

Guia Completo de C++ - Do Zero ao Herói

Índice

1. [Introdução e Configuração](#)
 2. [Primeiros Passos](#)
 3. [Variáveis e Tipos](#)
 4. [Entrada e Saída](#)
 5. [Operadores](#)
 6. [Estruturas de Controle](#)
 7. [Loops](#)
 8. [Funções](#)
 9. [Arrays](#)
 10. [Strings](#)
 11. [Ponteiros](#)
 12. [Estruturas \(Structs\)](#)
 13. [Classes e Objetos](#)
 14. [Herança](#)
 15. [STL - Standard Template Library](#)
 16. [Manipulação de Arquivos](#)
 17. [Tratamento de Erros](#)
 18. [Projetos Práticos](#)
-

Introdução {#introdução}

C++ é uma linguagem de programação poderosa, criada por Bjarne Stroustrup em 1979. É uma extensão da linguagem C com recursos de programação orientada a objetos.

Características do C++:

- **Velocidade:** Código compilado roda muito rápido
- **Controle:** Acesso direto à memória e hardware
- **Versatilidade:** Games, sistemas operacionais, aplicações web
- **Orientação a Objetos:** Organização de código em classes e objetos

Configurando o Ambiente

Opções de Compiladores:

- **Windows:** MinGW, Visual Studio, Code::Blocks
 - **Linux:** GCC (já vem instalado na maioria)
 - **Mac:** Xcode Command Line Tools
 - **Online:** cpp.sh, repl.it, ideone.com
-

🌟 Primeiros Passos {#primeiros-passos}

Seu Primeiro Programa

```
cpp

#include <iostream>
using namespace std;

int main() {
    cout << "Olá, Mundo C++!" << endl;
    return 0;
}
```

Explicação linha por linha:

```
cpp

#include <iostream>    // Inclui biblioteca para entrada/saída
using namespace std;  // Evita escrever std:: sempre
int main() {          // Função principal (obrigatória)
    // Seu código aqui
    return 0;          // Retorna 0 = sucesso
}
```

Estrutura Básica de um Programa

cpp

```
// Comentário de uma linha
```

```
/* Comentário  
de múltiplas  
linhas */
```

```
#include <iostream>    // Diretivas de pré-processador  
#include <string>
```

```
using namespace std;    // Declarações de namespace
```

```
// Declarações de funções
```

```
void minhaFuncao();
```

```
int main() {  
    // Código principal  
    cout << "Programa iniciado!" << endl;  
    minhaFuncao();  
    return 0;  
}
```

```
void minhaFuncao() {  
    cout << "Função executada!" << endl;  
}
```

Variáveis e Tipos {#variáveis-e-tipos}

Tipos Fundamentais

cpp

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

int main() {
    // NÚMEROS INTEIROS
    int idade = 25;                // -2 bilhões a +2 bilhões
    short ano = 2024;              // -32,768 a 32,767
    long populacao = 8000000000L;  // Números muito grandes
    unsigned int positivo = 100;   // Apenas positivos

    // NÚMEROS DECIMAIS
    float altura = 1.75f;          // 6-7 dígitos de precisão
    double peso = 70.5;            // 15-17 dígitos de precisão

    // CARACTERES
    char inicial = 'A';            // Um único caractere
    string nome = "João Silva";    // Texto (cadeia de caracteres)

    // BOOLEANOS
    bool estudante = true;         // verdadeiro ou falso
    bool trabalhando = false;

    // MOSTRANDO OS VALORES
    cout << "Nome: " << nome << endl;
    cout << "Idade: " << idade << " anos" << endl;
    cout << "Altura: " << altura << "m" << endl;
    cout << "Peso: " << peso << "kg" << endl;
    cout << "Inicial: " << inicial << endl;
    cout << "É estudante? " << (estudante ? "Sim" : "Não") << endl;

    return 0;
}
```

Constantes

cpp

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

int main() {
    // Constantes não podem ser modificadas
    const double PI = 3.14159;
    const int DIAS_SEMANA = 7;
    const string NOME_PROGRAMA = "Calculadora C++";

    // PI = 3.14; // ERRO! Não pode modificar constante

    cout << "π = " << PI << endl;
    cout << "Dias da semana: " << DIAS_SEMANA << endl;

    return 0;
}
```

Entrada e Saída {#entrada-e-saída}

Saída de Dados (cout)

cpp

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

int main() {
    int numero = 42;
    double preco = 19.99;
    string produto = "Livro C++";

