



PSEUDOLINGUAGEM

FAPESC – DESENVOLVEDORES PARA TECNOLOGIA DA INFORMAÇÃO

HABNER FABRÍCIO BOESING
habner.boesing@unoesc.edu.br

OBJETIVOS

- Pseudolinguagem
- Ambiente de desenvolvimento VisualG
- Variáveis e constantes
- Tipos de dados
- Declaração de variáveis
- Expressões com operadores aritméticos, relacionais e lógicos

PSEUDOLINGUAGEM

- Linguagem de programação simplificada
- Utiliza uma linguagem simples (nativa a quem o escreve, de forma a ser entendida por qualquer pessoa)
- Usada para escrever algoritmos, sem se preocupar com toda a sintaxe e os detalhes de uma linguagem de programação real.
- Auxilia no entendimento da construção de algoritmos e na aplicação da lógica de programação que depois pode ser adaptada para uma linguagem de programação real.
- Em nossas aulas usaremos a pseudolinguagem **Portugol**.



PORTUGOL

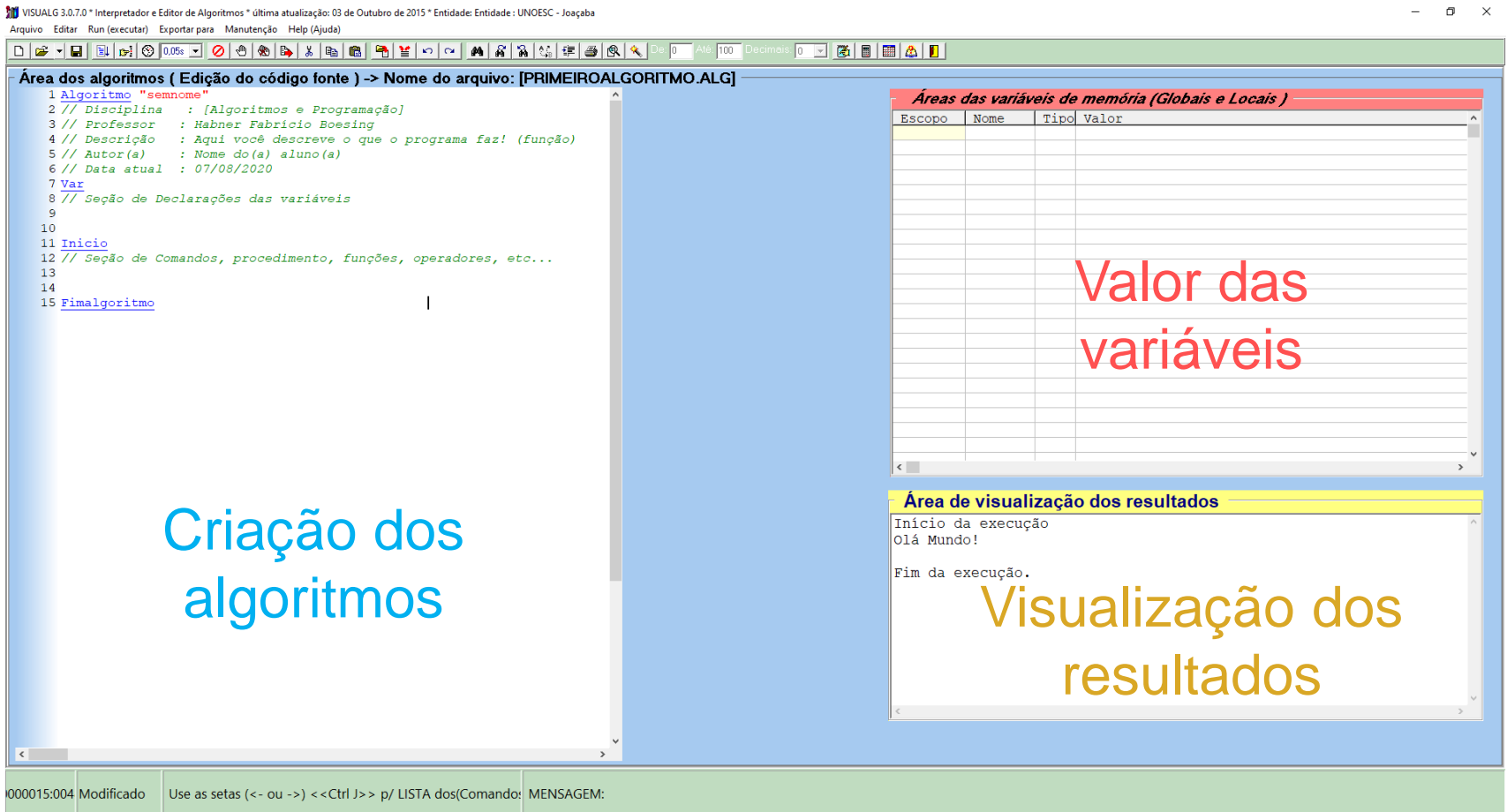
- Portugol, também conhecido como português estruturado é um pseudocódigo escrito em português.
- Objetivo é permitir construir algoritmos de forma simples e intuitiva, para que posteriormente estes algoritmos possam passar por um refinamento e ser adaptados em qualquer linguagem de programação.
- Lembre-se: Um programador não precisa conhecer todas as linguagens de programação, mas sim, precisa conhecer a lógica de programação.

