

Marcos Antônio Pereira Simões

Specialist in Teaching for Professional Education – SENAC - 2012

Specialist in Information Technology Governance – SENAC – 2010

Technologist in Data Processing – Estácio de Sá - 2007

E-mail:

marcos.matriz@notacontrol.com.br



Analista e Desenvolvedor de Sistemas

Dinâmica vitalizadora: “Rodada de Entrevistas”

Lógica de Programação



Lógica de Programação

Algoritmo

Sequência finita de instruções para se resolver um problema.

* aplica-se a diversas áreas de conhecimento

Lógica de Programação

Exemplo:

Problema: lavar roupa suja

Algoritmo:

- 1) Colocar a roupa em um recipiente
- 2) Colocar um pouco de sabão e amaciante
- 3) Encher de água
- 4) Mexer tudo até dissolver todo o sabão
- 5) Deixar de molho por vinte minutos
- 6) Esfregar a roupa
- 7) Enxaguar
- 8) Torcer

Lógica de Programação

Automação

Consiste em utilizar máquina(s) para executar o procedimento desejado de forma automática ou semiautomática.

Lógica de Programação

Algoritmo:

- 1) Colocar a roupa em um recipiente
- 2) Colocar um pouco de sabão e amaciante
- 3) Encher de água
- 4) Mexer tudo até dissolver todo o sabão
- 5) Deixar de molho por vinte minutos
- 6) Esfregar a roupa
- 7) Enxaguar
- 8) Torcer



Lógica de Programação

Mas o que algoritmo e automação tem a ver com programação de computadores?

Lógica de Programação

Mas o que algoritmo e
automação tem a ver com
programação de computadores?

Lógica de Programação

Computador

- Hardware - parte física (a máquina em si)
- Software - parte lógica (programas)
 - Sistema operacional (Windows, Linux, iOS)
 - Aplicativos (apps de escritório, app de câmera, navegador web)
 - Jogos
 - Utilitários (Antivírus, compactador de arquivos)
 - Outros



Lógica de Programação

Programas & Algoritmos

Programas de computador **são algoritmos** executados pelo computador (em linhas gerais).

Conclusão: o computador é uma máquina que **automatiza** a execução de **algoritmos**.

Qualquer algoritmo? Não. Apenas algoritmos computacionais:

- Processamento de dados
- Cálculos

Lógica de Programação

O que é preciso para se fazer um programa de computador?

Lógica de Programação

Vamos precisar de:

- Uma **linguagem de programação**: regras **léxicas** e **sintáticas** para se escrever o programa
- Uma **IDE**: software para editar e testar o programa
- Um **compilador**: software para transformar o **código fonte** em **código objeto**
- Um **gerador de código** ou **máquina virtual**: software que permite que o programa seja executado

Lógica de Programação

Linguagem de programação

É um conjunto de regras **léxicas** (ortografia) e **sintáticas** (gramática) para se escrever programas.

Lógica de Programação

Léxica

Diz respeito à correção das **palavras** "isoladas" (ortografia).

Exemplo (Português):

cachorro

caxorro



Linguagem de programação:

main

maim



Lógica de Programação

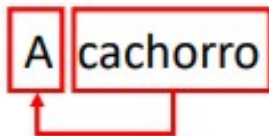
Sintática

Diz respeito à correção das **sentenças** (gramática).

Exemplo (Português):

O cachorro está com fome.

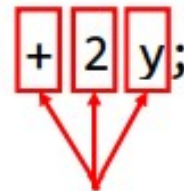
A cachorro está com fome.



Linguagem de programação:

$x = 2 + y;$

$x = + 2 y;$



Lógica de Programação

Linguagem de programação

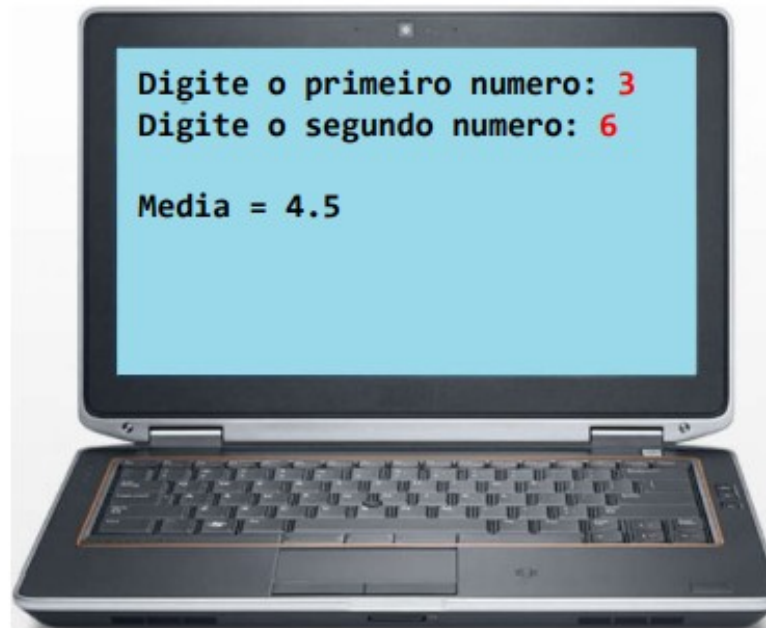
Exemplos de linguagens de programação:

C, Pascal, C++, Java, C#, Python, Ruby, PHP, JavaScript, etc.

Lógica de Programação

Exemplo de um programa:

Suponha um programa que solicita do usuário dois números e depois mostra a média aritmética deles:



Lógica de Programação

Solução em linguagem C

```
#include <stdio.h>

int main() {
    double x, y, media;

    printf("Digite o primeiro numero: ");
    scanf("%lf", &x);
    printf("Digite o segundo numero: ");
    scanf("%lf", &y);
    media = (x + y) / 2.0;
    printf("Media = %.1f\n", media);
    return 0;
}
```

Lógica de Programação

Solução em linguagem C++

```
#include <iostream>

using namespace std;

int main() {
    double x, y, media;

    cout << "Digite o primeiro numero: ";
    cin >> x;
    cout << "Digite o segundo numero: ";
    cin >> y;
    media = (x + y) / 2.0;
    cout << "Media = " << media << endl;
    return 0;
}
```

Lógica de Programação

Solução em linguagem C#

```
using System;

namespace programa {
    class Program {
        static void Main(string[] args) {
            double x, y, media;

            Console.Write("Digite o primeiro numero: ");
            x = double.Parse(Console.ReadLine());
            Console.Write("Digite o segundo numero: ");
            y = double.Parse(Console.ReadLine());
            media = (x + y) / 2.0;
            Console.WriteLine("Media = " + media);
        }
    }
}
```

Lógica de Programação

Solução em linguagem Java

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double x, y, media;

        System.out.print("Digite o primeiro numero: ");
        x = sc.nextDouble();
        System.out.print("Digite o segundo numero: ");
        y = sc.nextDouble();
        media = (x + y) / 2.0;
        System.out.println("Media = " + media);
        sc.close();
    }
}
```

Lógica de Programação

**IDE – Ambiente Integrado de
Desenvolvimento**

Lógica de Programação

IDE – Ambiente Integrado de Desenvolvimento

É um conjunto de softwares utilizado para a construção de programas.

Exemplos:

C/C++ : **Code Blocks**

Java : **Eclipse, NetBeans**

C# : **Microsoft Visual Studio**

Lógica de Programação

Funcionalidades de uma IDE

- Edição de código fonte (endentação, autocompletar, destaque de palavras, etc.)
- Depuração e testes
- Construção do produto final (build)
- Sugestão de modelos (templates)
- Auxiliar em várias tarefas do seu projeto
- Etc.

Lógica de Programação

Compilação e Interpretação
Código Fonte e Objeto
Máquina Virtual

Lógica de Programação

Código fonte: é aquele escrito pelo programador em linguagem de programação

```
#include <stdio.h>

int main() {
    double x, y, media;

    printf("Digite o primeiro numero: ");
    scanf("%lf", &x);
    printf("Digite o segundo numero: ");
    scanf("%lf", &y);
    media = (x + y) / 2.0;
    printf("Media = %.1f\n", media);
    return 0;
}
```

