


NExT

Nova Experiência de Trabalho

Tatyane Calixto
tscs@cesar.org.br

Erick Simões
esm@cesar.org.br




Todo nosso material de aula, exercícios e chamada vão ser disponibilizados diariamente via Google Classroom 


Classroom

Código da sala:

ywo4frr

Chat e nome

 Lembrando: as dúvidas devem ser direcionadas aos monitores no privado, para deixar o chat mais limpo 🙌

 Não esqueça de colocar seu no seu nome, aqui no Zoom, o número da sua sala, exemplo:

3 - Triago Sabão

Como fazer:

1. Clique em **Participantes**
2. Na lista que aparece, clique nos três pontinhos ao lado do seu nome ...
3. Clique em renomear



REVISÃO

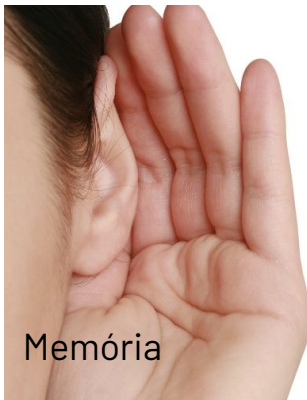
Fundamentos

Programar

Variáveis

Variáveis

- Na programação, uma variável é um objeto (uma posição, frequentemente localizada na memória) capaz de reter e representar um valor ou expressão.
- “Espaço” em memória para guardar um valor durante a execução de um programa.
- Devem ser declaradas antes de serem utilizadas



Memória

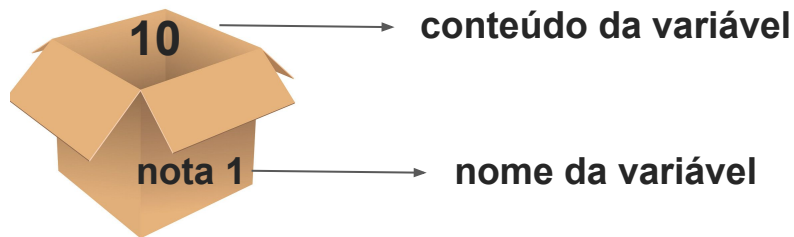


Declarando variável

Variáveis

Exemplo:

Você tem um programa que verifica as notas dos estudantes.



Programar

Fundamentos

Resumo:

Tipos de variáveis:

- **inteiro, real, caracter, cadeia, lógico**

Declaração de uma variável:

- **[tipo] [nome]**

Atribuição:

- **[nome] = [valor]**

Declaração com atribuição:

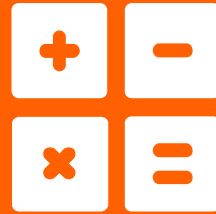
- **[tipo] [nome] = [valor]**

Variável x Constante:

- **Variável:** o valor pode mudar ao decorrer da execução do código
- **Constante:** o valor não muda depois que foi declarado

Exibir e receber informações:

- **escreva()**
- **leia()**



OPERADORES

Operadores

Atribuição

É possível realizar operações sobre as variáveis e sobre algumas constantes:

- **Atribuição:** guardar valor em uma variável

=

- **Aritmética:** soma, subtração, multiplicação, divisão, resto da divisão

+

-

*

/

%

- **Incremento:** incrementa o valor de uma variável

++

--

+=

-=

*=

/=

- **Relacional:** maior, menor, igual, diferente, maior ou igual, menor ou igual

>

<

>=

<=

==

!=

- **Lógica:** e, ou, não

e

ou

nao

Operadores

Atribuição

Operador de Atribuição

- Símbolo (recebe): =
- Sintaxe:
 - `[variável] = [valor]`
 - `[variável] = [expressão]`
- Primeiro avalia o lado direito do operador
- O valor de `[valor]` ou `[expressão]` é guardado na posição de memória endereçada por `[variável]`

Operadores

Atribuição

Operador de Atribuição

```
programa
{
    funcao inicio()
    {
        inteiro a, b
        a = 10
        b = 4 + 1
        a = b
        b = 7

        escreva("a: ", a)
        escreva("b: ", b)
    }
}
```

Operadores Aritmética

Operadores Aritméticos

- Para construir algoritmos que realizam cálculos matemáticos, precisamos utilizar os operadores aritméticos;
- As expressões aritméticas devem ser linearizadas:

Tradicional

$$\left\{ \left[\frac{7}{3} - (7 + 2) \right] + 5 \right\} . 2$$

Computacional

$$((7/3 - (7 + 2) + 5)) * 2$$

- Deve ser feito o mapeamento dos operadores utilizados tradicionalmente para o português estruturado, que é utilizado no Portugol Studio.