PORTUGOL – AMBIENTE DE DESENVOLVIMENTO

- Para que possam ser criados os algoritmos utilizando o pseudocódigo Portugol, é necessário instalar um ambiente de desenvolvimento no computador, para que a partir deste os códigos possam ser interpretados pela máquina.
- Alguns softwares que podem ser utilizados com Portugol são:
 - VisualG (<https://visualg3.com.br/>)
 - Portugol Studio (<http://lite.acad.univali.br/portugol/>)



INTERFACE VISUALG



ESTRUTURA DO ALGORITMO EM PORTUGOL

```
1 Algoritmo "semnome"  
2 // Disciplina    : [Algoritmos e Programação]  
3 // Professor     : Habner Fabrício Boesing  
4 // Descrição     : Aqui você descreve o que o programa faz! (função)  
5 // Autor(a)      : Nome do(a) aluno(a)  
6 // Data atual    : 07/08/2020  
7 Var  
8 // Seção de Declarações das variáveis  
9  
10  
11 Inicio  
12 // Seção de Comandos, procedimento, funções, operadores, etc...  
13  
14  
15 Fimalgoritmo
```

ESTRURA DO ALGORITMO EM PORTUGOL

```
1 Algoritmo "semnome"
2 // Disciplina   : [Algoritmos e Programação]
3 // Professor    : Habner Fabrício Boesing
4 // Descrição    : Aqui você descreve o que o programa faz! (função)
5 // Autor(a)     : Nome do(a) aluno(a)
6 // Data atual   : 07/08/2020
7 Var
8 // Seção de Declarações das variáveis
9
10
11 Inicio
12 // Seção de Comandos, procedimento, funções, operadores, etc...
13
14
15 Fimalgoritmo
```

Nome do algoritmo

Informações sobre o algoritmo

Declaração de Variáveis

Comandos do algoritmo

Número da linha

PRIMEIRO ALGORITMO – OLÁ MUNDO!

- Instalar o software VisualG
- Criar o algoritmo abaixo:

```
1 Algoritmo "primeiroalgoritmo"  
2 // Disciplina      : [Algoritmos e Programação]  
3 // Professor       : Habner Fabrício Boesing  
4 // Descrição       : Exibe a mensagem Olá Mundo! na tela  
5 // Autor(a)        : Nome do(a) aluno(a)  
6 // Data atual       : 07/08/2020  
7 Var  
8 // Seção de Declarações das variáveis  
9  
10  
11 Inicio  
12 // Seção de Comandos, procedimento, funções, operadores, etc...  
13 escreval("Olá Mundo!")  
14  
15 Fimalgoritmo
```