```
#include <iostream>

using namespace std;

int main() {
    double x, y, media;

    cout << "Digite o primeiro numero: ";
    cin >> x;
    cout << "Digite o segundo numero: ";
    cin >> y;
    media = (x + y) / 2.0;
    cout << "Media = " << media << endl;
    return 0;
}
```

```
using System;

namespace programa {
    class Program {
        static void Main(string[] args) {
            double x, y, media;

            Console.Write("Digite o primeiro numero: ");
            x = double.Parse(Console.ReadLine());
            Console.Write("Digite o segundo numero: ");
            y = double.Parse(Console.ReadLine());
            media = (x + y) / 2.0;
            Console.WriteLine("Media = " + media);
        }
    }
}
```

```
import java.util.Scanner;

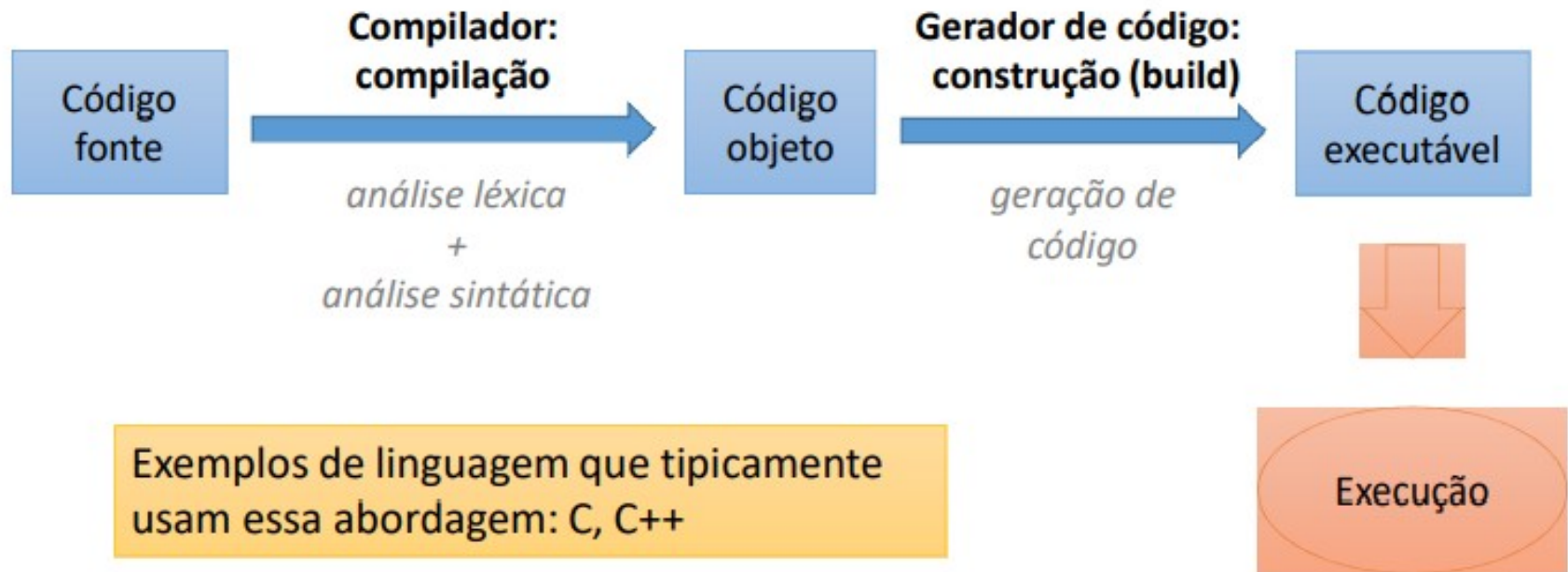
public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double x, y, media;