Operadores Aritmética

Operadores Aritméticos

Operadores Aritméticos	Portugol Studio	Descrição
Adição	+	Operador tradicional de adição
Subtração	-	Operador tradicional de subtração
Multiplicação	*	Operador tradicional de multiplicação
Divisão	/	Operador tradicional de divisão
Módulo (Resto da Divisão)	%	Resto da divisão inteira ($9\%2=1$)
Incremento	a++	Acrescenta 1 ao valor da variável
Decremento	a--	Diminui 1 do valor da variável

Operadores Aritmética

Operadores Aritméticos

Os operadores possuem as mesmas regras de precedência aritmética da matemática:

- $1 + 5 * 8 / 4 * 2 = ?$
- $(1 + 5) * (8 / 4) * 2 = ?$
- $1 + (5 * (8 / 4) * 2) = ?$

Operadores Incremento

Operadores de Incremento

- Símbolos: `++` `--` `+=` `-=` `*=` `/=`
- Sintaxe:
 - `[variável]++`
 - `[variável] += [valor]`
- Servem para escrevermos algumas operações de forma mais simples:

```
a++
```

é o mesmo que

```
a = a + 1
```

```
b += 4
```

é o mesmo que

```
b = b + 4
```



EXEMPLOS

variáveis

EXEMPLO 1:

Mostrar a soma, produto e média aritmética de três números informados pelo usuário.

EXEMPLO 2:

Ler o tempo de duração de um exame expresso em horas e mostrar em segundos.

Variáveis

EXEMPLOS

Variáveis

EXEMPLOS

EXEMPLO 1:

@CESAR 2022 | Todos os Direitos Reservados



```
programa
{

    funcao inicio()
    {
        real n1, n2, n3, soma, produto, media

        escreva("Informe o 1º número: ")
        leia(n1)
        escreva("Informe o 2º número: ")
        leia(n2)
        escreva("Informe o 3º número: ")
        leia(n3)

        soma = n1 + n2 + n3
        produto = n1 * n2 * n3
        media = (n1 + n2 + n3) / 3

        escreva(soma, "\n", produto, "\n", media)
    }
}
```

EXEMPLO 2:

@CESAR 2022 | Todos os Direitos Reservados



Variáveis

EXEMPLOS

```
programa
{
    funcao inicio()
    {
        inteiro horas, segundos

        escreva("Insira o tempo em horas: ")
        leia(horas)

        segundos = horas * 60 * 60

        escreva("Tempo do exame em segundos: ", segundos)
    }
}
```

Operadores Relacionais

Operadores Relacionais

Realizam comparações e retornam valores lógicos:
verdadeiro ou **falso**;

Operadores Relacionais	Matemática	Portugol Studio
Maior	$>$	$>$
Menor	$<$	$<$
Maior ou igual	\geq	$>=$
Menor ou igual	\leq	$<=$
Igual	$=$	$==$
Diferente	\neq	$!=$

Operadores Lógicos

Operadores Lógicos

Servem para combinar resultados de expressões retornando se o resultado final é VERDADEIRO ou FALSO.

Operadores Lógicos	Portugol Studio	Significado
Conjunção	e	O resultado será verdadeiro se uma parte e a outra parte forem verdadeiras
Disjunção	ou	O resultado será verdadeiro se uma parte ou a outra parte forem verdadeiras
Não Lógico	nao	O resultado será a inversão do valor lógico. Se for verdadeiro , torna-se falso .

Operadores Lógicos

Operadores Lógicos

- Como chegamos a estes valores?
- Há um método chamado tabela-verdade, que mostra os resultados das aplicações dos operadores lógicos.

A	B	A e B	A ou B	nao A	nao B
verdadeiro	verdadeiro				
verdadeiro	falso				
falso	verdadeiro				
falso	falso				

Operadores Lógicos

Vamos praticar?

A = 10

B = 15

C = 2

Operadores Lógicos

Expressões			Resultado
A == B	e	B > C	
A != B	ou	B < C	
	nao	A > B	
A >= B	e	B == C	
A <= B	ou	B != C	

Operadores

Lógicos

Operadores - Lógicos

- Vamos praticar? E no Portugal, como funciona?

```
programa
{
    funcao inicio()
    {
        escreva(verdadeiro e verdadeiro, "\n")
        escreva(verdadeiro e falso, "\n")
        escreva(falso e falso, "\n")
    }
}
```




EXEMPLOS

lógica

EXEMPLO 3:

Receba o gênero e a idade do usuário e verifique se ele deve realizar o alistamento militar.

