

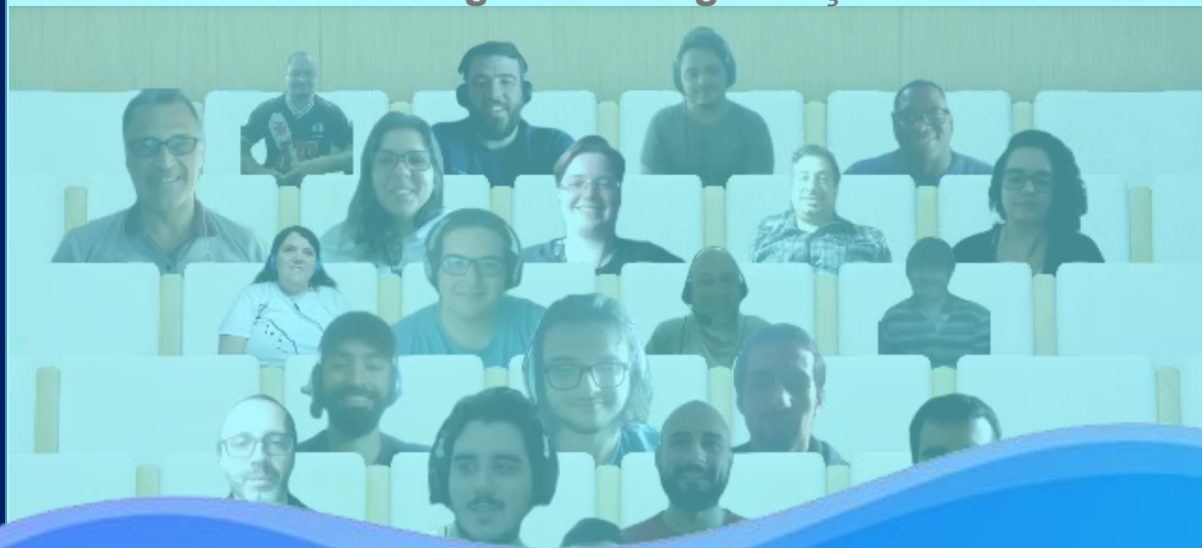
53 45 52 45 49 20 46 49 45 4c 20  
41 4f 53 20 50 52 45 43 45 49 54  
4f 53 20 44 41 20 48 4f 4e 52 41  
20 45 20 44 41 20 43 49 c3 8a 4e  
43 49 41 2c 20 50 52 4f 4d 4f 56  
45 4e 44 4f 20 4f 20 55 53 4f 20  
45 20 4f 20 44 45 53 45 4e 56 4f  
4c 56 49 4d 45 4e 54 4f 20 44 41  
20 49 4e 46 4f 52 4d c3 81 54 49  
43 41 20 45 4d 20 42 45 4e 45 46  
c3 8d 43 49 4f 20 44 4f 20 43 49  
44 41 44 c3 83 4f 20 45 20 44 41  
20 53 4f 43 49 45 44 41 44 45 2e

## RESIDÊNCIA DE SOFTWARE

**CAPACITAR**  
**TREINAR**  
**EMPREGAR**  
**TRANSFORMAR**



## Lógica de Programação



Introdução  
Compreender o conceito de variável e constante

Roni Schanuel  
08-08-2022

# Algumas características do curso

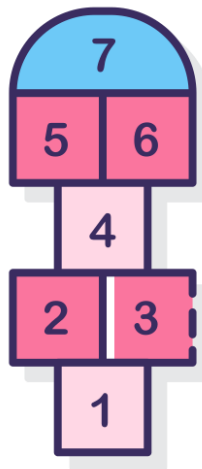
- ❖ Dedicção
- ❖ Trabalho em equipe
- ❖ Organização
- ❖ Adaptação às mudanças
- ❖ Criatividade
- ❖ Participação
- ❖ Iniciativa

# Avaliações

Testes de conhecimento individuais, trabalhos em grupo e autoavaliação que deverão ter como resultado as características esperadas.