        System.out.print("Digite o primeiro numero: ");
        x = sc.nextDouble();
        System.out.print("Digite o segundo numero: ");
        y = sc.nextDouble();
        media = (x + y) / 2.0;
        System.out.println("Media = " + media);
        sc.close();
    }
}
```

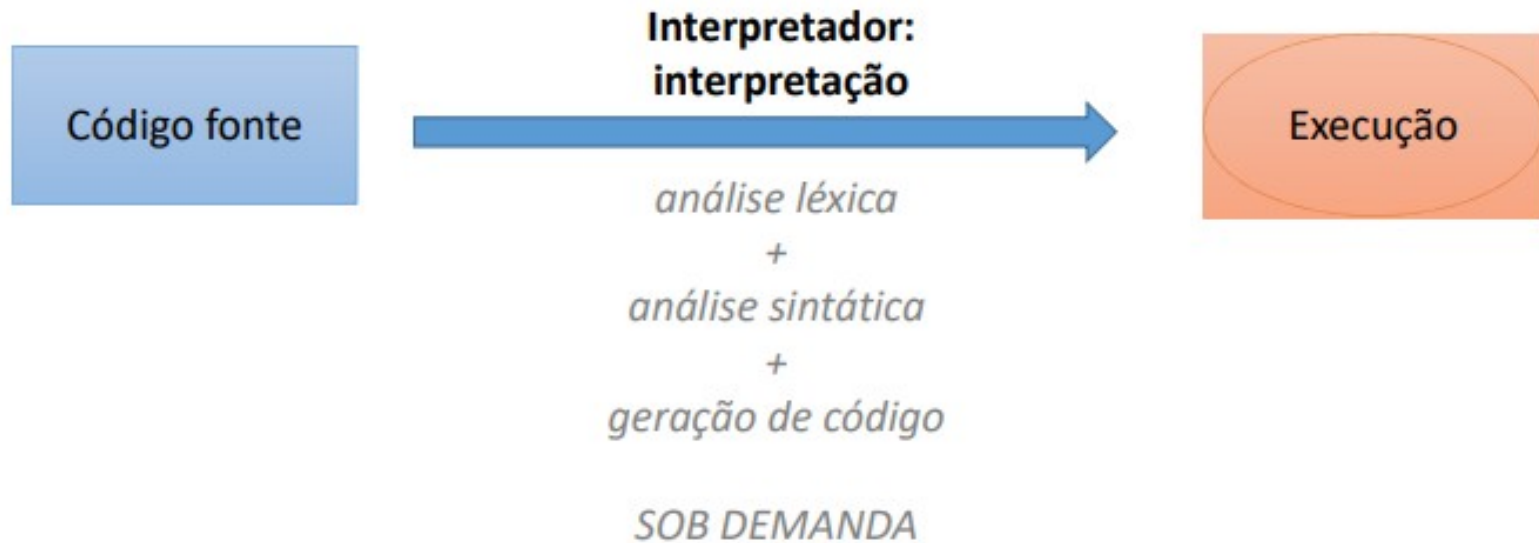
Lógica de Programação

Compilação



Lógica de Programação

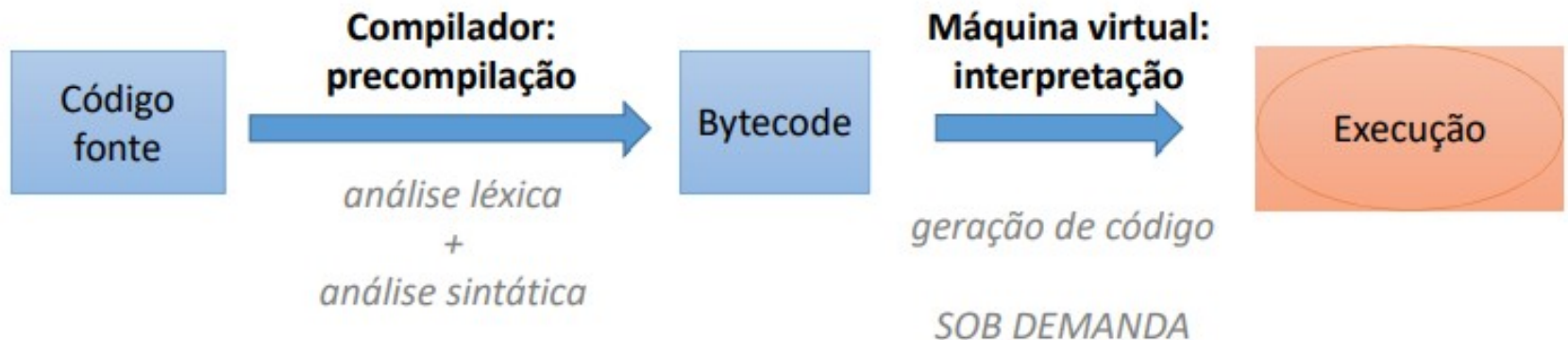
Interpretação



Exemplos de linguagem que tipicamente usam essa abordagem: PHP, JavaScript, Python, Ruby

Lógica de Programação

Abordagem híbrida



Exemplos de linguagem que tipicamente usam essa abordagem: Java (JVM), C# (Microsoft .NET Framework)

Lógica de Programação

Vantagens

Compilação:

- velocidade do programa
- auxílio do compilador antes da execução

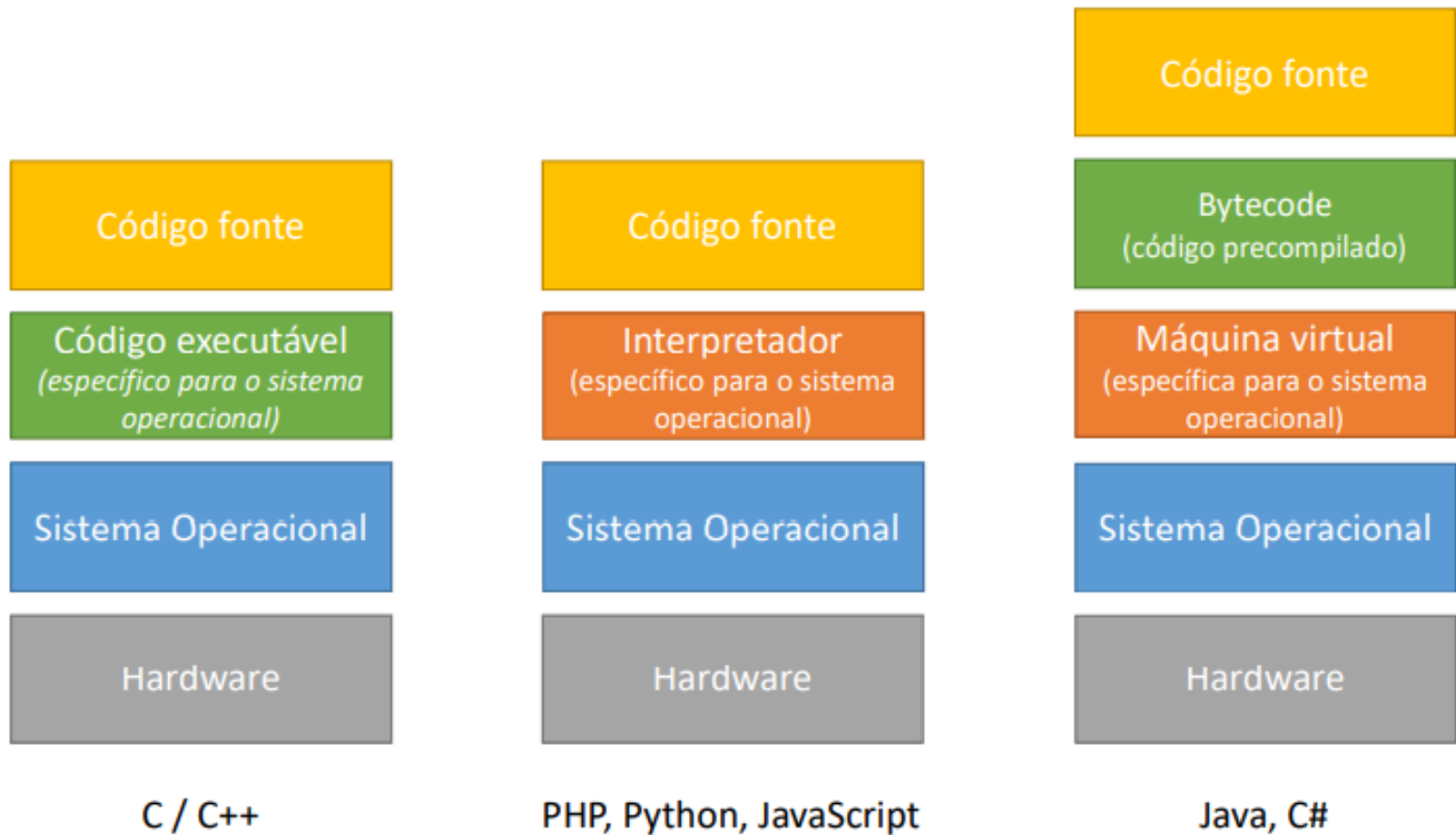
Interpretação:

- flexibilidade de manutenção do aplicativo em produção
- expressividade da linguagem
- código fonte não precisa ser recompilado para rodar em plataformas diferentes