    // Saída básica
    cout << "Produto: " << produto << endl;
    cout << "Preço: R$ " << preco << endl;

    // Formatação de números
    cout << fixed << setprecision(2); // 2 casas decimais
    cout << "Preço formatado: R$ " << preco << endl;

    // Saída sem quebra de linha
    cout << "Digite algo: ";

    return 0;
}
```

Entrada de Dados (cin)

```
cpp

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

int main() {
    string nome;
    int idade;
    double salario;
    char confirmacao;

    // Entrada de dados
    cout << "Nome completo: ";
    getline(cin, nome); // Lê linha inteira (com espaços)

    cout << "Idade: ";
    cin >> idade;

    cout << "Salário: R$ ";
    cin >> salario;

    cout << "Confirma os dados? (s/n): ";
    cin >> confirmacao;

    // Mostrando os dados
    cout << "\n=== DADOS CADASTRADOS ===" << endl;
    cout << "Nome: " << nome << endl;
    cout << "Idade: " << idade << " anos" << endl;
    cout << "Salário: R$ " << fixed << setprecision(2) << salario << endl;
    cout << "Status: " << (confirmacao == 's' ? "Confirmado" : "Pendente") << endl;

    return 0;
}
```

Operadores {#operadores}

Operadores Aritméticos

cpp

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

int main() {
    int a = 10, b = 3;

    cout << "a = " << a << ", b = " << b << endl;
    cout << "a + b = " << (a + b) << endl; // Soma: 13
    cout << "a - b = " << (a - b) << endl; // Subtração: 7
    cout << "a * b = " << (a * b) << endl; // Multiplicação: 30
    cout << "a / b = " << (a / b) << endl; // Divisão inteira: 3
    cout << "a % b = " << (a % b) << endl; // Resto: 1

    // Divisão real
    double c = 10.0, d = 3.0;
    cout << "c / d = " << (c / d) << endl; // 3.33333

    // Potência
    cout << "a^b = " << pow(a, b) << endl; // 103 = 1000

    // Incremento e decremento
    int x = 5;
    cout << "x = " << x << endl; // 5
    cout << "x++ = " << x++ << endl; // 5 (mostra depois incrementa)
    cout << "x = " << x << endl; // 6
    cout << "++x = " << ++x << endl; // 7 (incrementa depois mostra)

    return 0;
}
```

Operadores de Comparação

cpp

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

int main() {
    int x = 10, y = 20;

    cout << "x = " << x << ", y = " << y << endl;
    cout << "x == y: " << (x == y) << endl; // igual: 0 (falso)
    cout << "x != y: " << (x != y) << endl; // diferente: 1 (verdadeiro)
    cout << "x < y: " << (x < y) << endl; // menor: 1
    cout << "x > y: " << (x > y) << endl; // maior: 0
    cout << "x <= y: " << (x <= y) << endl; // menor ou igual: 1
    cout << "x >= y: " << (x >= y) << endl; // maior ou igual: 0

    return 0;
}
```

Operadores Lógicos

cpp

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

int main() {
    bool a = true, b = false;

    cout << "a = " << a << ", b = " << b << endl;
    cout << "a && b: " << (a && b) << endl; // AND: 0
    cout << "a || b: " << (a || b) << endl; // OR: 1
    cout << "!a: " << (!a) << endl; // NOT: 0

    // Exemplo prático
    int idade = 25;
    bool temCNH = true;
    bool podeAlugarCarro = (idade >= 21) && temCNH;

    cout << "Pode alugar carro? " << (podeAlugarCarro ? "Sim" : "Não") << endl;

    return 0;
}
```

Estruturas de Controle {#estruturas-de-controle}

If, Else If, Else

cpp

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

int main() {
    double nota;

    cout << "Digite sua nota (0-10): ";
    cin >> nota;

    if (nota >= 9.0) {
        cout << "🏆 Excelente! Parabéns!" << endl;
    } else if (nota >= 7.0) {
        cout << "😊 Bom trabalho!" << endl;
    } else if (nota >= 5.0) {
        cout << "😐 Razoável, pode melhorar." << endl;
    } else if (nota >= 0.0) {
        cout << "😞 Precisa estudar mais." << endl;
    } else {
        cout << "❌ Nota inválida!" << endl;
    }