Lógica

EXEMPLOS

EXEMPLO 3:

```
programa
{

    funcao inicio()
    {
        cadeia genero
        inteiro idade

        leia(genero)
        leia(idade)

        escreva("Deve se alistar ", genero == "masculino"
e idade == 18)
    }
}
```

Lógica

EXEMPLOS

EXEMPLO 4:

O programa deve ler um número inteiro e depois imprimir:

- O antecessor desse número;
- O sucessor desse número;
- O número elevado a 4ª potência;
- A raiz quadrada desse número (aproximadamente)

 5 min

Lógica

EXEMPLOS



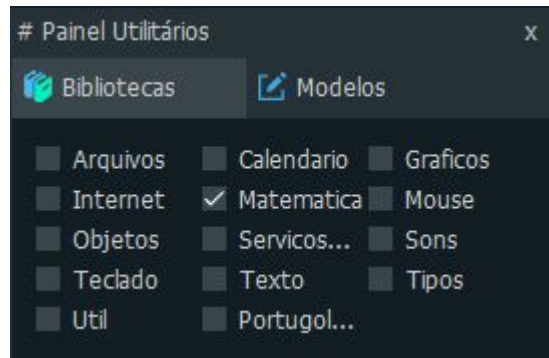
BIBLIOTECAS

Bibliotecas

Matemática

Bibliotecas

- Chamamos de biblioteca um conjunto de soluções já desenvolvidas para resolver um problema conhecido.
- O Portugol Studio oferece uma série de bibliotecas já implementadas para uso.
- Algumas delas são:
 - Matematica
 - Texto
 - Calendario
 - Util



Bibliotecas

Matemática

Bibliotecas

- Para usar uma biblioteca, basta incluí-la no início do código:

```
programa
{
    inclui biblioteca Matematica

    funcao inicio()
    {
        escreva(Matematica.potencia(10, 2))
    }
}
```

```
programa
{
    inclui biblioteca Matematica --> mat

    funcao inicio()
    {
        escreva(mat.potencia(10, 2))
    }
}
```

EXEMPLO 4:

O programa deve ler um número inteiro e depois imprimir:

- O antecessor desse número;
- O sucessor desse número;
- O número elevado a 4ª potência;
- A raiz quadrada desse número (aproximadamente)

 5 min

Variáveis

EXEMPLOS

EXEMPLO 4:

```
programa
{
    inclui biblioteca Matematica --> mat
    funcao inicio()
    {
        inteiro num
        leia(num)

        escreva("Antecessor: ", num - 1, "\n")
        escreva("Sucessor: ", num + 1, "\n")
        escreva("Potência: ", mat.potencia(num, 4), "\n")
        escreva("Raiz: ", mat.raiz(num, 2.0), "\n")
    }
}
```

Variáveis

EXEMPLOS



Estruturas de Decisão

Estrutura Condicional

Estruturas de Decisão

Introdução

Até agora, todos os algoritmos vistos eram sequenciais

- Instruções são executadas uma após a outra
- De cima para baixo

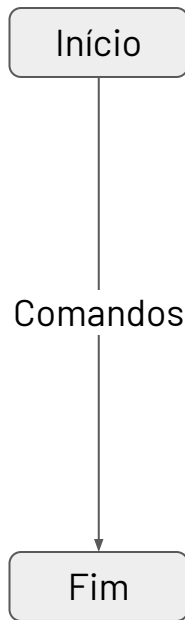
Porém, problemas reais, em sua maioria, exigem uma tomada de decisão no algoritmo, onde há comandos que desviam o fluxo de execução