Abordagem híbrida



Lógica de Programação

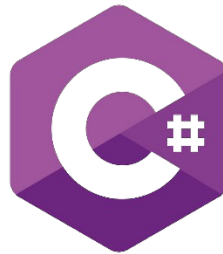


Lógica de Programação

- Visual Studio Community



- Linguagem: C#



Lógica de Programação

Instalação do Visual Studio

<https://docs.microsoft.com/pt-br/visualstudio/install/install-visual-studio?view=vs-2022>

Lógica de Programação

Cursos:

<https://www.ev.org.br/areas-de-interesse/programacao>

<https://www.udemy.com/>

<https://www.alura.com.br/>

Lógica de Programação

Primeiro Programa em C#

- * Aplicativo de Console

- * Dicas

Indentação automática: CTRL + K + D

Fonte: CTRL + Mouse Scroll

Rodar o projeto: CTRL + F5

Lógica de Programação

* Dicas

Alterar o tema de cores do quadro IDE e as janelas de ferramentas no Visual Studio:

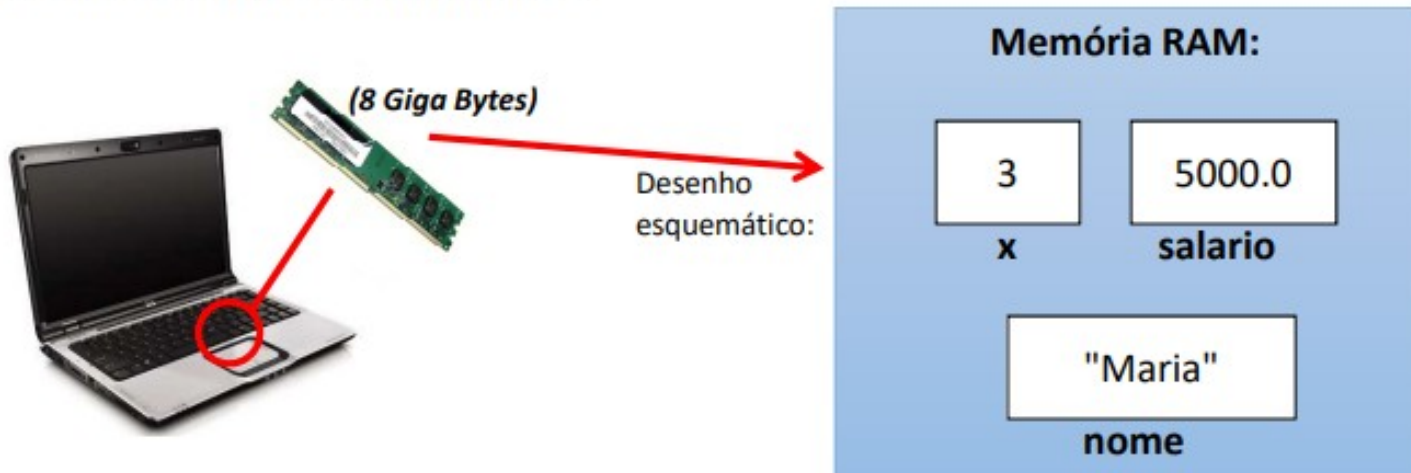
- Na barra de menus, escolha Opções de Ferramentas>.
- Na lista de opções, escolha Ambiente > Geral.
- Na lista de temas De cor, escolha o tema escuro padrão, o tema Claro, o tema Azul ou o tema Azul (Contraste Extra).