    return 0;
}
```

Switch Case

cpp

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

int main() {
    int opcao;

    cout << "=== CALCULADORA ===" << endl;
    cout << "1. Somar" << endl;
    cout << "2. Subtrair" << endl;
    cout << "3. Multiplicar" << endl;
    cout << "4. Dividir" << endl;
    cout << "Escolha uma opção: ";
    cin >> opcao;

    double num1, num2, resultado;
    cout << "Digite dois números: ";
    cin >> num1 >> num2;

    switch (opcao) {
        case 1:
            resultado = num1 + num2;
            cout << "Soma: " << resultado << endl;
            break;

        case 2:
            resultado = num1 - num2;
            cout << "Subtração: " << resultado << endl;
            break;

        case 3:
            resultado = num1 * num2;
            cout << "Multiplicação: " << resultado << endl;
            break;

        case 4:
            if (num2 != 0) {
                resultado = num1 / num2;
                cout << "Divisão: " << resultado << endl;
            } else {
                cout << "❌ Erro: Divisão por zero!" << endl;
            }
            break;

        default:
            cout << "❌ Opção inválida!" << endl;
    }

    return 0;
```

```
return 0;
```

```
}
```

Operador Ternário

cpp

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int idade;
```

```
    cout << "Digite sua idade: ";
```

```
    cin >> idade;
```

```
    // Sintaxe: condição ? valor_se_verdadeiro : valor_se_falso
```

```
    string categoria = (idade >= 18) ? "Adulto" : "Menor";
```

```
    string permissao = (idade >= 21) ? "Pode beber" : "Não pode beber";
```

```
    cout << "Categoria: " << categoria << endl;
```

```
    cout << "Permissão: " << permissao << endl;
```

```
    // Aninhando operadores ternários
```

```
    string faixa = (idade < 13) ? "Criança" :
```

```
                  (idade < 18) ? "Adolescente" :
```

```
                  (idade < 60) ? "Adulto" : "Idoso";
```

```
    cout << "Faixa etária: " << faixa << endl;
```

```
    return 0;
```

```
}
```

Loops {#loops}

Loop For

cpp

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

int main() {
    // Loop básico
    cout << "Contando de 1 a 10:" << endl;
    for (int i = 1; i <= 10; i++) {
        cout << i << " ";
    }
    cout << endl;

    // Loop regressivo
    cout << "Contagem regressiva:" << endl;
    for (int i = 10; i >= 1; i--) {
        cout << i << " ";
    }
    cout << "🚀 Decolagem!" << endl;

    // Loop com incremento diferente
    cout << "Números pares de 0 a 20:" << endl;
    for (int i = 0; i <= 20; i += 2) {
        cout << i << " ";
    }
    cout << endl;

    // Tabuada
    int numero;
    cout << "Digite um número para a tabuada: ";
    cin >> numero;

    cout << "\nTabuada do " << numero << ":" << endl;
    for (int i = 1; i <= 10; i++) {
        cout << numero << " x " << i << " = " << (numero * i) << endl;
    }

    return 0;
}
```

Loop While

cpp

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

int main() {
    // Jogo de adivinhação
    int numeroSecreto = 42;
    int tentativa;
    int tentativas = 0;

    cout << "🎯 Jogo de Adivinhação!" << endl;
    cout << "Adivinhe o número entre 1 e 100:" << endl;

    while (true) {
        cout << "Sua tentativa: ";
        cin >> tentativa;
        tentativas++;

        if (tentativa == numeroSecreto) {
            cout << "🎉 Parabéns! Você acertou em " << tentativas << " tentativas!" <<
                break;
        } else if (tentativa < numeroSecreto) {
            cout << "📉 Muito baixo! Tente um número maior." << endl;
        } else {
            cout << "📈 Muito alto! Tente um número menor." << endl;
        }

        if (tentativas >= 7) {
            cout << "😓 Muitas tentativas! O número era " << numeroSecreto << endl;
            break;
        }
    }

    return 0;
}
```

Loop Do-While

cpp

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

int main() {
    char continuar;
    int soma = 0;
    int numero;
    int contador = 0;

    cout << "📊 Calculadora de Média" << endl;

    do {
        cout << "Digite um número: ";
        cin >> numero;
        soma += numero;
        contador++;

        cout << "Deseja continuar? (s/n): ";
        cin >> continuar;

    } while (continuar == 's' || continuar == 'S');

    if (contador > 0) {
        double media = static_cast<double>(soma) / contador;
        cout << "Soma: " << soma << endl;
        cout << "Quantidade: " << contador << endl;
        cout << "Média: " << media << endl;
    }

    return 0;
}
```

