

## Estrutura de Dados

Universidade do Estado de Mato Grosso  
Faculdade de Ciências Exatas e Tecnológicas

Prof. Me. João Ricardo dos Santos Rosa

[joao.santos@unemat.br](mailto:joao.santos@unemat.br)

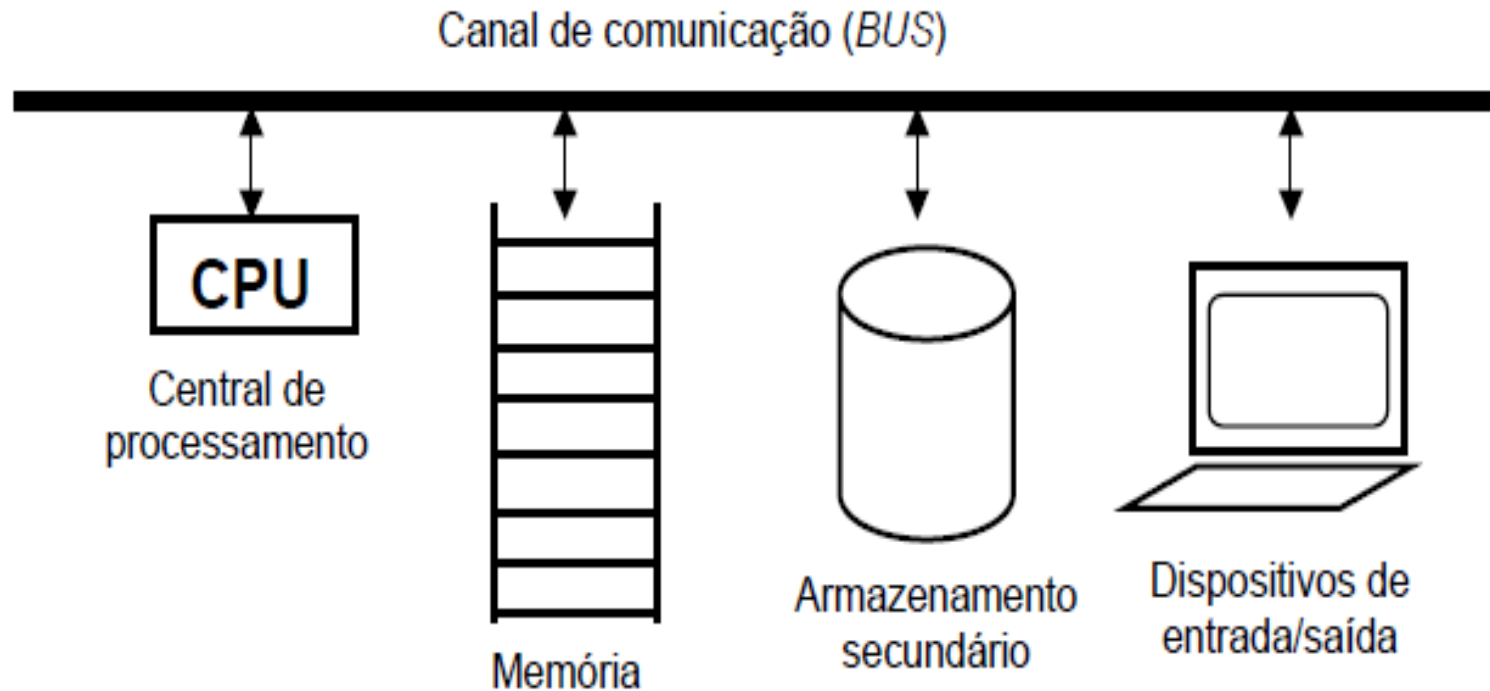
2025/01

# Introdução a LINGUAGEM C++

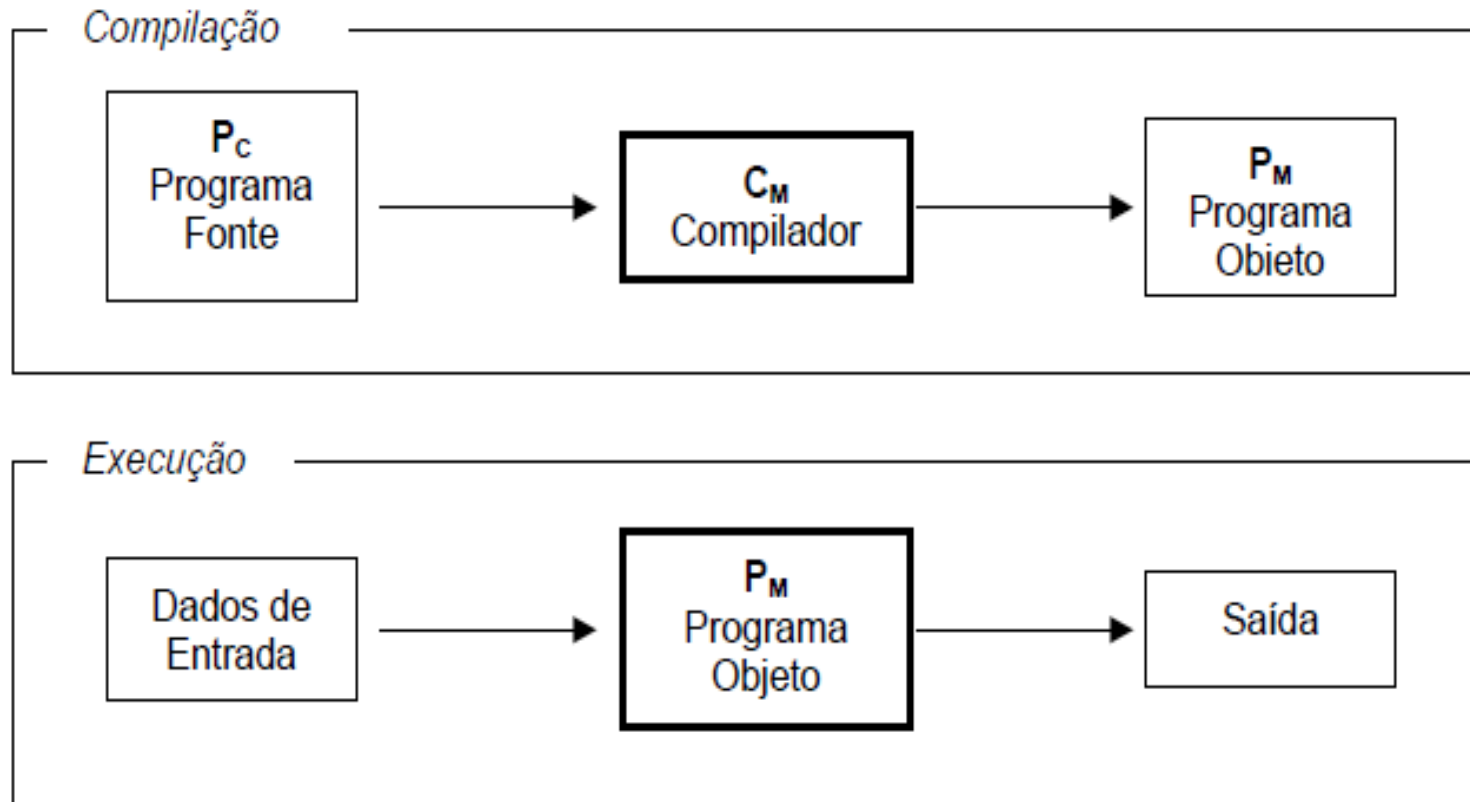
# *Linguagem C++*

- A linguagem C++, assim como as linguagens Fortran e Pascal, são ditas linguagens “convencionais”, projetadas a partir dos elementos fundamentais da arquitetura de Von Neuman, que serve como base para praticamente todos os computadores em uso

# *Elementos de um computador típico*



# *Execução de programas com linguagem compilada*



# *Em pseudocódigo*

**Algoritmo** PrimeiroPrograma

**Variáveis**

**Início**

Escreval “Meu primeiro programa em C”

**Fim**

# *Em C++*

```
#include <iostream>
```

```
int main() {  
    std::cout << "Meu Primeiro Programa em  
C++\n";  
    return 0;  
}
```

# Alocação na memória

Tipo	Tamanho	Representatividade
char	1 byte	-128 a 127
unsigned char	1 byte	0 a 255
short int	2 bytes	-32 768 a 32 767
unsigned short int	2 bytes	0 a 65 535
long int	4 bytes	-2 147 483 648 a 2 147 483 647
unsigned long int	4 bytes	0 a 4 294 967 295

Tipo	Tamanho	Representatividade
char	1 byte	-128 a 127
unsigned char	1 byte	0 a 255
short int	2 bytes	-32 768 a 32 767
unsigned short int	2 bytes	0 a 65 535
long int	4 bytes	-2 147 483 648 a 2 147 483 647
unsigned long int	4 bytes	0 a 4 294 967 295

# *Tipos de dados e representação*

Linguagem C++	Tipo de Dado
Char	Armazena do tipo <b>caracteres</b> .
Int	armazena valores numéricos do tipo <b>Inteiros</b>
Float	Armazena números com ponto flutuante ( <b>reais</b> ) com precisão simples.
Double	armazena números com ponto flutuante, com precisão dupla, ou seja normalmente possui o <b>dobro da capacidade de uma variável do tipo float</b> .