Lógica de Programação

Variáveis

Definição informal:

Em programação, uma variável é uma porção de memória (RAM) utilizada para armazenar dados durante a execução dos programas.



Lógica de Programação

Tipos básicos em C#

byte	0 .. 255
sbyte	-128 .. 127
short	-32,768 .. 32,767
ushort	0 .. 65,535
int	-2,147,483,648 .. 2,147,483,647
uint	0 .. 4,294,967,295
long	-9,223,372,036,854,775,808 .. 9,223,372,036,854,775,807
ulong	0 .. 18,446,744,073,709,551,615

Lógica de Programação

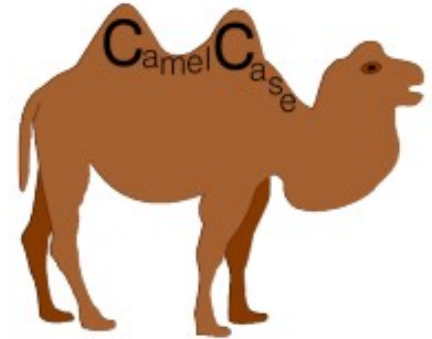
Tipos básicos em C#

float	-3.402823e38 .. 3.402823e38
double	-1.79769313486232e308 .. 1.79769313486232e308
decimal	-79228162514264337593543950335 .. 79228162514264337593543950335
char	A Unicode character.
string	A string of Unicode characters.
bool	True or False.
object	An object.

Lógica de Programação

Nomes de variáveis

- Não pode começar com dígito: use uma letra ou _
- Não pode ter espaço em branco
- Não usar acentos ou til
- Sugestão: use o padrão "camel case"



Errado:

```
int 5minutos;  
int salário;  
int salário do funcionario;
```

Correto:

```
int _5minutos;  
int salario;  
int salarioDoFuncionario;
```

Lógica de Programação

Um programa de computador é capaz de realizar essencialmente três operações:



Lógica de Programação

Para escrever na tela um texto qualquer

Sem quebra de linha ao final:

```
Console.Write("Bom dia!");
```

Com quebra de linha ao final:

```
Console.WriteLine("Bom dia!");
```

Lógica de Programação

Para escrever o conteúdo de uma variável com ponto flutuante

Suponha uma variável tipo **double** declarada e iniciada:

```
double x = 10.35784;
```

```
Console.WriteLine(x);
```

```
Console.WriteLine(x.ToString("F2"));
```

```
Console.WriteLine(x.ToString("F4"));
```

```
Console.WriteLine(x.ToString("F2", CultureInfo.InvariantCulture));
```


using System.Globaliztion;

Lógica de Programação

Para concatenar vários elementos em um mesmo comando de escrita

Regra geral:

```
elemento1 + elemento2 + elemento3 + ... + elementoN
```

```
Console.WriteLine("RESULTADO = " + x);
```

```
Console.WriteLine("O valor do troco é " + x + " reais");
```

```
Console.WriteLine("O valor do troco é " + x.ToString("F2") + " reais");
```

Lógica de Programação

```
using System;
using System.Globalization;

namespace Curso {
    class Program {
        static void Main(string[] args) {

            double x = 10.35784;
            int y = 32;
            string z = "Maria";
            char w = 'F';

            Console.Write("Olá mundo!");
            Console.WriteLine("Bom dia!");
            Console.WriteLine("Até mais!");

            Console.WriteLine();
            Console.WriteLine(x);
            Console.WriteLine(x.ToString("F2"));
            Console.WriteLine(x.ToString("F4"));
            Console.WriteLine(x.ToString("F2", CultureInfo.InvariantCulture));

            Console.WriteLine();
            Console.WriteLine("RESULTADO = " + x);
            Console.WriteLine("O valor do troco é " + x + " reais");
            Console.WriteLine("O valor do troco é " + x.ToString("F2") + " reais");

            Console.WriteLine();
            Console.WriteLine("A paciente " + z + " tem " + y + " anos e seu sexo é: " + w);

            Console.ReadLine();

        }
    }
}
```


Lógica de Programação

Processamento de dados

Comando de atribuição.

Sintaxe:

<variável> = <expressão>;



Lê-se "recebe"

REGRA:

- 1) A expressão é calculada
- 2) O resultado da expressão é armazenado na variável

A expressão numérica deve usar as operações de adição (+), subtração (-) , divisão (/) e multiplicação (*).

Lógica de Programação

Exemplo 1