Controle de Loops (break e continue)

cpp

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

int main() {
    cout << "Números de 1 a 20 (pulando múltiplos de 3):" << endl;

    for (int i = 1; i <= 20; i++) {
        // Pula múltiplos de 3
        if (i % 3 == 0) {
            continue; // Pula para próxima iteração
        }

        cout << i << " ";

        // Para no 15
        if (i >= 15) {
            break; // Sai do loop
        }
    }
    cout << endl;

    // Exemplo com loops aninhados
    cout << "\nTabela de multiplicação (1 a 5):" << endl;

    for (int i = 1; i <= 5; i++) {
        for (int j = 1; j <= 5; j++) {
            cout << (i * j) << "\t";
        }
        cout << endl;
    }

    return 0;
}
```

Funções {#funções}

Funções Básicas

cpp

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

// Função sem parâmetros e sem retorno
void saudar() {
    cout << "Olá! Bem-vindo ao programa!" << endl;
}

// Função com parâmetros e sem retorno
void saudarPessoa(string nome, int idade) {
    cout << "Olá, " << nome << "! Você tem " << idade << " anos." << endl;
}

// Função com parâmetros e com retorno
int somar(int a, int b) {
    return a + b;
}

// Função com parâmetros padrão
double calcularArea(double raio, double pi = 3.14159) {
    return pi * raio * raio;
}

int main() {
    saudar();
    saudarPessoa("João", 25);

    int resultado = somar(10, 20);
    cout << "10 + 20 = " << resultado << endl;

    double area = calcularArea(5.0);
    cout << "Área do círculo: " << area << endl;

    return 0;
}
```

Sobrecarga de Funções

cpp

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

// Diferentes versões da mesma função
int somar(int a, int b) {
    cout << "Somando inteiros: ";
    return a + b;
}

double somar(double a, double b) {
    cout << "Somando decimais: ";
    return a + b;
}

int somar(int a, int b, int c) {
    cout << "Somando três inteiros: ";
    return a + b + c;
}

string somar(string a, string b) {
    cout << "Concatenando strings: ";
    return a + b;
}

int main() {
    cout << somar(5, 3) << endl;
    cout << somar(5.5, 3.2) << endl;
    cout << somar(1, 2, 3) << endl;
    cout << somar("Olá, ", "Mundo!") << endl;

    return 0;
}
```

Passagem por Referência

cpp

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

// Passagem por valor (cópia)
void trocarPorValor(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
    cout << "Dentro da função - a: " << a << ", b: " << b << endl;
}

// Passagem por referência (original)
void trocarPorReferencia(int &a, int &b) {
    int temp = a;
    a = b;
    b = temp;
    cout << "Dentro da função - a: " << a << ", b: " << b << endl;
}

// Função para calcular múltiplos valores
void calcularCirculo(double raio, double &area, double &perimetro) {
    const double PI = 3.14159;
    area = PI * raio * raio;
    perimetro = 2 * PI * raio;
}

int main() {
    int x = 10, y = 20;

    cout << "Antes - x: " << x << ", y: " << y << endl;

    trocarPorValor(x, y);
    cout << "Após valor - x: " << x << ", y: " << y << endl;

    trocarPorReferencia(x, y);
    cout << "Após referência - x: " << x << ", y: " << y << endl;

    // Calculando área e perímetro
    double area, perimetro;
    calcularCirculo(5.0, area, perimetro);
    cout << "Área: " << area << ", Perímetro: " << perimetro << endl;

    return 0;
}
```

Funções Recursivas

cpp

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

// Fatorial
long long fatorial(int n) {
    if (n <= 1) {
        return 1;
    }
    return n * fatorial(n - 1);
}

// Fibonacci
long long fibonacci(int n) {
    if (n <= 1) {
        return n;
    }
    return fibonacci(n - 1) + fibonacci(n - 2);
}

// Potência
double potencia(double base, int expoente) {
    if (expoente == 0) {
        return 1;
    }
    if (expoente > 0) {
        return base * potencia(base, expoente - 1);
    } else {
        return 1.0 / potencia(base, -expoente);
    }
}

int main() {
    int n;

    cout << "Digite um número: ";
    cin >> n;

    cout << "Fatorial de " << n << " = " << fatorial(n) << endl;

    cout << "Sequência de Fibonacci até " << n << ":" << endl;
    for (int i = 0; i < n; i++) {
        cout << fibonacci(i) << " ";
    }
    cout << endl;

    cout << "2^" << n << " = " << potencia(2, n) << endl;

