ALGORITMOS III

Prof. Ms. Ronan Loschi.

ronan.loschi@unifasar.edu.br

31-98759-9555

UNIFASAR
CENTRO UNIVERSITÁRIO SANTA RITA

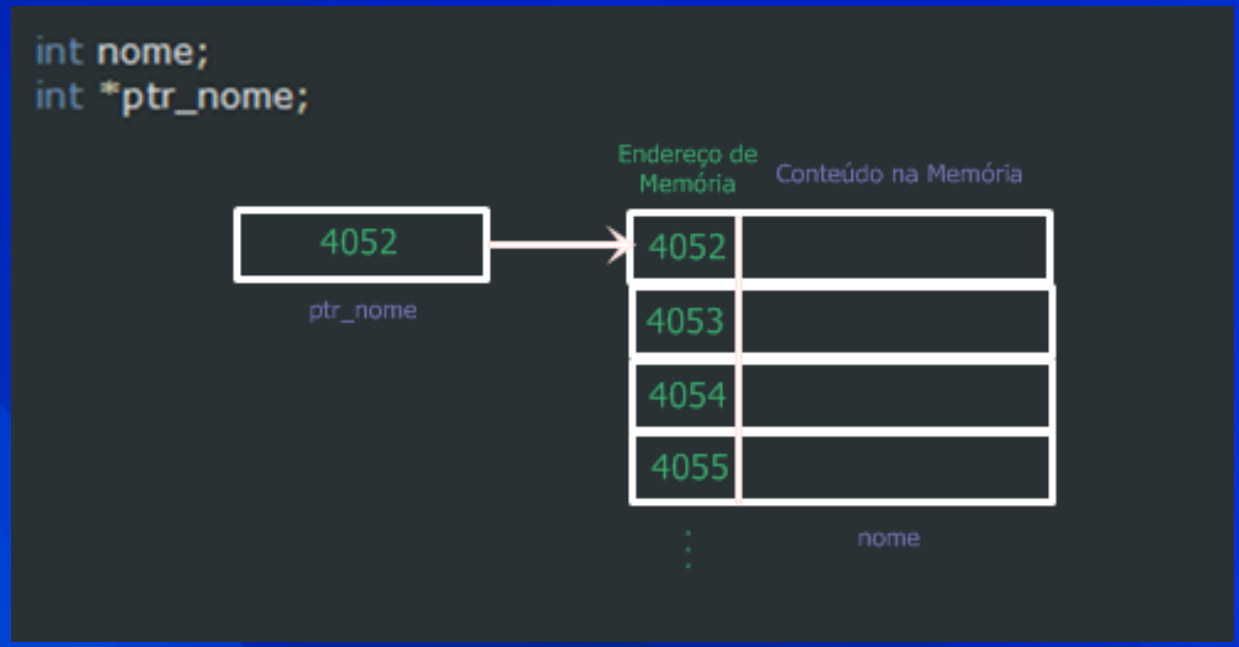
PONTEIROS EM C++

DEFINIÇÃO:

- Um ponteiro **é uma variável que armazena o endereço de memória de outra variável**.