```
int x, y;
```

```
x = 5;
```

```
y = 2 * x;
```

```
Console.WriteLine(x);
```

```
Console.WriteLine(y);
```

Lógica de Programação

Exemplo 2

```
int x;  
double y;
```

```
x = 5;
```

```
y = 2 * x;
```

```
Console.WriteLine(x);  
Console.WriteLine(y);
```

Lógica de Programação

$$((2 + 33 + 3) * 2) + (1400 - 100) * 3$$

Crie uma variável chamada valor que receba o resultado acima;

Imprima este valor no seguinte formato:
Valor = xxxxx

Lógica de Programação

Exemplo 3

```
double b, B, h, area;
```

```
b = 6.0;
```

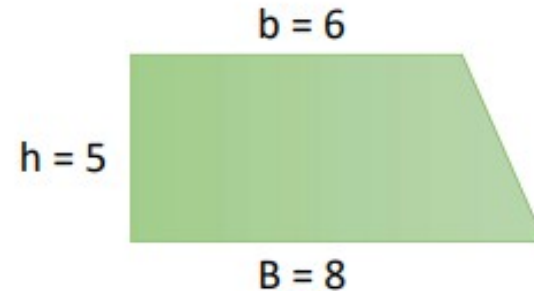
```
B = 8.0;
```

```
h = 5.0;
```

```
area = (b + B) / 2.0 * h;
```

```
Console.WriteLine(area);
```

Área do trapézio



$$area = \frac{(b + B)}{2} \times h$$

No exemplo:

$$\begin{aligned} area &= \frac{(6 + 8)}{2} \times 5 \\ &= \frac{14}{2} \times 5 = 7 \times 5 = 35 \end{aligned}$$

Lógica de Programação

Exemplo 4

```
int a, b;  
double resultado;  
  
a = 5;  
b = 2;  
  
resultado = a / b;  
  
Console.WriteLine(resultado);
```

Lógica de Programação

Exemplo 4

```
int a, b;  
double resultado;
```

```
a = 5;  
b = 2;
```

```
resultado = a / b;
```

```
Console.WriteLine(resultado);
```

Casting

É a conversão explícita de um tipo para outro.

É necessário quando o compilador não é capaz de “adivinhar” que o resultado de uma expressão deve ser de outro tipo.



```
resultado = (double) a / b;
```

Lógica de Programação

Exemplo 5

```
double a;  
int b;
```

```
a = 5.0;  
b = a;
```

```
Console.WriteLine(b);
```

Lógica de Programação

Plataforma BEECROWD antiga URI

<https://www.beecrowd.com.br/judge/pt/login?redirect=%2Fpt>

Lógica de Programação

Soma Simples

Leia dois valores inteiros, no caso para variáveis A e B. A seguir, calcule a soma entre elas e atribua à variável **SOMA**. A seguir escreva o valor desta variável.

Entrada

O arquivo de entrada contém 2 valores inteiros.

Saída

Imprima a mensagem "SOMA" com todas as letras maiúsculas, com um espaço em branco antes e depois da igualdade seguido pelo valor correspondente à soma de A e B. Como todos os problemas, não esqueça de imprimir o fim de linha após o resultado, caso contrário, você receberá "Presentation Error".

Exemplos de Entrada	Exemplos de Saída
30 10	SOMA = 40
-30 10	SOMA = -20
0 0	SOMA = 0

Lógica de Programação

Produto Simples

Leia dois valores inteiros. A seguir, calcule o produto entre estes dois valores e atribua esta operação à variável **PROD**. A seguir mostre a variável **PROD** com mensagem correspondente.

Entrada

O arquivo de entrada contém 2 valores inteiros.

Saída

Imprima a mensagem "PROD" e a variável **PROD** conforme exemplo abaixo, com um espaço em branco antes e depois da igualdade. Não esqueça de imprimir o fim de linha após o produto, caso contrário seu programa apresentará a mensagem: *"Presentation Error"*.

Exemplos de Entrada	Exemplos de Saída
3 9	PROD = 27
-30 10	PROD = -300
0 9	PROD = 0