# Algoritmo

- Sequência **finita de passos** que levam à execução de uma tarefa
- Algo muito comum no nosso dia a dia, sendo de TI ou não =D



# Programa

- **Algoritmo** escrito em uma **linguagem** de programação

# Linguagem natural



## Linguagem de máquina



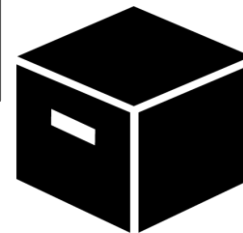
# Linguagem de programação



# Programador



# Compilador



# O que vamos utilizar?

- ❖ Uma linguagem de programação possui comandos e sintaxes para escrever o código.
- ❖ IDE(Integrated Development Environment) - É um ambiente usado para desenvolver uma aplicação.
- ❖ Compilador - programa responsável por transformar o código fonte em código de máquina.

# E qual linguagem usaremos neste curso?

- Neste nivelamento o foco é entender os princípios da programação. Assim, utilizaremos a IDE Portugol Studio, que possui uma linguagem própria que aproxima a linguagem de programação ao português!

{Portugol  Studio}

# Estrutura inicial de um código em Portugol

programa

```
{  
    /* Declaração de variáveis, estruturas e outras funções */  
  
    funcao inicio ()  
    {  
        /*Execução da função início*/  
    }  
}
```



# Fluxograma - Outra forma de representação

- É uma forma universal de representação, pois se utiliza de figuras geométricas para ilustrar passos a serem seguidos para a resolução de problemas



Indica o início ou fim do algoritmo



Indica o sentido do fluxo de dados



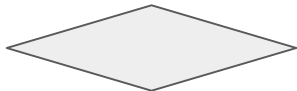
Indica cálculos e atribuições de valores



Representa a entrada de dados



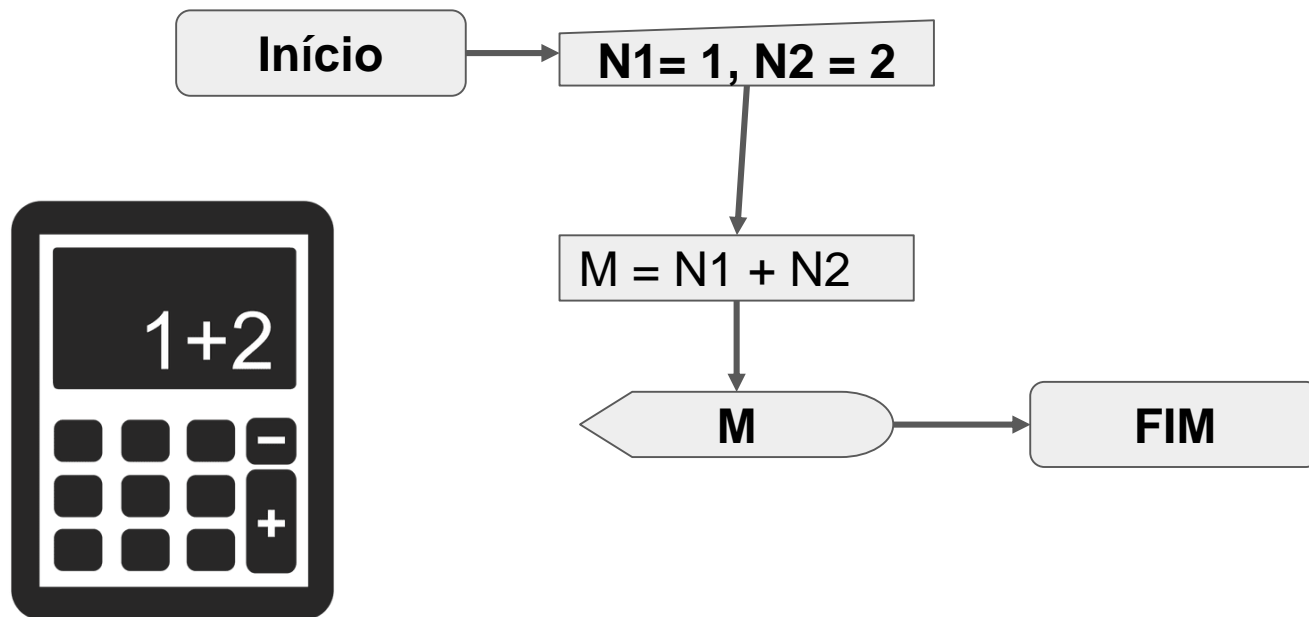
Representa a saída de dados



Tomada de decisão

# Fluxograma

Exemplo : Soma de 2 números



# O que iremos aprender :

- Operações de entrada e saída
- O que são variáveis e constantes
- Desvios condicionais ( se e senão )
- Operadores Lógicos ( E, OU ... )
- Laços de repetição ( enquanto, para ... até )
- Estruturas de dados ( Vetores, Matrizes, Filas e Pilhas )
- Subrotinas ( Funções )
  - Recursividade
  - Bibliotecas

