inteiros** e depois imprimir a lista dos números digitados *(use como base o código fornecido pelo professor!)*
  - A inserção deve parar se um número negativo for digitado
  - O número negativo não deve ser inserido na lista
  - Imprima a lista ao final.

Atividade 05 – 10 minutos

- Altere o programa anterior para que ele receba uma lista de até 50 inteiros **distintos** e depois imprimir a lista dos números digitados
- Para esse programa, crie uma função

void inserirSemRepetir(int v[], int valor, int &pos, int max)

Atividade 06 – 5 minutos

- Modifique o programa anterior para que ele guarde até 50 números de telefone (long) distintos.
  - A inserção deve parar se um número com menos de 8 dígitos (menor que 10000000) for digitado
  - Números com menos de 8 dígitos não devem ser guardados

Voltando ao Problema Original: Exemplo

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?

Exemplo 09

- Modifique o programa para que ele trabalhe com uma lista de Alunos, descritos pela seguinte estrutura:

Aluno

- Matrícula (int)
- Média (float)

Parte II
Dos Ponteiros às Listas Dinâmicas

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 10**

```
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 11

Recordando: Endereços

- Podemos guardar endereço em uma variável?

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

- Funcionou?

Exemplo 11

- Precisamos de um tipo novo de variável...

Ponteiros

Ponteiros

- **Exemplo 13**: 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

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

Exemplo 12

```
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

- Acompanhe o **exemplo 14**:
  - Conhecendo o básico sobre ponteiros.

Ponteiros

- Acompanhe o **exemplo 15**:
  - Mudando valores usando ponteiros.

Ponteiros

- Acompanhe o **exemplo 16**:
  - Ponteiros nulos: NULL.

Ponteiros

- Acompanhe o **exemplo 17**:
  - Ponteiros para estruturas.

Ponteiros

- Acompanhe o **exemplo 18**:
  - Ponteiros para estruturas e o referenciador **->** .