    return 0;
}
```

```
return 0;
```

```
}
```

Arrays {#arrays}

Arrays Unidimensionais

cpp

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

int main() {
    // Declaração e inicialização
    int numeros[5] = {10, 20, 30, 40, 50};
    string frutas[3] = {"Maçã", "Banana", "Laranja"};

    // Acessando elementos
    cout << "Primeiro número: " << numeros[0] << endl;
    cout << "Última fruta: " << frutas[2] << endl;

    // Modificando elementos
    numeros[1] = 25;
    frutas[1] = "Manga";

    // Percorrendo o array
    cout << "Números: ";
    for (int i = 0; i < 5; i++) {
        cout << numeros[i] << " ";
    }
    cout << endl;

    // Usando range-based for (C++11)
    cout << "Frutas: ";
    for (const string& fruta : frutas) {
        cout << fruta << " ";
    }
    cout << endl;

    // Calculando média
    double notas[4];
    double soma = 0;

    cout << "Digite 4 notas:" << endl;
    for (int i = 0; i < 4; i++) {
        cout << "Nota " << (i + 1) << ": ";
        cin >> notas[i];
        soma += notas[i];
    }

    double media = soma / 4;
    cout << "Média: " << media << endl;

    return 0;
}
```

Arrays Multidimensionais

```
cpp

#include <iostream>
using namespace std;

int main() {
    // Matriz 3x3
    int matriz[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };

    // Exibindo a matriz
    cout << "Matriz 3x3:" << endl;
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << matriz[i][j] << " ";
        }
        cout << endl;
    }

    // Jogo da velha
    char tabuleiro[3][3];

    // Inicializando o tabuleiro
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            tabuleiro[i][j] = ' ';
        }
    }

    // Colocando algumas jogadas
    tabuleiro[0][0] = 'X';
    tabuleiro[1][1] = 'O';
    tabuleiro[2][2] = 'X';

    // Exibindo o tabuleiro
    cout << "\nTabuleiro do Jogo da Velha:" << endl;
    for (int i = 0; i < 3; i++) {
        cout << " " << tabuleiro[i][0] << " | " << tabuleiro[i][1] << " | " << tabulei
        if (i < 2) cout << "---|---|---" << endl;
    }

    return 0;
}
```


cpp

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

```
// Função para imprimir array
```

```
void imprimirArray(int arr[], int tamanho) {
    for (int i = 0; i < tamanho; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}
```

```
// Função para encontrar o maior elemento
```

```
int encontrarMaior(int arr[], int tamanho) {
    int maior = arr[0];
    for (int i = 1; i < tamanho; i++) {
        if (arr[i] > maior) {
            maior = arr[i];
        }
    }
    return maior;
}
```

```
// Função para ordenar array (Bubble Sort)
```

```
void ordenarArray(int arr[], int tamanho) {
    for (int i = 0; i < tamanho - 1; i++) {
        for (int j = 0; j < tamanho - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                // Trocar elementos
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}
```

```
// Função para buscar elemento
```

```
int buscarElemento(int arr[], int tamanho, int elemento) {
    for (int i = 0; i < tamanho; i++) {
        if (arr[i] == elemento) {
            return i; // Retorna o índice
        }
    }
    return -1; // Não encontrado
}
```

```
int main() {
```

```
    int numeros[10] = {64, 24, 25, 12, 22, 11, 99, 5, 77, 20};
```



```

int numeros[10] = {64, 34, 25, 12, 22, 11, 90, 5, 77, 30};
int tamanho = 10;

cout << "Array original: ";
imprimirArray(numeros, tamanho);

cout << "Maior elemento: " << encontrarMaior(numeros, tamanho) << endl;

int busca = 22;
int posicao = buscarElemento(numeros, tamanho, busca);
if (posicao != -1) {
    cout << "Elemento " << busca << " encontrado na posição " << posicao << endl;
} else {
    cout << "Elemento " << busca << " não encontrado" << endl;
}

ordenarArray(numeros, tamanho);
cout << "Array ordenado: ";
imprimirArray(numeros, tamanho);

return 0;
}

```

Strings {#strings}

Operações com Strings

cpp

```

#include <iostream>
#include <string>
#include <cctype>
using namespace std;

int main() {
    string nome = "João Silva";
    string sobrenome = "Santos";

    // Concatenação
    string nomeCompleto = nome + " " + sobrenome;
    cout << "Nome completo: " << nomeCompleto << endl;

    // Comprimento
    cout << "Tamanho: " << nomeCompleto.length() << " caracteres" << endl;

    // Acessando caracteres
    cout << "Primeira letra: " << nomeCompleto[0] << endl;
    cout << "Última letra: "

```