**Com exemplos práticos!!**



Mas antes, vamos explorar o Portugal Studio

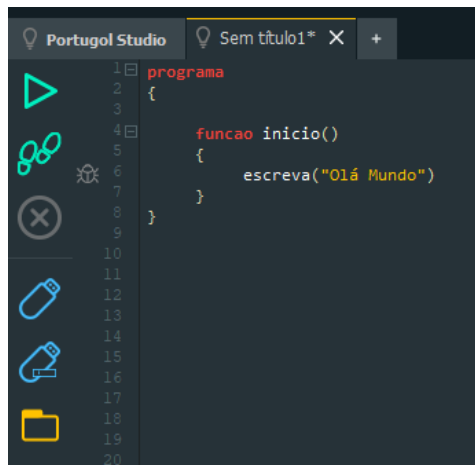


# Nosso primeiro programa: Olá mundo!

Execute no Portugol Studio o código : Olá Mundo

- O que esse código faz?
- Quais dificuldades vocês tiveram em entender este trecho de código?

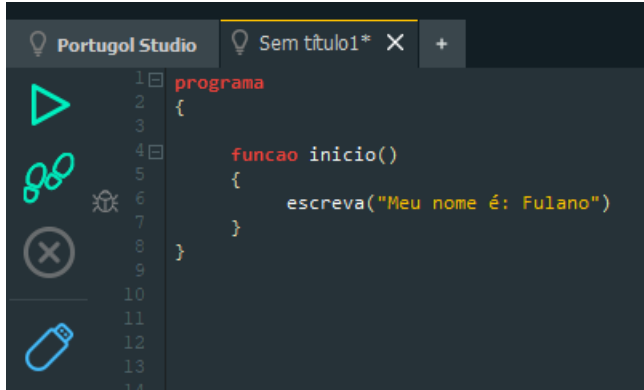
<HELLO WORLD />



```
1 programa
2 {
3
4     funcao inicio()
5     {
6         escreva("Olá Mundo")
7     }
8 }
9
10
11
12
13
14
15
16
17
18
19
20
```

# Identificando-se : Qual é o seu nome?

Execute os seguintes programas - **‘Meu nome é: Fulano’**



```
1 programa
2 {
3
4     funcao inicio()
5     {
6         escreva("Meu nome é: Fulano")
7     }
8 }
9
10
11
12
13
14
```

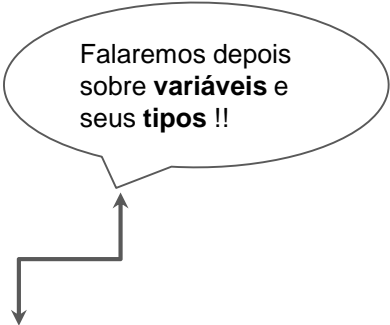


# Operações de entrada e saída

Fazendo a leitura do nome a partir do teclado.

programa

```
{  
    funcao inicio ()  
    {  
        cadeia nome /*cadeia se refere ao tipo da variável que é uma cadeia de  
caracteres*/  
  
        escreva("Digite seu nome: ")  
        leia(nome)  
        escreva("Seu nome é : ", "\n", nome )  
    }  
}
```



Falaremos depois  
sobre **variáveis** e  
seus **tipos** !!

**cadeia** nome /\*cadeia se refere ao **tipo** da **variável** que é uma **cadeia de**

# Operações de entrada e saída

Quando escrevemos :

**cadeia** nome

**leia**(nome)

**leia** é uma operação de **entrada** que permite que o que escrevemos no teclado seja **lido e armazenado** na variável “**nome**”. Logo estamos **entrando** com uma informação no programada durante sua execução.



# Por que **entrada** e **saída**?

Quando escrevemos :

```
cadeia nome = Maria
```

```
escreva("Meu nome é: ", nome)
```

**escreva** é uma operação de **saída** que permite que a informação escrita entre seus parênteses “()” seja apresentado na tela do computador, logo como é uma informação de apresentação, entendemos como uma informação de **saída**.