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;
```

Exemplo 19

- 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 20**: crie uma função que **imprime** um produto.

Exemplo Alocação Dinâmica

- **Exemplo 21**: 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

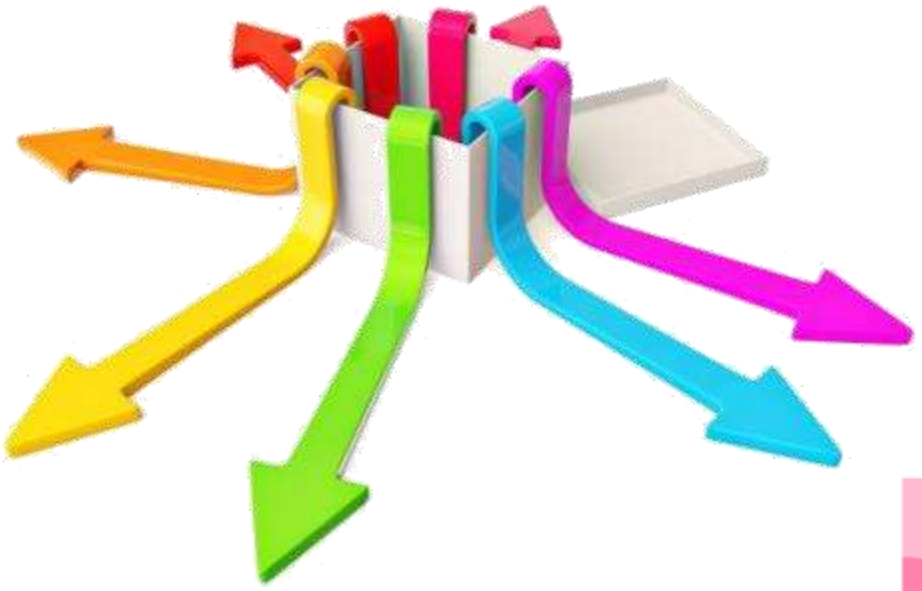
- **Exemplo 22**: Modifique o programa anterior para que ele crie **3** produtos em tempo de execução, imprima os **3** e, posteriormente, remova-os da memória.
- Modifique o programa anterior para que, além de imprimir os conteúdos de cada produto, **também imprima o endereço na memória** de cada produto.

Momento Lúdico: Entendendo uma Lista Encadeada

Discussão sobre a Dinâmica

Discussão sobre Listas Encadeadas

- Os itens estavam contíguos (lado a lado) ou dispersos?



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

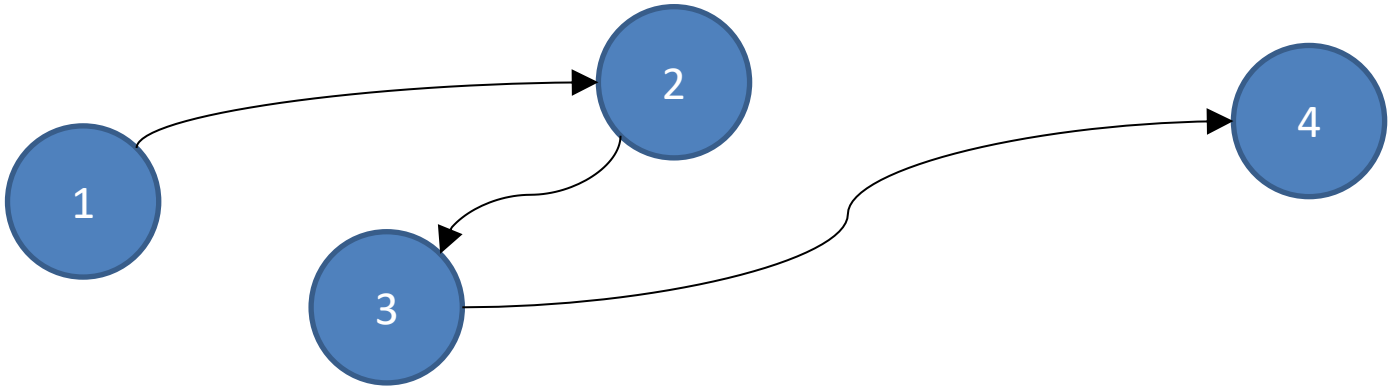
- É simples adicionar outro item na sequência?
  - Ao adicionar outro item, o que haveria nele?
  - Precisa mudar alguma coisa nos outros itens?

Mentimeter

https://www.menti.com/

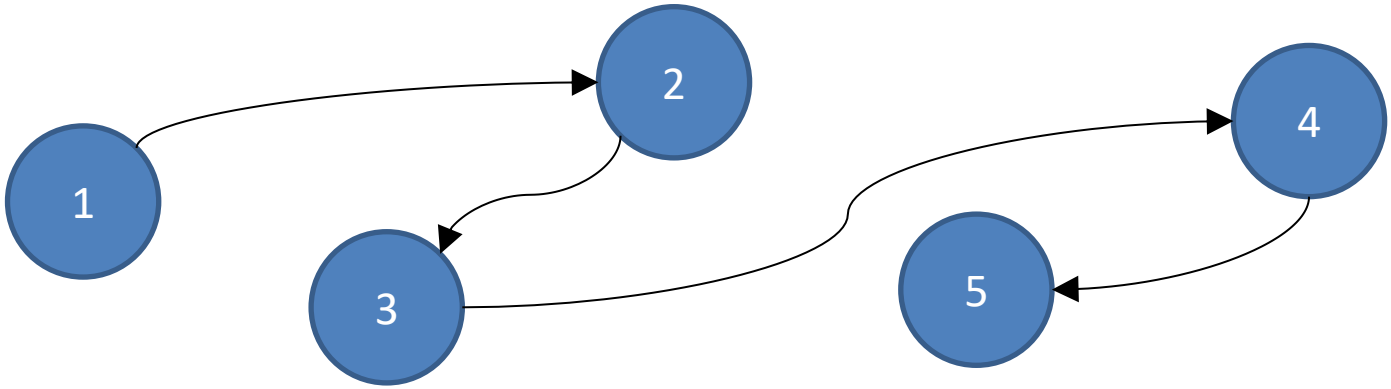
Discussão sobre Listas Encadeadas