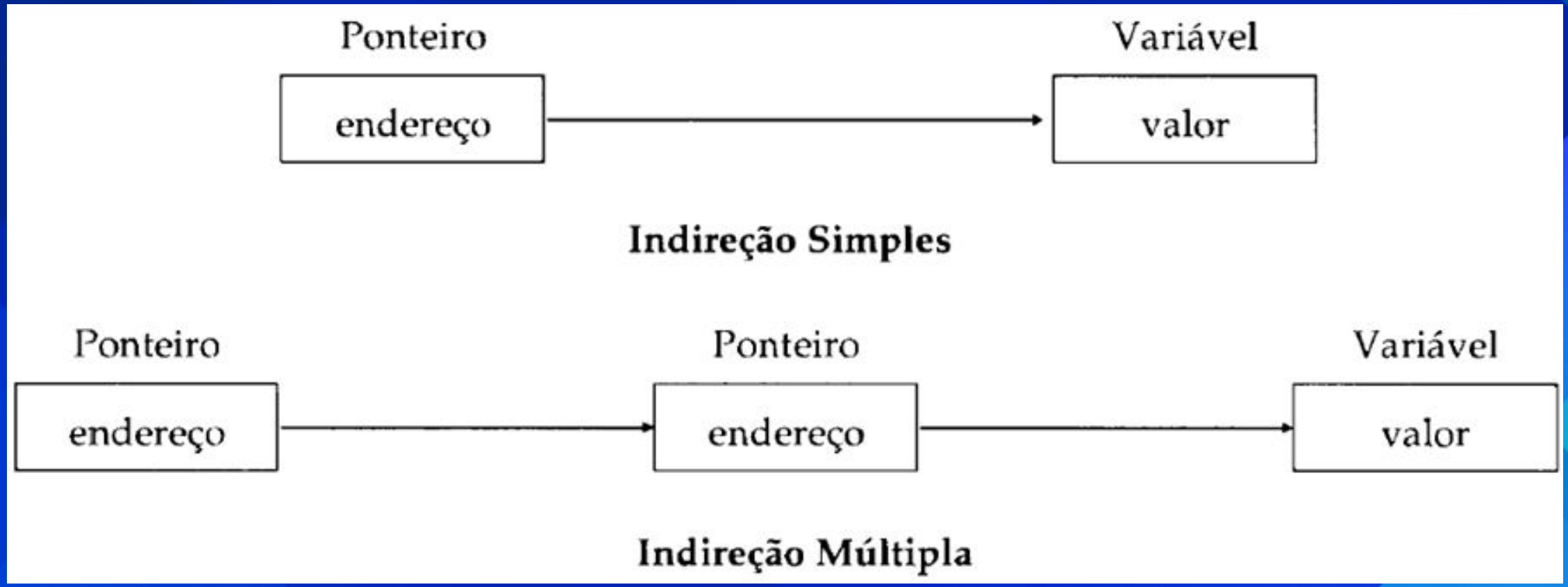
- Permite **manipular memória diretamente**, tornando o código mais eficiente.

- Utiliza-se o operador * (desreferência) para acessar o valor armazenado no endereço.



UNIFASAR
CENTRO UNIVERSITÁRIO SANTA RITA

Indireção simples e indireção Múltipla:



Aplicação de Ponteiros em Engenharia de Software

- Utilizados para alocação dinâmica de memória.
- Aplicados em estruturas de dados dinâmicas (**listas encadeadas, árvores, grafos**).
- Melhoram a **eficiência do código** ao manipular grandes volumes de dados.
- Fundamentais no desenvolvimento de sistemas operacionais e drivers.

DECLARAÇÃO DE UM PONTEIRO EM C++

```
Tipo* nome;
int* ptr;
```

- **ptr** é um ponteiro para um **inteiro**.
- O operador **&** retorna o endereço da variável.
- O operador ***** acessa o valor do endereço armazenado no ponteiro.

Exemplo 1:

```cpp
#include <iostream>

using namespace std;

int main() {

    int x = 10;  //DECLRANDO A VARIÁVEL X DO TIPO INTEIRO
    int* ptr;    //DECLARANDO O PONTEIRO PTR PARA APONTAR PARA UM INTEIRO
    ptr = &x;   //APONTANDO PTR PARA O ENDEREÇO DE X


    cout << "Valor de x: " << x << endl;

    cout << "Endereço de x: " << &x << endl;

    cout << "Valor armazenado no ponteiro: " << *ptr << endl;

    return 0;
}
```

Exemplo 2:

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

int main() {
    int x = 5;
    int* ptr = &x;

    cout << "Valor original de x: " << x << endl;
    cout << "Endereço de x: " << ptr << endl;