# *Variáveis*



# *Sintaxe e declarações de variáveis em C++*

- Em C devemos listar primeiro o tipo, depois o nome da variável.

## **Sintaxe:**

**<tipo> <nome\_da\_variável>;**

**Int** idade;

**Float** kg;

# *Sintaxe e declarações de variáveis em C++*

**Exemplo de declaração de variável do tipo **inteiro**:**

**int** contador;

onde:

- **int** é o tipo da variável (inteiro)
- contador é o **nome da variável**.

**Declarando uma variável **real** (ponto flutuante):**

**float** salario;

**Declarando uma variável do tipo **caracter**:**

**char** letra;

# *Regras para nome de Variáveis*

Devemos observar algumas regras para criar os identificadores das variáveis.

O nome de uma variável deve sempre iniciar com uma letra, jamais deve iniciar com um número.

Exemplo:

`float total2; //` está correto

`float 2total; //` está errado, pois não iniciou com uma letra.

# *Regras para nome de Variáveis*

- Como visto, o nome de uma variável pode conter dígitos em qualquer posição, menos na primeira.

## **Utilize nomes representativos**

Exemplo:

```
float salario;
```

- Neste exemplo o nome da variável já diz o que ela vai armazenar.

# *Operadores racionais e Lógicos*

## **Os operadores relacionais em C++ são:**

< menor que

> maior que

<= menor ou igual que

>= maior ou igual que

== igual a

!= diferente de

&& operador binário E (AND)

|| operador binário OU (OR)



# *Saida de dados*

# Função `std::cout`

## Função `std::cout`

Em C++, a forma padrão e mais segura para saída de dados é através da biblioteca `<iostream>` e do objeto `std::cout`.

O `cout` permite exibir valores de forma formatada (sejam eles constantes, variáveis ou o resultado de expressões) para o console.

Informalmente, podemos dizer que a forma básica de uso é:

```
std::cout << valor_a_ser_impresso;
```

O operador `<<` (chamado de operador de inserção de fluxo) é usado para "inserir" os dados no fluxo de saída. Ele pode ser encadeado para imprimir múltiplos valores na mesma linha.



# *Exemplo de saída de Dados*

```
#include <iostream>
```

```
int main() {
```

```
    int idade = 25;
```

```
    float peso = 70.5;
```

```
    std::cout << "A idade é: " << idade << " anos." << std::endl;
```

```
    std::cout << "O peso é: " << peso << " kg." << std::endl;
```

```
    std::cout << "O resultado de 10 * 5 é: " << 10 * 5 << std::endl;
```

```
    return 0;
```

```
}
```

# String de controle

Para exibir uma variável na tela, usamos `std::cout` e o operador `<<`. A melhor parte é que não precisamos de códigos de formato, pois o C++ entende o tipo da variável automaticamente.

Para exibir:

- Caracteres (char)
  - `std::cout << 'A';`
- Números inteiros (int)
  - `std::cout << 10;`
- Números reais (float, double)
  - `std::cout << 3.14;`
- Strings (char[ ], std::string)
  - `std::cout << "Olá Mundo";`

# *String de controle*

## Exemplo:

```
#include <iostream>
```

```
int main() {
```

```
    int idade = 25;
```

```
    float altura = 1.75f;
```

```
    std::cout << "Você tem " << idade << " anos e " << altura << " metros. \n" <<  
    std::endl;
```

```
    return 0;
```

```
}
```

# *Comando de Atribuição*

Atribuir significa armazenar um valor em uma variável. Em linguagem C usamos o operador **=** para fazer uma atribuição.

Exemplo:

```
X = 10;
```

Este comando atribui o valor 10 para a variável X.

Para poder atribuir um valor a uma variável, esta tem que ter sido previamente declarada.

# *Exemplo 1*

**Desenvolva um programa que tem a finalidade de converter valores de temperatura dados em Celsius para Fahrenheit e exibir ao usuário.(Ainda não aprendemos entrada de dados). O valor para o grau célsius fica a seu critério.**

**Considere a fórmula de conversão:**

$$\text{Fahrenheit} = (\text{Celsius} * 1,8) + 32$$

# Exemplo 1

```
#include <iostream>
```

```
int main() {
```

```
    float celsius, fahrenheit;
```

```
    celsius = 45;
```

```
    fahrenheit = (celsius * 1.8) + 32;
```

```
    // std::endl foi substituído por "\n"
```

```
    std::cout << "O valor convertido em Fahrenheit é " << fahrenheit << "\n";
```

```
    return 0;
```

```
}
```