- 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)

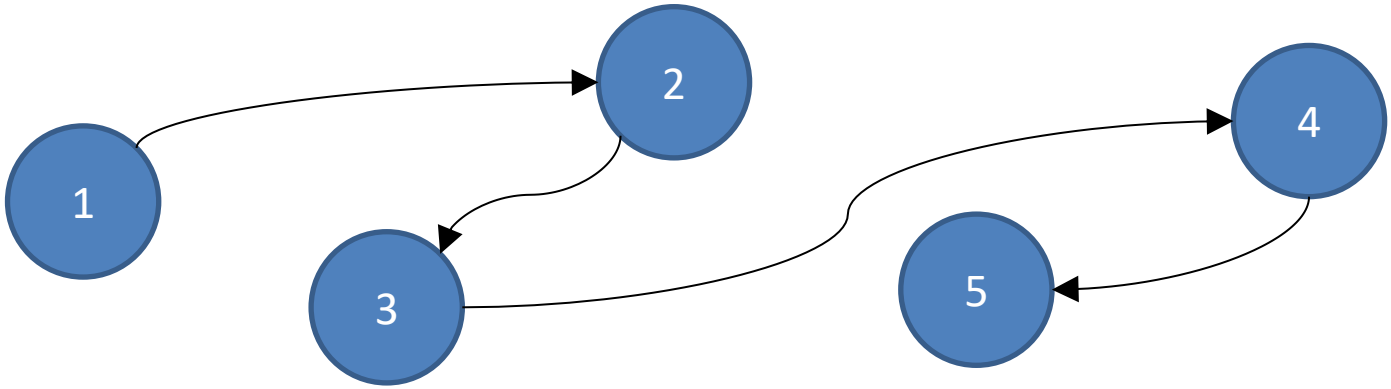


Discussão sobre Listas Encadeadas

- Li
  in
- Para acres no fim?
- Criar o no
- Associá-lo a lista (encadeá-lo)

Como **criar** novos elementos?

Já vimos isso?



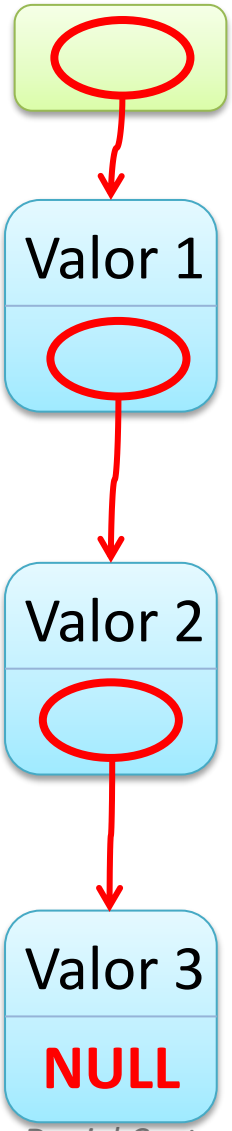
Alocação Dinâmica de Memória

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

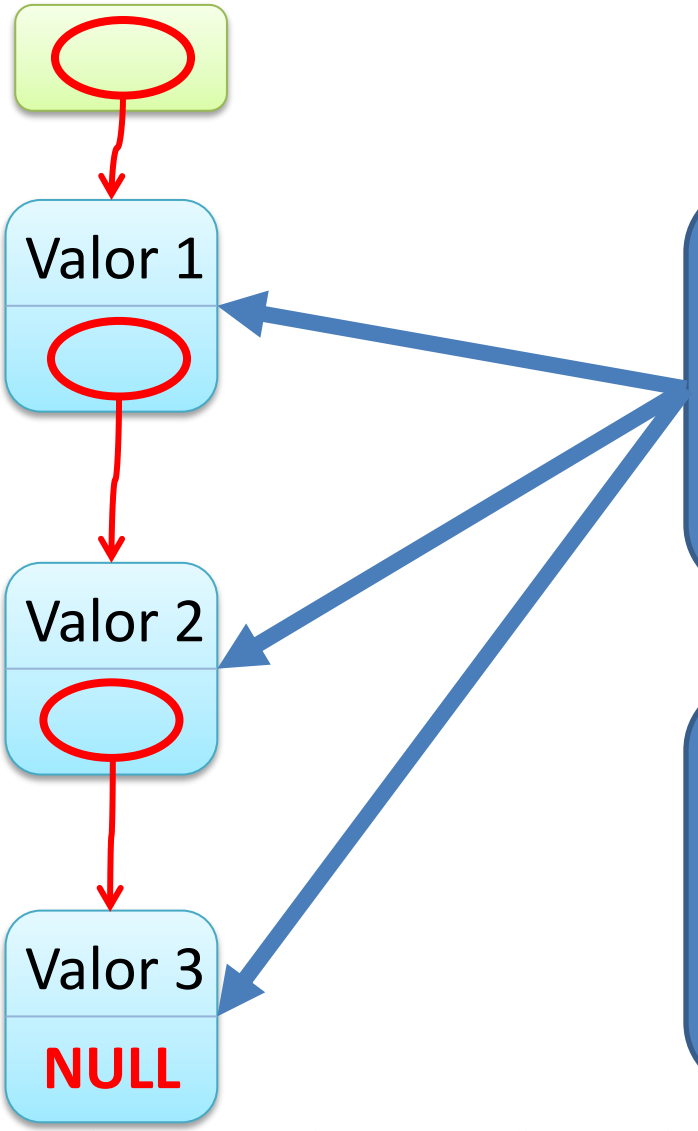
Representando Listas Encadeadas

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!



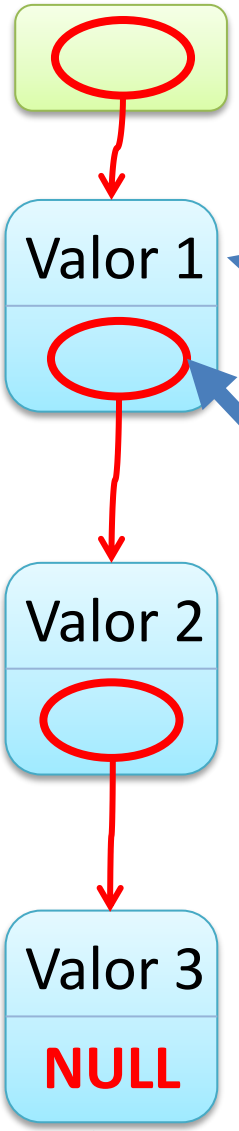
Representando Listas Encadeadas



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

Criaremos uma struct chamada **no**

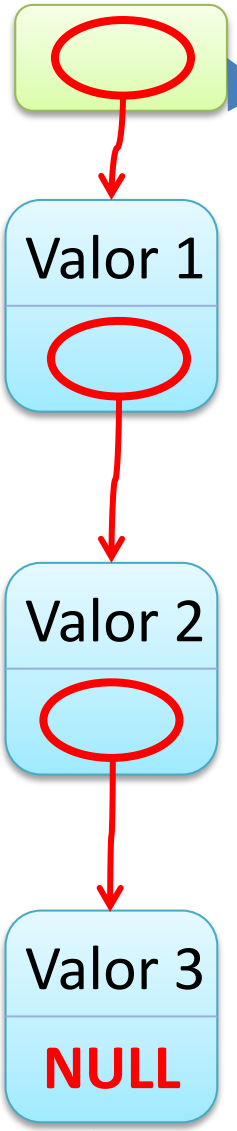
Representando Listas Encadeadas



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

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

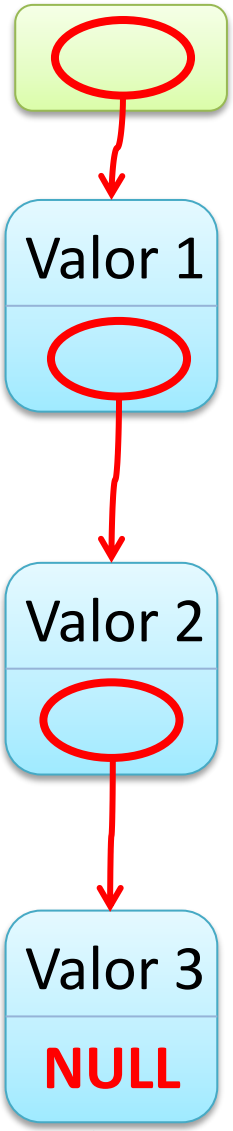
Representando Listas Encadeadas




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

O primeiro nó é apontado por um simples ponteiro

Representando Listas Encadeadas



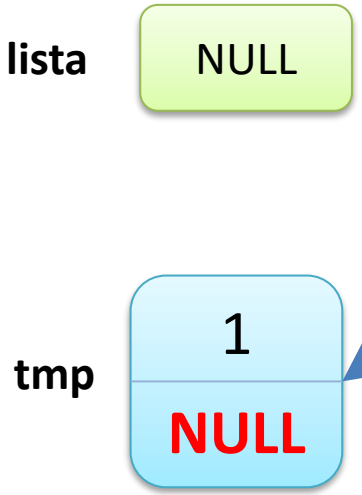