    *ptr += 10;  // Modifica o valor de x através do ponteiro

    cout << "Novo valor de x: " << x << endl;
    return 0;
}
```

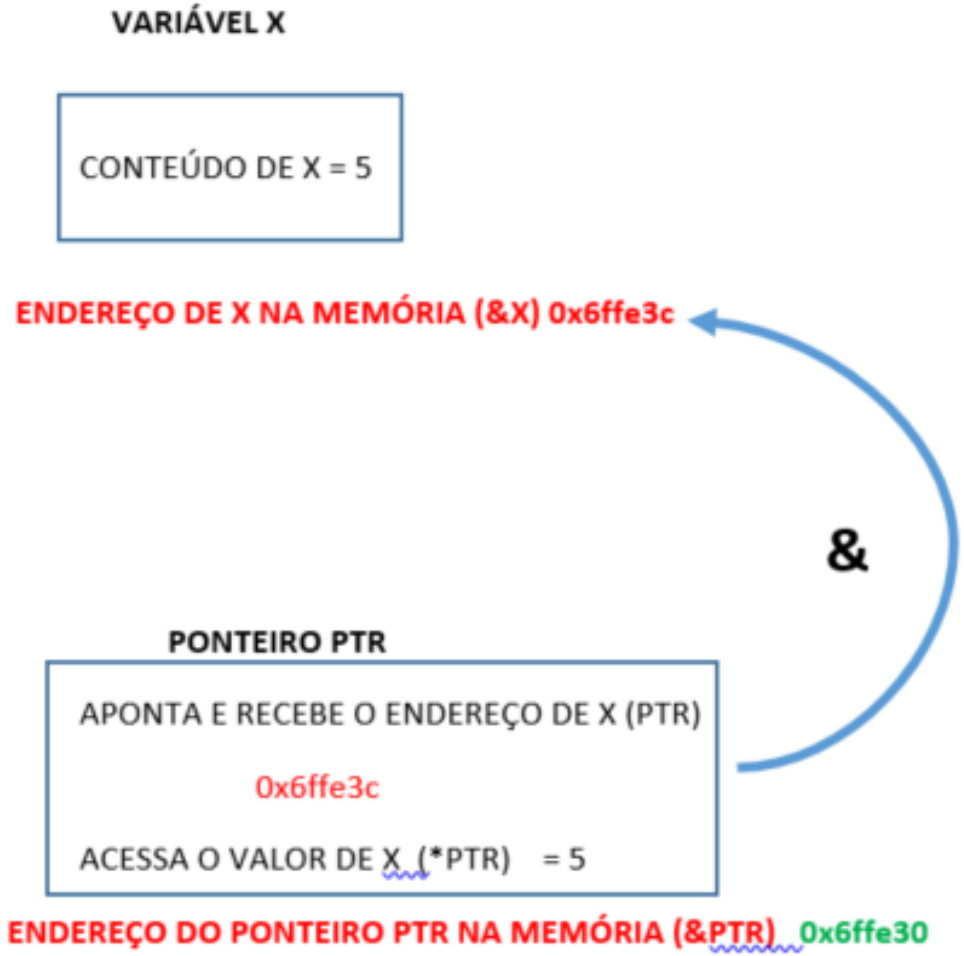
Exemplo 3:

```
Valor original de x: 5
Endereþo de x: 0x6ffe3c
Endereþo de x: 0x6ffe3c
Valor original de x ACESSADO pelo pornteiro ptr: 5
Endereþo do ponteiro ptr: 0x6ffe30
Novo valor de x: 15
Novo valor armazenado no pornteiro ptr: 15
```

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

int main() {
    int x = 5; // VALOR ORIGINAL DE X
    int* ptr = &x; // PONTEIRO PARA O ENDEREÇO DE X

    cout << "Valor original de x: " << x << endl;
    cout << "Endereço de x: " << &x << endl;  // endereço da variável X Diretamente em X
    cout << "Endereço de x: " << ptr << endl; // endereço para o qual o ponteiro aponta/recebe
    cout << "Valor original de x ACESSADO pelo pornteiro ptr: " << *ptr << endl;

    cout << "Endereço do ponteiro ptr: " << &ptr << endl; // Endereço do Ponteirp ptr

    *ptr += 10;  // Modifica o valor de x através do ponteiro

    cout << "Novo valor de x: " << x << endl;
    cout << "Novo valor armazenado no pornteiro ptr: " << *ptr << endl;
    return 0;
}
```

UNIFASAR
CENTRO UNIVERSITÁRIO SANTA RITA

Exemplo 3:




```
*PTR += 10; // Modifica o valor de x através do ponteiro
```

UNIFASAR
CENTRO UNIVERSITÁRIO SANTA RITA

Exemplo 4:

```
O endereço de a na memória é = 0x6ffe2c
O endereço de b na memória é = 0x6ffe28

O valor original de b é o maior: 10
O valor modificado de a é o maior: 17

Agora os valores de a e de b são iguais, sendo: a= 17 e b =17
```

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

int main() {
    setlocale(LC_ALL, "Portuguese");
    int a = 7; // VALOR ORIGINAL DE A
    int b = 10; // VALOR ORIGINAL DE B
    int* ptr1 = &a; // PONTEIRO PARA O ENDEREÇO DE a
    int* ptr2 = &b; // PONTEIRO PARA O ENDEREÇO DE b

    cout << "O endereço de a na memória é = " << ptr1 << endl;
    cout << "O endereço de b na memória é = " << ptr2 << endl<<endl;

    if (*ptr1>*ptr2)
        cout << "O valor original de a é o maior: " << *ptr1 << endl;
    else
        cout << "O valor original de b é o maior: " << *ptr2 << endl;