## *Exemplo 2*

**Desenvolva um programa que tem a finalidade de mostrar o nome de 5 pessoas e suas respectivas idades.**

# Exemplo 2

```
#include <iostream>
```

```
int main() {
```

```
    int idade1, idade2, idade3, idade4, idade5;
```

```
    idade1 = 22;
```

```
    idade2 = 44;
```

```
    idade3 = 55;
```

```
    idade4 = 66;
```

```
    idade5 = 77;
```

```
    // Todos os std::endl foram substituídos por "\n"
```

```
    std::cout << "A idade de lucas é " << idade1 << " anos\n";
```

```
    std::cout << "A idade de joão é " << idade2 << " anos\n";
```

```
    std::cout << "A idade de pedro é " << idade3 << " anos\n";
```

```
    std::cout << "A idade de fernando é " << idade4 << " anos\n";
```

```
    std::cout << "A idade de jose é " << idade5 << " anos\n";
```

```
    return 0;
```

```
}
```



# *Atividade 1*

**Crie um programa que mostre o nome de 5 cidades e suas respectivas temperaturas**

# Atividade 1

```
#include <iostream>
```

```
#include <string>
```

```
int main() {
```

```
    std::string cidade1 = "Caceres";
```

```
    float temperatura1 = 25.5;
```

```
    std::string cidade2 = "Sinop";
```

```
    float temperatura2 = 30.0;
```

```
    std::string cidade3 = "Cuiaba";
```

```
    float temperatura3 = 32.8;
```

```
    std::string cidade4 = "Rondonopolis";
```

```
    float temperatura4 = 28.1;
```

```
    // Todos os std::endl foram substituídos por "\n"
```

```
    std::cout << "--- Temperaturas das Cidades ---\n";
```

```
    std::cout << "Cidade: " << cidade1 << " | Temperatura: " << temperatura1 << "°C\n";
```

```
    std::cout << "Cidade: " << cidade2 << " | Temperatura: " << temperatura2 << "°C\n";
```

```
    std::cout << "Cidade: " << cidade3 << " | Temperatura: " << temperatura3 << "°C\n";
```

```
    std::cout << "Cidade: " << cidade4 << " | Temperatura: " << temperatura4 << "°C\n";
```

```
    return 0;
```

```
}
```

# *Entrada de dados*



# *Função scanf*

`std::cin`

Em C++, o objeto `std::cin` (que vem da biblioteca `<iostream>`) é usado para ler valores fornecidos via teclado. Ele é usado junto com o operador de extração de fluxo `>>`.

A forma geral é:

**`std::cin >> variavel;`**

# *Função scanf*

## Exemplo

```
#include <iostream>
```

```
int main() {  
    int idade;
```

```
    // Pede ao usuário para digitar um valor  
    std::cout << "Por favor, digite sua idade: ";
```

```
    // Captura o valor digitado e armazena na variável 'idade'  
    std::cin >> idade;
```

```
    // Exibe o valor capturado  
    std::cout << "Você digitou: " << idade << " anos." << std::endl;
```

```
    return 0;
```

```
}
```

## *Exemplo*

**Escreva um programa para o açougue do seu zé onde o valor do KG da carne moída é R\$29.90 e o frango R\$19.90. Neste programa o usuário deve inserir a quantidade de KG e devera ser impresso o valor a ser pago**

```
#include <iostream>

int main() {
    float carne, frango, resultado;

    std::cout << "Digite a quantidade em KG da carne moida\n";
    std::cin >> carne;

    std::cout << "Digite a quantidade em KG do frango\n";
    std::cin >> frango;

    carne = carne * 29.90;
    frango = frango * 19.90;
    resultado = carne + frango;

    std::cout << "O valor total a ser pago é R$ " << resultado << " reais\n";

    return 0;
}
```

## *Exemplo 3*

**Algoritmo onde o usuário possa entrar com 5 números reais e exiba a media aritmética dos números digitados**



# Exemplo 3

```
#include <iostream>

int main() {
    float n1, n2, n3, n4, n5;
    float soma;
    float media;

    std::cout << "Digite o 1o numero:\n";
    std::cin >> n1;

    std::cout << "Digite o 2o numero:\n";
    std::cin >> n2;

    std::cout << "Digite o 3o numero:\n";
    std::cin >> n3;

    std::cout << "Digite o 4o numero:\n";
    std::cin >> n4;

    std::cout << "Digite o 5o numero:\n";
    std::cin >> n5;

    soma = n1 + n2 + n3 + n4 + n5;

    media = soma / 5.0;

    std::cout << "A media eh: " << media << "\n";

    return 0;
}
```