```
struct no {
        int valor;
        no *ptr;
};

no *lista;
```

Inicializando Listas Encadeadas e Inserindo nós

Listas Encadeadas: Inicializando


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

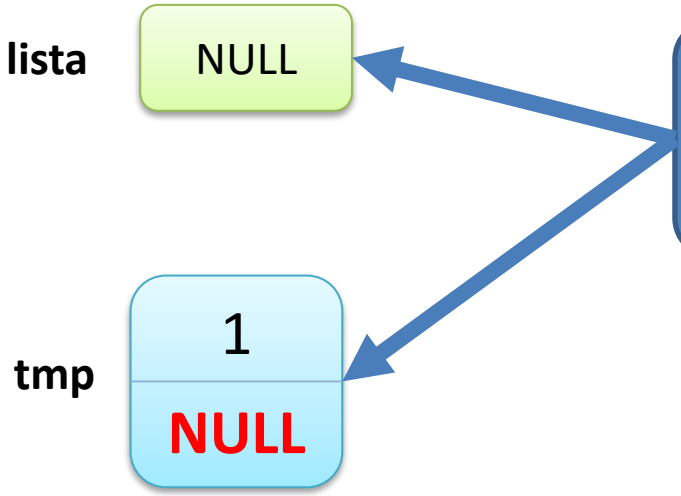
Listas Encadeadas: Inserção



Como criamos um elemento?

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

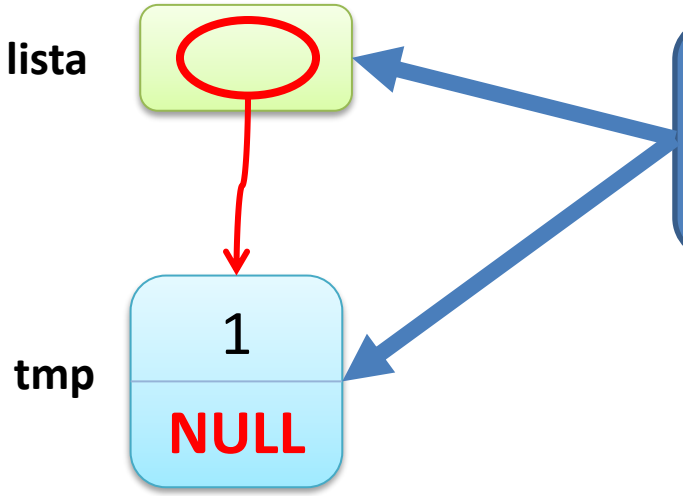
Listas Encadeadas: Inserção



Como colocar o elemento na lista?

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

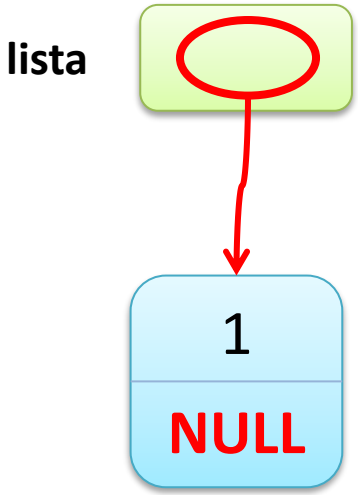
Listas Encadeadas: Inserção



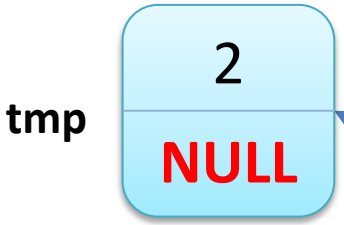
Como colocar o elemento na lista?

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

Listas Encadeadas: Inserção

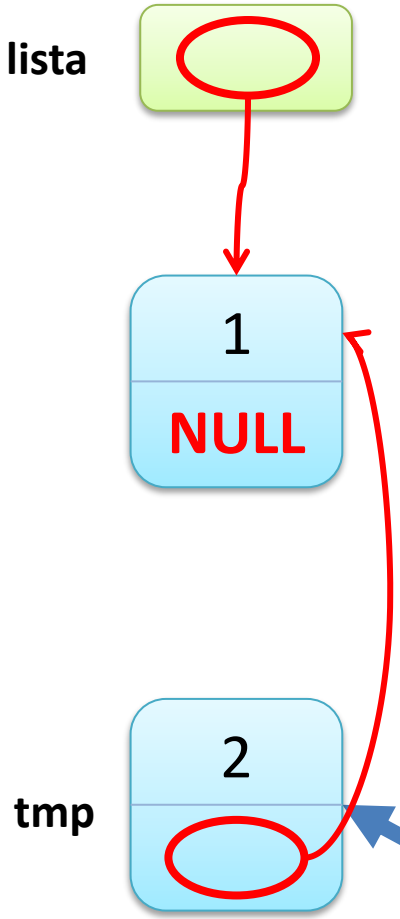