- Algumas instruções podem ser ignoradas
- Depende da condição dada

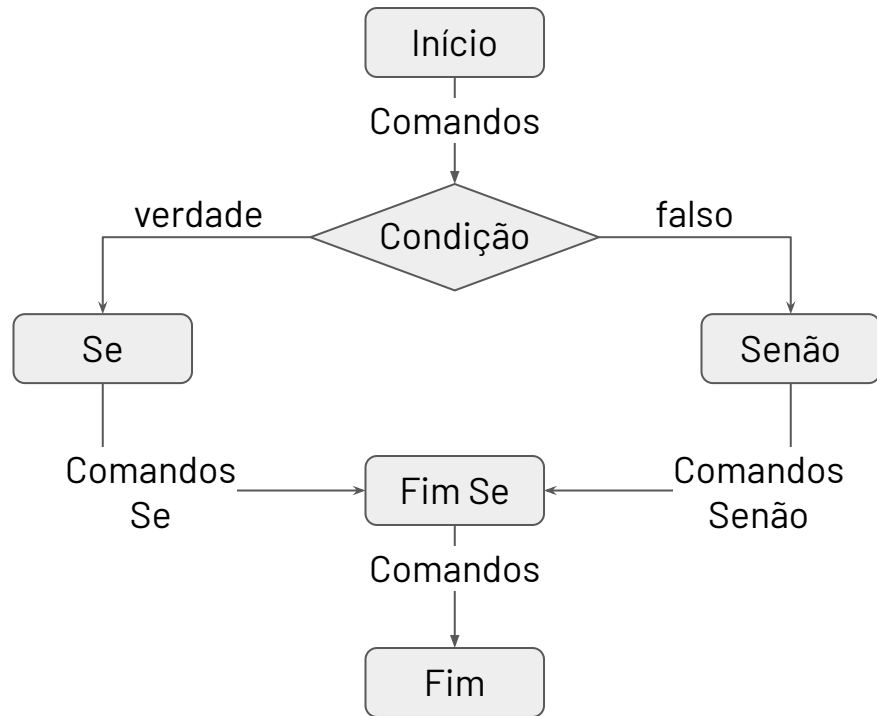
Estruturas de Decisão

Introdução

Execução normal



Instrução de Decisão



Estruturas de Decisão

Quando usar as instruções de decisão?

- Quando queremos que uma condição seja analisada;
- Dependem de uma condição;
- Resultado da condição deve retornar VERDADEIRO ou FALSO;
- Caso esta condição seja verdadeira, um comando será executado;
- Caso esta condição seja falsa, outro comando será executado.

Estruturas de Decisão

Decisão Simples

Instrução de Decisão Simples

- Utiliza a seguinte sintaxe:

```
se(condição) {  
    // código a ser executado  
}
```

- A expressão da *condição* é avaliada;
- Se o resultado da avaliação é **verdadeiro**
- Os comandos dentro do bloco **indentado** são executados

Estruturas de Decisão

Decisão Simples

Instrução de Decisão Simples

- Se o resultado da avaliação é FALSO, tudo dentro do escopo do "se" é ignorado
- Execução segue normal depois do "se"

```
se (num > 10) {  
    escreva(num, " é maior que 10")  
}
```



EXEMPLOS

se

EXEMPLO 5:

Ler dois valores reais, efetuar a adição e apresentar seu resultado apenas se o valor somado for maior ou igual a 10.

EXEMPLO 6:

Construa um programa que peça a entrada de dois números inteiros x e y.

- Se os números forem iguais, imprima: *"São iguais"*.
- Se x for maior que y, imprima: *"A entrada x é maior que entrada y"*.
- Se y for maior que x, imprima: *"A entrada x é menor que entrada y"*.

Se
EXEMPLOS

EXEMPLO 5:

```
programa
{
    funcao inicio()
    {
        real n1, n2, total

        leia(n1)
        leia(n2)

        total = n1 + n2

        se(total >= 10){
            escreva(total)
        }
    }
}
```

Se

EXEMPLOS

EXEMPLO 6:

```
funcao inicio()
{
    inteiro x, y

    leia(x)
    leia(y)

    se(x == y){
        escreva("São iguais")
    }

    se(x > y){
        escreva(x, " é maior que ", y)
    }

    se(x < y){
        escreva(x, " é menor que ", y)
    }
}
```

Se

EXEMPLOS

EXEMPLO 7:

Receba o gênero e a idade do usuário e verifique se ele deve realizar o alistamento militar.

EXEMPLO 8:

Ler 4 números inteiros e efetuar a soma deles apenas se todos forem ímpares.

Se

EXEMPLOS

EXEMPLO 7:

```
programa
{
    funcao inicio()
    {
        cadeia genero
        inteiro idade

        leia(genero)
        leia(idade)

        se(genero == "masculino" e idade == 18) {
            escreva("Deve se alistar ")
        }
    }
}
```

Se

EXEMPLOS

EXEMPLO 8:

```
funcao inicio()
{
    inteiro n1, n2, n3, n4

    leia(n1, n2, n3, n4)

    se(n1%2 != 0 e n2%2 != 0 e n3%2 != 0 e n4%2 != 0) {
        escreva(n1 + n2 + n3 + n4)
    }
}
```

Se
EXEMPLOS

Breakout Time!

Resolva os desafios da lista de exercícios com sua sala no breakout room.

A lista possui exercícios em duas categorias:

- Exercícios fundamentais;
- Exercícios de aprofundamento.

Se precisar de ajuda, chame uma das pessoas monitoras ou professoras.



Lista de Exercícios 02



C . E . S . A . R

Pessoas impulsionando inovação.
Inovação impulsionando negócios.

Tatyane Calixto
tscs@cesar.org.br

Erick Simões
esm@cesar.org.br

e a melhor equipe de monitores
da CESAR School 