    *ptr1 += 10;  // Modifica o valor de a através do ponteiro
    *ptr2 += 5;  // Modifica o valor de b através do ponteiro

if (*ptr1>*ptr2)
        cout << "O valor modificado de a é o maior: " << *ptr1 << endl<<endl;
    else
        cout << "O valor modificado de b é o maior: " << *ptr2 << endl<<endl;

    ptr2=ptr1;
    cout << "Agora os valores de a e de b são iguais, sendo: a= " << *ptr1 <<" e b =" <<*ptr2<<endl;

    return 0;
}
```

UNIFASAR
CENTRO UNIVERSITÁRIO SANTA RITA

Exemplo 4 - VETOR:

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

int main() {
    setlocale(LC_ALL, "Portuguese");

    int vetor[10];  // Declaração do vetor de 10 posições
    int* ptr = vetor; // Ponteiro apontando para o primeiro elemento do vetor

    // Preenchendo o vetor com valores de 1 a 10
    for (int i = 0; i < 10; i++) {
        //*(ptr + i) = i + 1; // Atribui o valor (i+1) OU (0+1 = 1) à posição i do vetor via ponteir
        ptr[i] = i + 1; // Atribui o valor (i+1) OU (0+1 = 1) à posição i do vetor via ponteiro
    }

    // Acessando e exibindo os valores usando o ponteiro
    cout << "Valores do vetor acessados pelo ponteiro:" << endl;
    for (int i = 0; i < 10; i++) {
        cout << "Posição " << i << ": " <<ptr[i]<< endl;
        cout << "Posição " << i << ": " <<*(ptr+i)<< endl;
        }

 cout << "Endereços do vetor:" << endl;
    for (int i = 0; i < 10; i++) {
        cout << "Endereço " << i << " acessado pelo vetor: " <<&vetor[i]<<endl;
        cout << "Endereço " << i << " acessado pelo ponteiro: "<<(ptr+i)<<endl<<endl;
    }

    return 0;
}
```

```
Valores do vetor acessados pelo ponteiro:
Posição 0: 1
Posição 0: 1
Posição 1: 2
Posição 1: 2
Posição 2: 3
Posição 2: 3
Posição 3: 4
Posição 3: 4
Posição 4: 5
Posição 4: 5
Posição 5: 6
Posição 5: 6
Posição 6: 7
Posição 6: 7
Posição 7: 8
Posição 7: 8
Posição 8: 9
Posição 8: 9
Posição 9: 10
Posição 9: 10
Endereços do vetor:
Endereço 0 acessado pelo vetor: 0x6ffe00
Endereço 0 acessado pelo ponteiro: 0x6ffe00

Endereço 1 acessado pelo vetor: 0x6ffe04
Endereço 1 acessado pelo ponteiro: 0x6ffe04

Endereço 2 acessado pelo vetor: 0x6ffe08
Endereço 2 acessado pelo ponteiro: 0x6ffe08

Endereço 3 acessado pelo vetor: 0x6ffe0c
Endereço 3 acessado pelo ponteiro: 0x6ffe0c

Endereço 4 acessado pelo vetor: 0x6ffe10
Endereço 4 acessado pelo ponteiro: 0x6ffe10

Endereço 5 acessado pelo vetor: 0x6ffe14
Endereço 5 acessado pelo ponteiro: 0x6ffe14

Endereço 6 acessado pelo vetor: 0x6ffe18
Endereço 6 acessado pelo ponteiro: 0x6ffe18

Endereço 7 acessado pelo vetor: 0x6ffe1c
Endereço 7 acessado pelo ponteiro: 0x6ffe1c

Endereço 8 acessado pelo vetor: 0x6ffe20
Endereço 8 acessado pelo ponteiro: 0x6ffe20

Endereço 9 acessado pelo vetor: 0x6ffe24
Endereço 9 acessado pelo ponteiro: 0x6ffe24
```