```
if (lista == NULL) {
        lista = tmp;
}
```



E para acrescentar um outro elemento?

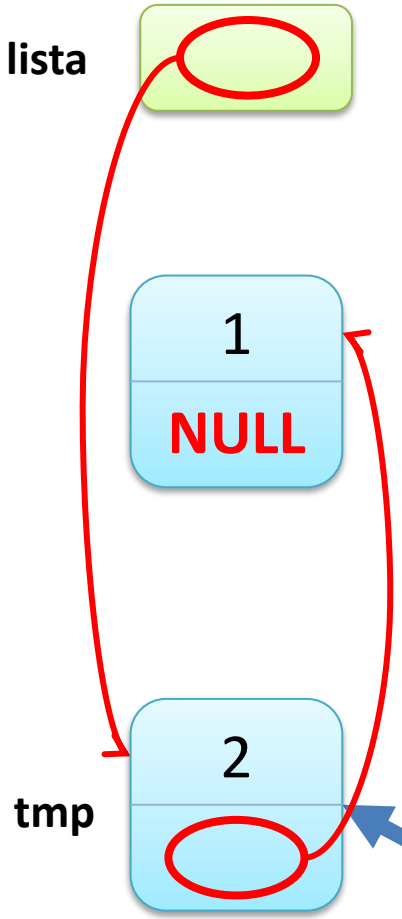
Listas Encadeadas: Inserção




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

E para acrescentar um outro elemento?

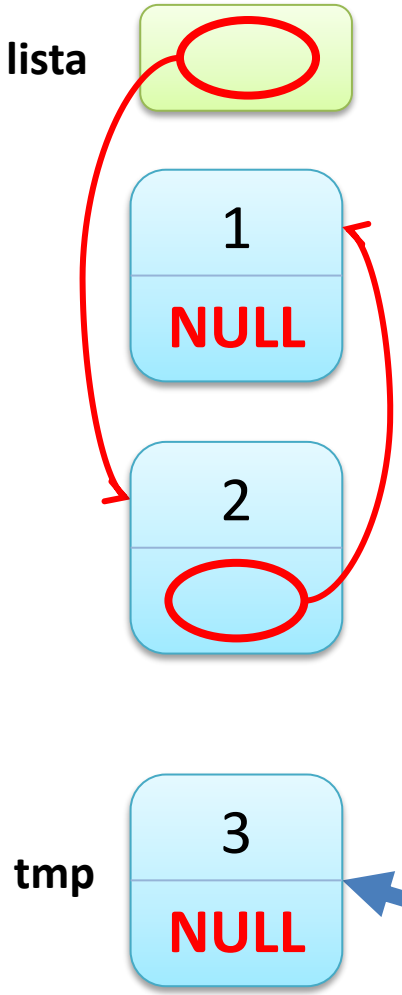
Listas Encadeadas: Inserção




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

E para acrescentar um outro elemento?

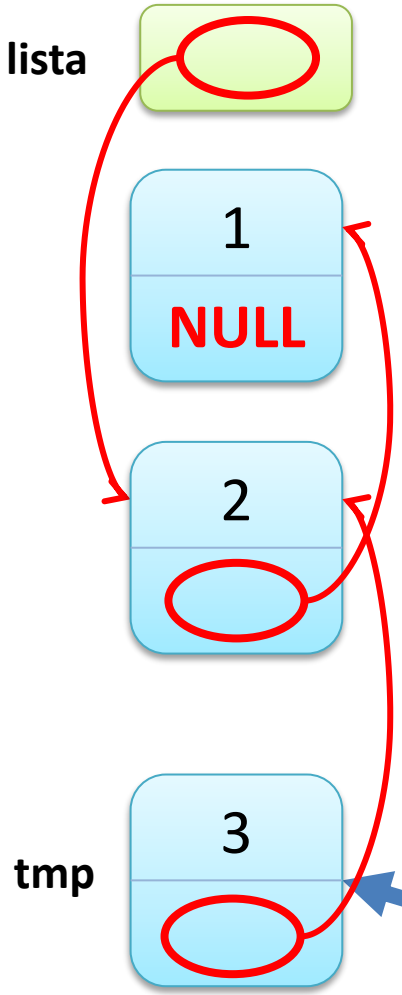
Listas Encadeadas: Inserção




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

E para acrescentar um outro elemento?

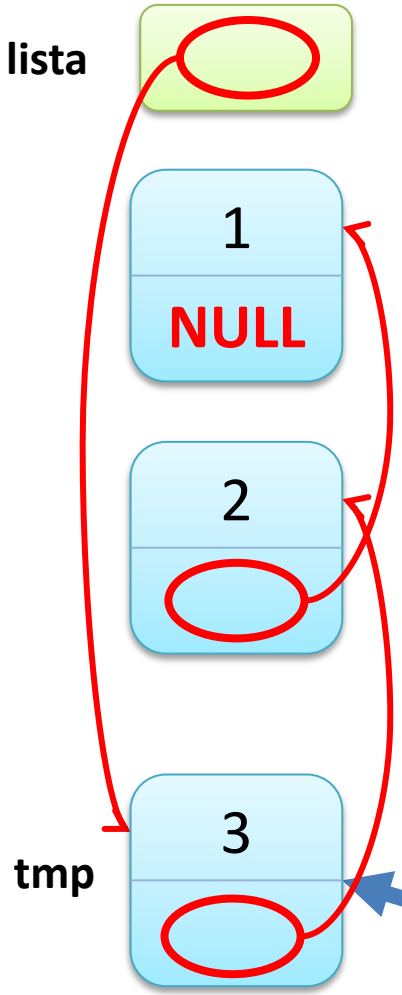
Listas Encadeadas: Inserção




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

E para acrescentar um outro elemento?