# Voltando ao programa anterior...

Fazendo a leitura do nome a partir do teclado.

programa

{

funcao inicio ()

{

caracteres\*/

**cadeia** nome /\*cadeia se refere ao **tipo** da **variável** que é uma **cadeia de**

escreva("Digite seu nome: ")

leia(nome)

escreva("Seu nome é : ", "\n", nome )

}

}

Ficou faltando entendermos o que são as **variáveis**!



# Variáveis e constantes

- **Variáveis** e **constantes** representam uma posição na memória, onde pode ser armazenado um **único dado** ( valor ).
- Possuem **tipo**, **nome** e um **valor**
- A diferença entre variáveis e constantes é que enquanto o **valor da variável pode mudar** durante a execução do programa o **valor da constante não**.

```
cadeia nome = "Raul"  
escreva (nome)  
nome = "Pedro"  
escreva ("\n",nome)
```

O que será  
impresso na tela?



# Variáveis e constantes

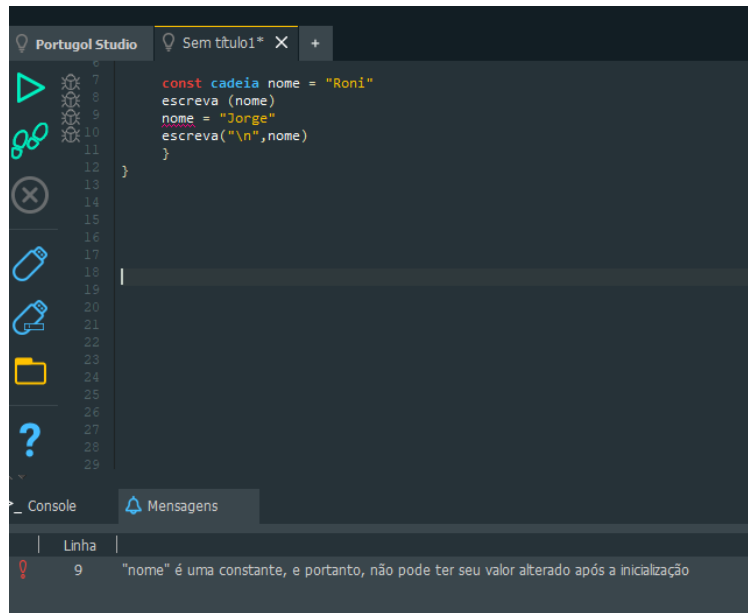
- Em algumas linguagens ( incluindo Portugol ) as variáveis podem ser tipadas, ou seja, aceitam apenas valores referentes ao seu tipo, representado antes do nome da variável:
  - Tipos de variáveis na linguagem do Portugol Studio
    - inteiro : Número inteiros -> 1 ; 2 ; 3
    - real : Números de ponto flutuante -> 1.1 ; 3.14 ; 10.3
    - cadeia : Cadeia de caracteres -> “Adoro estudar programação”
    - caracter : Apenas um caractere -> “A”, “1”
    - logico : Caractere booleano : verdadeiro, falso

# Variáveis e constantes

- Finalmente, para declarar uma **constante** basta colocar o indicador **const** antes da declaração da constante

**const** cadeia nome = Raul  
escreva (nome)  
nome = Pedro  
escreva (nome)

O que será impresso na tela?



The screenshot shows the Portugol Studio IDE with a code editor and a console window. The code editor contains the following code:

```
1 const cadeia nome = "Roni"
2 escreva (nome)
3 nome = "Jorge"
4 escreva("\n", nome)
5 }
```

The console window at the bottom shows a message:

9 "nome" é uma constante, e portanto, não pode ter seu valor alterado após a inicialização