Exemplo 5 - MATRIZ:

```cpp
#include <iostream>
using namespace std;
int main() {
    setlocale(LC_ALL, "Portuguese");
    int matriz[5][5]; // Declaração de uma matriz 5x5
    int *ptr; // Ponteiro para armazenar o endereço da matriz
    int i, j, cont = 1;
    ptr = &matriz[0][0]; // Inicializando o ponteiro no primeiro elemento da matriz

    // Preenchendo a matriz com valores de 1 a 25
    for (i = 0; i < 5; i++) {
        for (j = 0; j < 5; j++) {
            matriz[i][j] = cont;
            cont++;
        }
    }
  cout <<"Exibindo os valores da matriz usando o ponteiro:"<<endl;
    for (i = 0; i < 25; i++) {
        cout << *(ptr + i) << " "; // Acessando elementos via aritmética de ponteiros
        if ((i + 1) % 5 == 0) cout << endl; // Quebra de linha para formatar a matriz
    }
    cout<<endl;

    cout <<"Exibindo os endereços da matriz usando o ponteiro:"<<endl;
    for (i = 0; i < 25; i++) {
        cout << (ptr + i) << " "; // Acessando elementos via aritmética de ponteiros
        if ((i + 1) % 5 == 0) cout << endl; // Quebra de linha para formatar a matriz
    }
    cout <<endl;
    cout <<"Exibindo os endereços da matriz:"<<endl;
    for (i = 0; i < 5; i++) {
        for (j = 0; j < 5; j++) {
        cout << (&matriz[i][j]) << " "; // Acessando elementos via aritmética de ponteiros
        }
    cout << endl; // Quebra de linha para formatar a matriz
}
    return 0;}
```

```
Exibindo os valores da matriz usando o ponteiro:
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25

Exibindo os endereços da matriz usando o ponteiro:
0x6ffdd0 0x6ffdd4 0x6ffdd8 0x6ffddc 0x6ffde0
0x6ffde4 0x6ffde8 0x6ffdec 0x6ffdf0 0x6ffdf4
0x6ffdf8 0x6ffdfc 0x6ffe00 0x6ffe04 0x6ffe08
0x6ffe0c 0x6ffe10 0x6ffe14 0x6ffe18 0x6ffe1c
0x6ffe20 0x6ffe24 0x6ffe28 0x6ffe2c 0x6ffe30

Exibindo os endereços da matriz:
0x6ffdd0 0x6ffdd4 0x6ffdd8 0x6ffddc 0x6ffde0
0x6ffde4 0x6ffde8 0x6ffdec 0x6ffdf0 0x6ffdf4
0x6ffdf8 0x6ffdfc 0x6ffe00 0x6ffe04 0x6ffe08
0x6ffe0c 0x6ffe10 0x6ffe14 0x6ffe18 0x6ffe1c
0x6ffe20 0x6ffe24 0x6ffe28 0x6ffe2c 0x6ffe30
```

Exercício 1:

Exercício 1: Ajuste Dinâmico de Salário

Em uma empresa de tecnologia, o RH precisa atualizar o salário de um funcionário após uma promoção. O programa deve:

1. Receber o salário atual do funcionário.
2. Receber o percentual de aumento.
3. Utilizar um ponteiro para modificar o valor do salário.
4. Exibir o salário antes e depois do aumento.

Exercício 2:

Comparação de Idades

Dois amigos querem saber qual deles é mais velho. O programa deve:

1. Solicitar ao usuário duas idades.
2. Utilizar ponteiros para comparar os valores.
3. Exibir a idade maior e a diferença entre elas.

Exercício 3:

Preenchendo e Exibindo Notas com Ponteiros

Uma professora deseja armazenar as notas de uma prova de cinco alunos. O programa deve:

1. Solicitar cinco notas ao usuário.
2. Utilizar um ponteiro para armazenar e acessar esses valores.
3. Exibir as notas informadas.

Exercício 4:

Analisando Temperaturas Semanais

Um meteorologista deseja analisar a variação de temperatura durante uma semana. O programa deve:

1. Solicitar sete temperaturas ao usuário.
2. Utilizar um ponteiro para encontrar e exibir a maior e a menor temperatura.

Exercício 4:

Matriz de Estoque em um Supermercado

Um supermercado deseja armazenar a quantidade de produtos disponíveis em diferentes seções usando uma matriz 3x3. O programa deve:

1. Solicitar ao usuário que preencha a matriz com as quantidades de produtos.
2. Utilizar um ponteiro para acessar e exibir os valores da matriz.

OBRIGADO!

UNIFASAR
CENTRO UNIVERSITÁRIO SANTA RITA