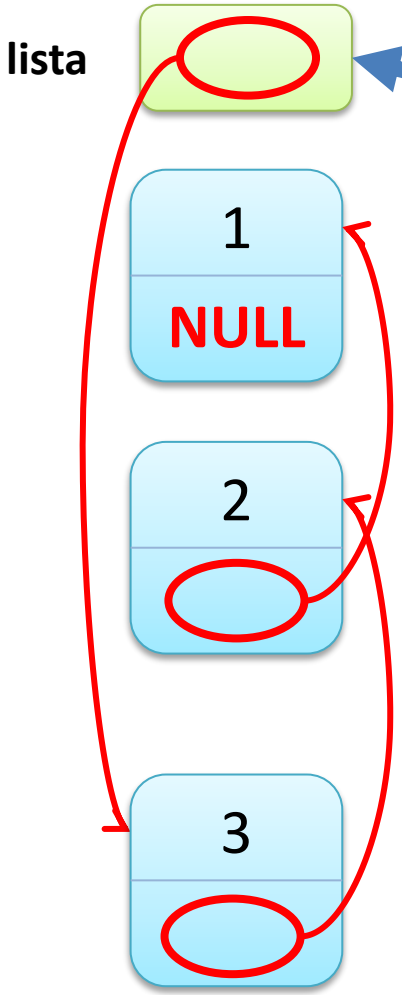
Listas Encadeadas: Inserção




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

E para acrescentar um outro elemento?

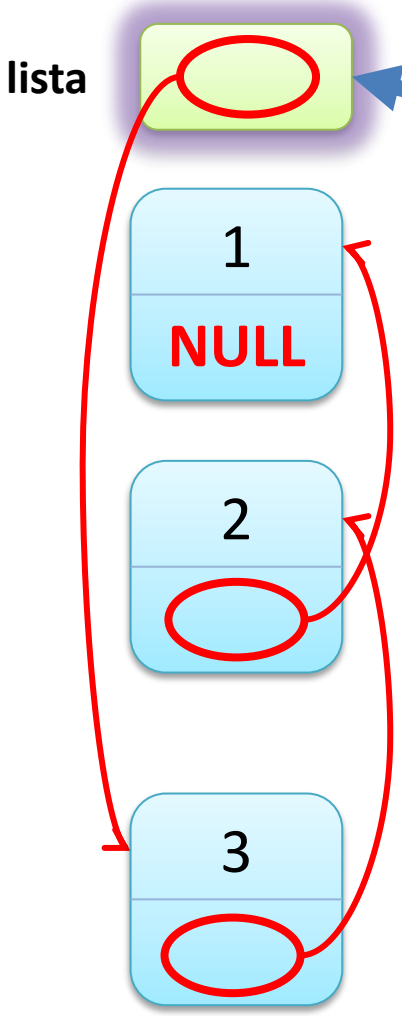
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

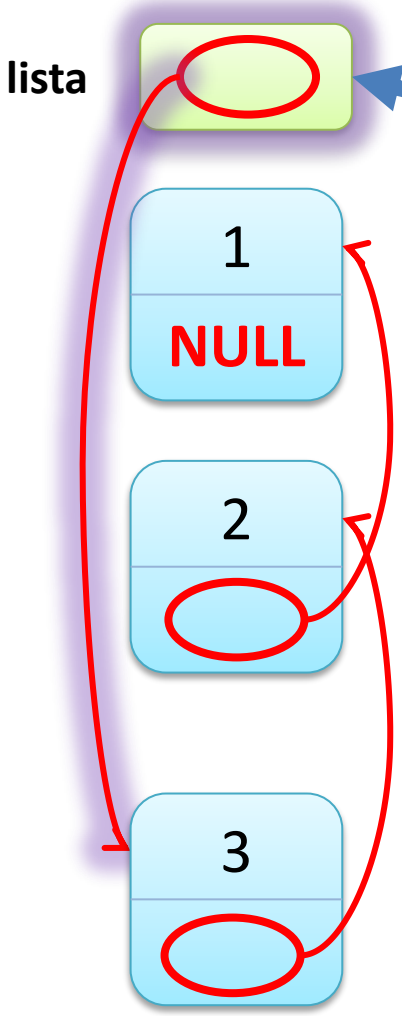
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

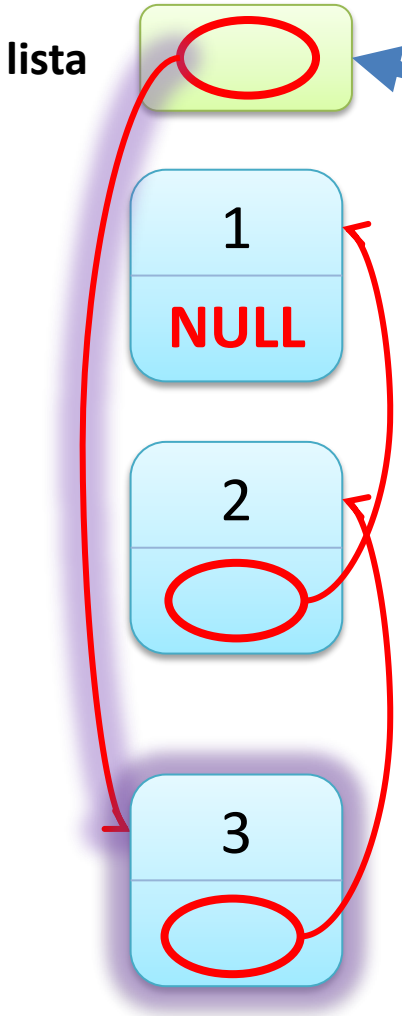
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

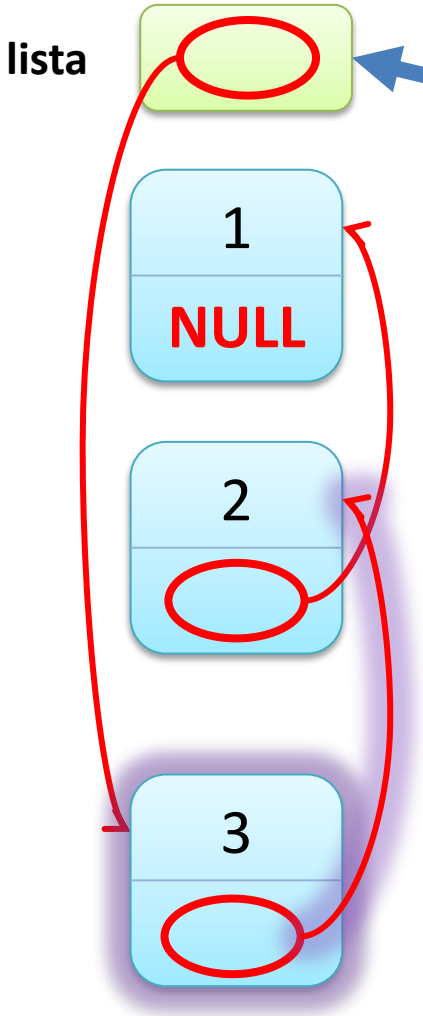
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

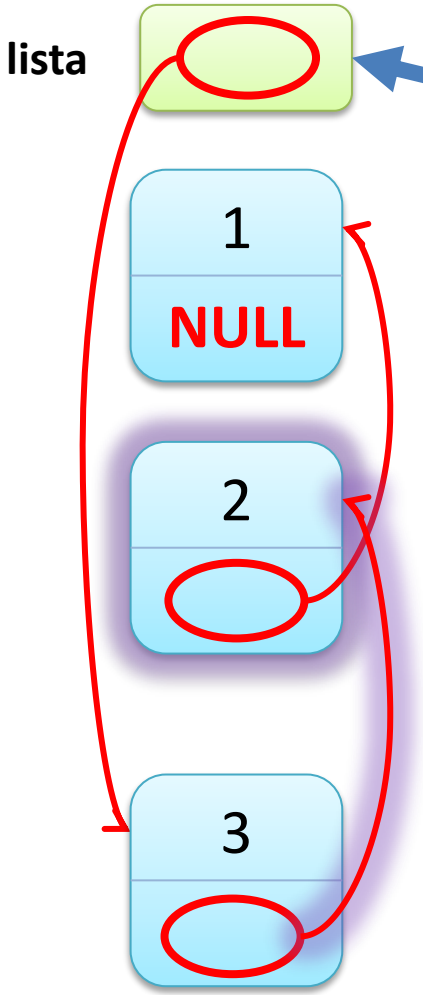
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

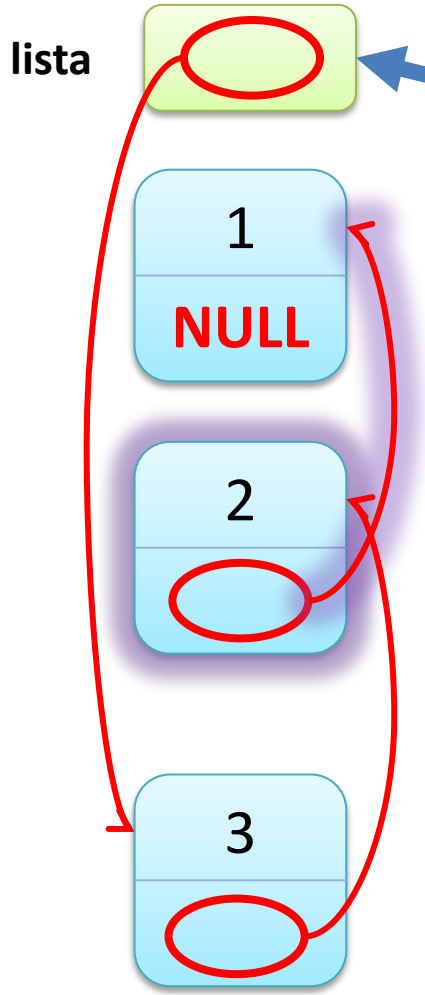
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

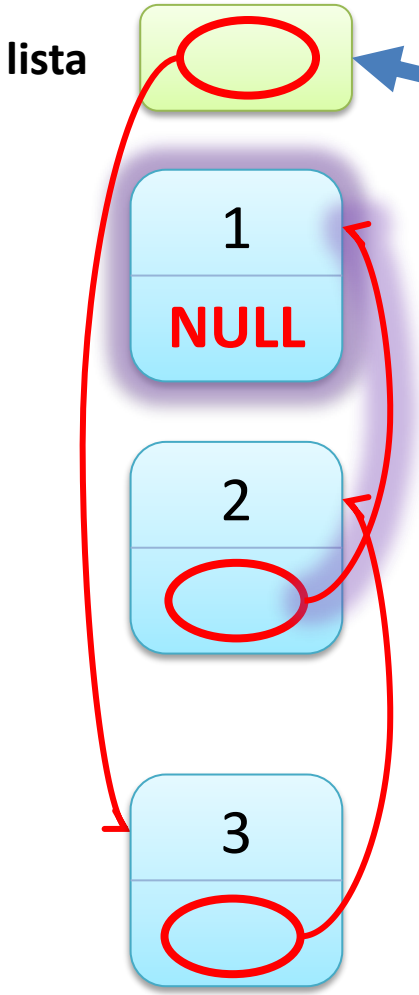
Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

Listas Encadeadas: Listar/Percorrer



Como percorrer uma lista encadeada?

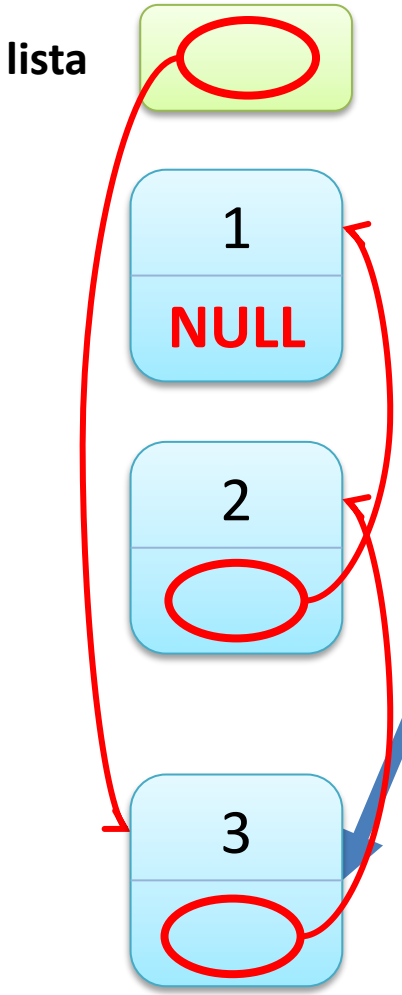
Basta seguir os ponteiros...

Listas Encadeadas: Listar/Percorrer

Como percorrer uma lista encadeada?

Basta seguir os ponteiros...

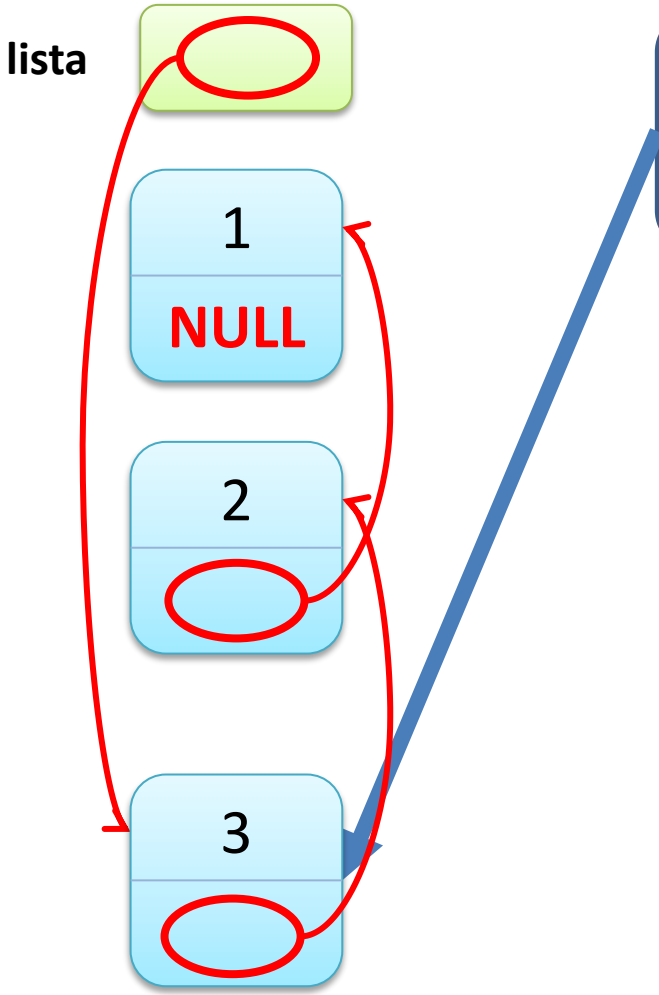
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

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

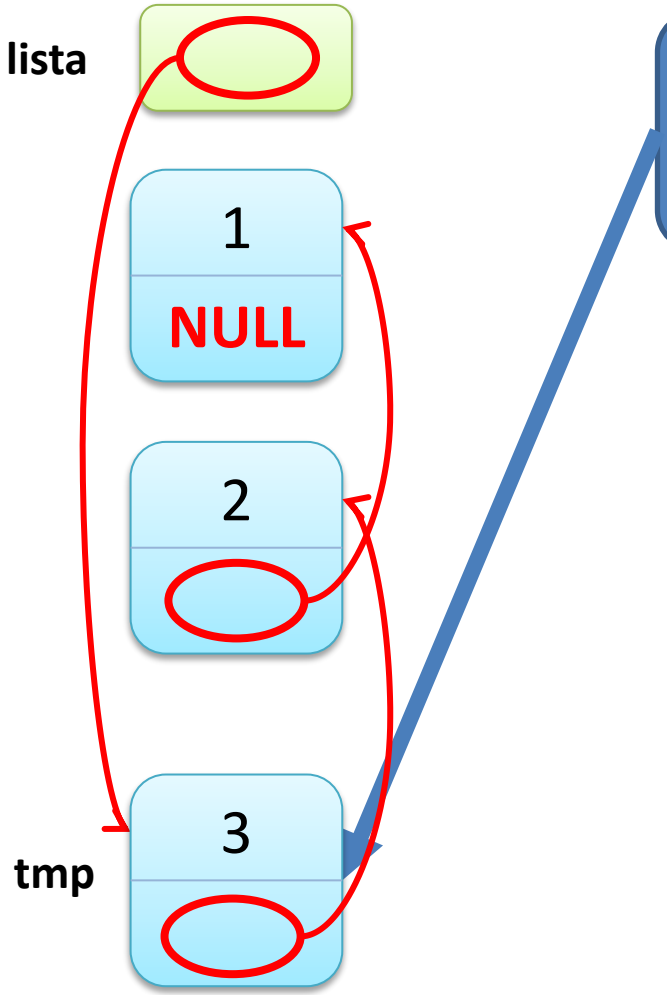
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

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

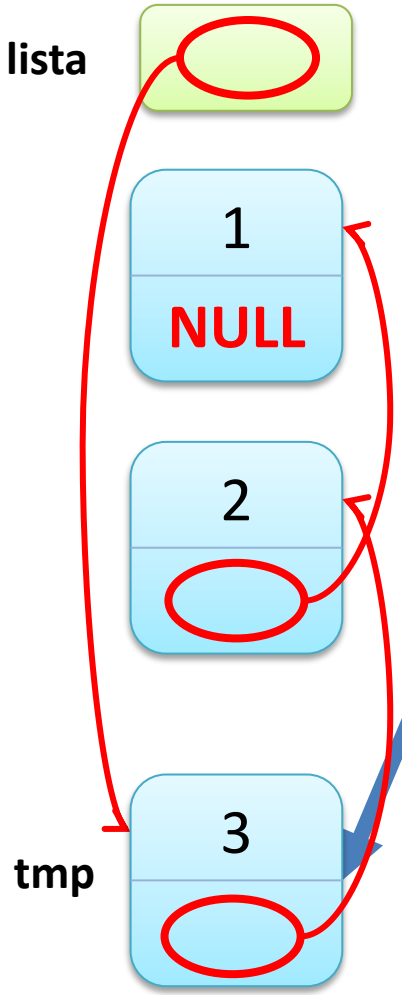
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

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

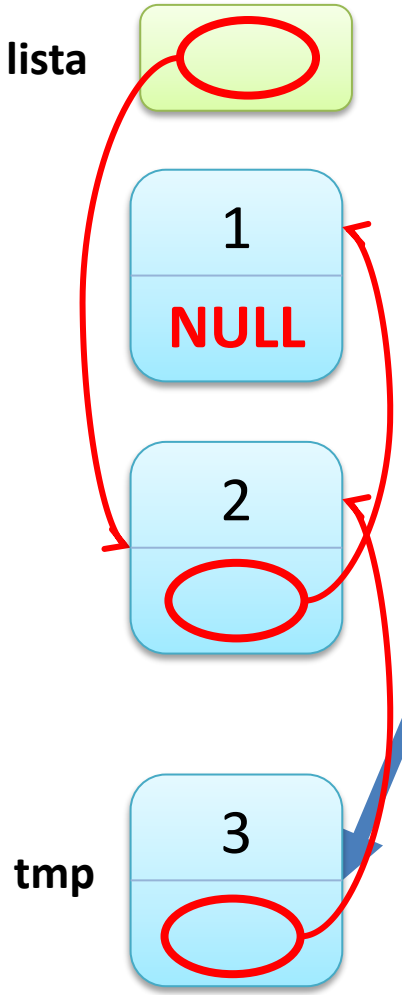