# Exemplo Variáveis

inteiro idade

real peso

real altura

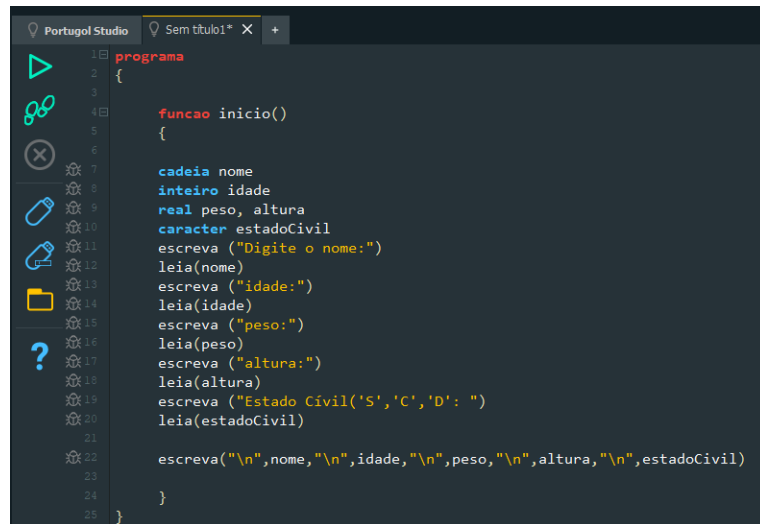
idade = 10

## Memória

idade	10
peso	
altura	

## Exercício:

Faça um programa para ler o nome, idade, peso, altura e estado civil sendo do tipo caracter de uma pessoa e exibir os dados na tela.



```
Portugol Studio  Sem título1* X +
1 programa
2 {
3
4     funcao inicio()
5     {
6
7         cadeia nome
8         inteiro idade
9         real peso, altura
10        caracter estadoCivil
11        escreva ("Digite o nome:")
12        leia(nome)
13        escreva ("idade:")
14        leia(idade)
15        escreva ("peso:")
16        leia(peso)
17        escreva ("altura:")
18        leia(altura)
19        escreva ("Estado Civil('S','C','D': ")
20        leia(estadoCivil)
21
22        escreva("\n",nome,"\n",idade,"\n",peso,"\n",altura,"\n",estadoCivil)
23
24    }
25 }
```

# Concatenação de Strings

```
programa
{
    funcao inicio()
    {
        cadeia texto1, texto2
        escreva("Digite o primeiro texto:")
        leia (texto1)

        escreva("Digite o segundo texto:")
        leia (texto2)

        escreva(texto1 + "-" + texto2)
    }
}
```

## Resultado

```
Digite o texto:Programacao
Digite o segundo texto:Computadores
Programacao-Computadores
```

# Operadores Matemáticos

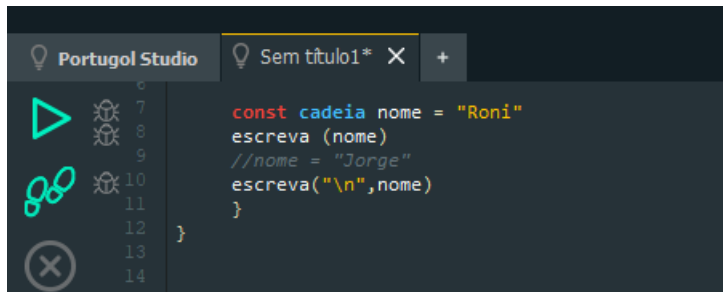
- + soma
- subtração
- \* multiplicação
- / divisão
- % resto da divisão

```
programa
{
    funcao inicio()
    {
        escreva("soma:", 2+4, "\n")
        escreva("subtração:", 1-1, "\n")
        escreva("multiplicação:", 4*2, "\n")
        escreva("divisão:", 9/3, "\n")
        escreva("resto:", 9%3, "\n")
    }
}
```



# Comentários

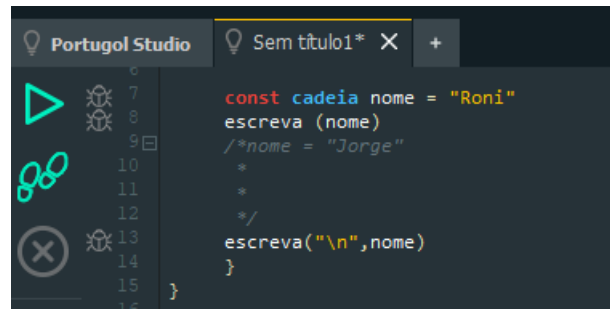
- Usado para ignorar uma parte do código ou para inserir informações sobre um determinado trecho de código.



The screenshot shows the Portugol Studio interface with a dark theme. The title bar includes 'Portugol Studio' and a tab labeled 'Sem título1\*'. The code editor displays the following code:

```
6  
7 const cadeia nome = "Roni"  
8 escreva (nome)  
9 //nome = "Jorge"  
10 escreva("\n",nome)  
11 }  
12  
13  
14
```

On the left side of the editor, there are icons for running (a green play button), debugging (a green bug), and closing (a grey X button).



The screenshot shows the Portugol Studio interface with a dark theme. The title bar includes 'Portugol Studio' and a tab labeled 'Sem título1\*'. The code editor displays the following code:

```
6  
7 const cadeia nome = "Roni"  
8 escreva (nome)  
9 /*nome = "Jorge"  
10 *  
11 *  
12 */  
13 escreva("\n",nome)  
14 }  
15  
16
```

On the left side of the editor, there are icons for running (a green play button), debugging (a green bug), and closing (a grey X button).

# Teclas de Atalho

Pressione F11 para visualizar as teclas de atalho



	Ação	Atalho
	Abrir um arquivo	Ctrl+O
	Ajuda	F1
	Ativar/desativar a centralização de código fonte	Shift+Pause/Break
	Crear novo arquivo	Ctrl+N
	Depurar programa atual	Shift+F5
	Depurar próxima instrução do programa atual	Shift+F9
	Dicas de Interface	F3
	Executar programa atual	Shift+F6
	Exibir tela inicial	Alt+Home
	Expandir/restaurar o tamanho do editor	Shift+Esc
	Fechar a aba atual	Ctrl+Q
	Fechar todas as abas	Ctrl+Shift+Q
	Interromper a execução/depuração do programa atual	Shift+F7
	Pesquisar e/ou substituir um texto no arquivo atual	Ctrl+F
	Salvar o arquivo atual	Ctrl+S
	Salvar uma cópia do arquivo atual	Ctrl+Shift+S
	Selecionar a aba anterior (à esquerda)	Alt+Esquerda
	Selecionar a próxima aba (à direita)	Alt+Direita

**Preenchimento Automático - CTRL + Espaço**

## Exercícios

- 1) Leia dois valores pelo teclado e imprima a soma.
- 2) Construir um algoritmo que leia um número e exiba na tela o seu sucessor e antecessor.
- 3) Construa um algoritmo que leia o nome de um aluno, disciplina, quatro notas e exiba na tela a média.
- 4) Faça um programa com duas variáveis ano\_nascimento que receberá o ano que você nasceu e outra variável com o nome ano\_futuro que deverá ser atribuído o valor 2035. Criar uma variável com o nome resultado para calcular a diferença. No final escreva na tela qual será a sua idade em 2035.

## Exercícios

5) Uma empresa paga R\$10.00 por hora normal trabalhada e R\$ 15.00 por hora extra. Escreva um algoritmo que leia o total de horas normais trabalhadas e o total de horas extras trabalhadas por um empregado em um ano e calcule o salário anual deste trabalhador.

*Exemplo : Entrada : Digite o número de horas trabalhadas no ano : 1760*

*Digite o número de horas extras trabalhadas no ano : 400*

*Saída :   Seu salário anual é de : R\$ 23600*

6) Escreva um programa que leia a temperatura em Celsius e retorne o valor em Fahrenheit

$$F = C \times 1,8 + 32$$

*Para que você consiga converter de grau Celsius para Fahrenheit, basta multiplicar a temperatura em graus Celsius por 1,8 e somar 32.*

*Exemplo:*

*37° C para Fahrenheit:*

$$F = 37 \times 1,8 + 32;$$

$$F = 98,6;$$

*37 graus Celsius equivalem a 98,6 graus Fahrenheit.*

## Exercícios

7) Criar um algoritmo que leia dois números inteiros e imprima a seguinte saída:

Dividendo, Divisor, Quociente e Resto

8) Calcule a área e o preço de um terreno

$\text{area} = \text{largura} \times \text{comprimento}$

$\text{preco} = \text{área} \times \text{preco do metro quadrado}$

9) Fazer um algoritmo para ler a distância percorrida em km e o total gasto de combustível, no final deverá ser exibido o consumo médio do carro.

10) Criar um algoritmo para ler um tempo em segundos e fazer a impressão no console no seguinte formato:  
hora:minuto:segundos