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

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

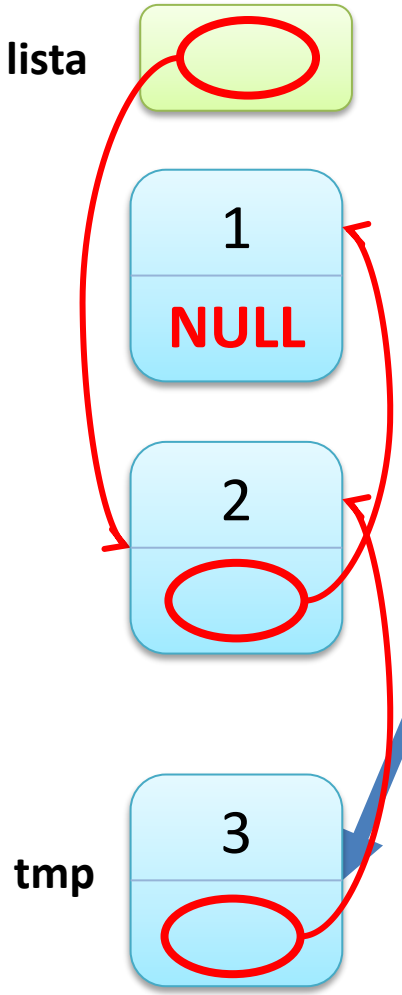
Listas Encadeadas: Remoção



Como remover o primeiro elemento?

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

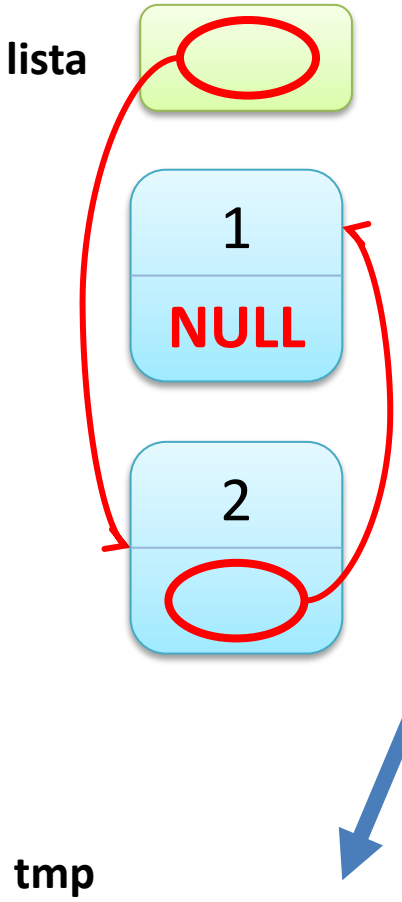
Listas Encadeadas: Remoção



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

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
Fazer no Padlet

ENCERRAMENTO

Resumo e Próximos Passos

- Funções
  - E Listas Lineares Estáticas
- Ponteiros e Alocação de Memória
  - Listas Dinâmicas
- **Pós Aula**: Aprenda Mais, Pós Aula e Desafio!
  - No padlet: https://padlet.com/djcaetano/paradigmas

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

PERGUNTAS?