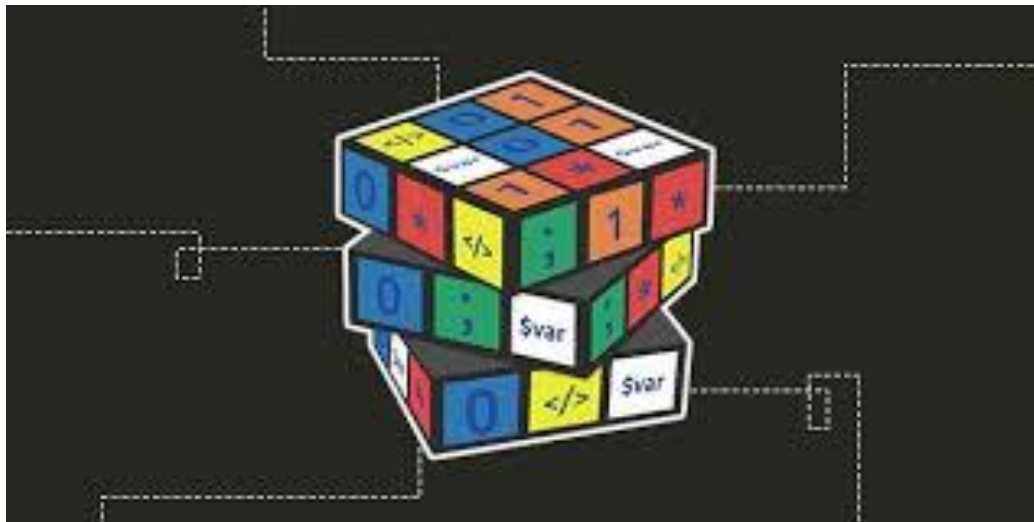
LISTA DE ATALHOS

- Salvar: Ctrl+S
- Salvar Como: Ctrl+Alt+C
- Desfazer: Ctrl+Z
- Refazer: Shift+Ctrl+Z
- Executar algoritmo: F9
- Executar algoritmo passo a passo: F8
- Indentação do código: Ctrl+G

*Para mais atalhos e informações conferir o manual do VisualG

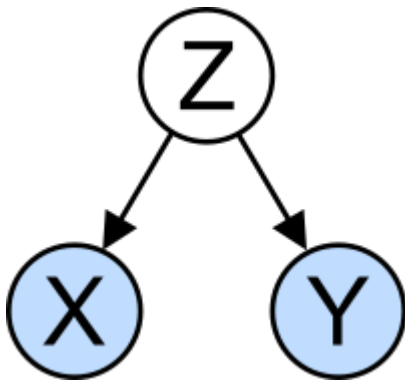
VARIÁVEIS E CONSTANTES

- O computador utiliza a memória RAM para armazenar dados temporários.
- Na programação também temos dados que são armazenados temporariamente na memória enquanto o programa está sendo executado.
- Temos duas formas de armazenar estes dados: utilizando variáveis ou constantes.

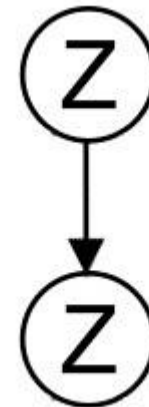


DIFERENÇAS ENTRE VARIÁVEIS E CONTANTES

- Uma **variável** é um espaço para armazenar um tipo de dado que **poderá ser modificado quando necessário** durante a execução do programa.
- Uma **constante** é um espaço para armazenar um tipo de dado que já possui um dado definido no início do programa e **não poderá ser alterado**.



Variável



Constante

TIPOS DE DADOS DE VARIÁVEIS

- Quando são armazenados variáveis na memória é necessário definir qual será o tipo de dado que será armazenado.
- Dentre os tipos de dados disponíveis no Visualg, temos:

Tipo	Descrição	Exemplos
caracter	Conjuntos de caracteres	“Texto entre aspas duplas”
inteiro	Valores numéricos do tipo inteiro, ou seja, sem casas decimais	-3, -2, -1, 0, 1, 2, 3
real	Valores numéricos do tipo real, ou seja, com casas decimais	3,44
logico	Valores lógicos	VERDADEIRO ou FALSO
vetor	Tipo de dado estruturado, que suporta vetores e matrizes	vetor [1..10]

NOMEAÇÃO DE VARIÁVEIS

- Toda variável deve receber um nome, o qual tem que respeitar algumas regras de nomeação:
 - Deve sempre começar com uma letra;
 - Aceita letras maiúsculas e minúsculas;
 - Não podem conter acentuação;
 - Não podem conter espaços ou caracteres especiais, exceto o sublinhado “_”;
 - Não podem conter acentuação;
 - O limite máximo é de 30 caracteres;
 - Não podem existir duas variáveis com mesmo nome;

Nomes válidos: nome, NOME, Nome, nome_l, nome_l, endereco

Nomes inválidos: nome*, nome l, lnome, endereço

DECLARAÇÃO DE VARIÁVEIS

- A declaração de variáveis representa a criação das variáveis no algoritmo, ou seja, você declara para o algoritmo que você vai utilizar variáveis para armazenamento de valores.
- A declaração de variáveis pode ocorrer somente na seção de variáveis.
- Para declarar uma variável é necessário inserir o nome da variável seguido de dois pontos e a identificação do tipo da variável.
- Exemplo:

```
7 Var  
8 // Seção de Declarações das variáveis  
9 nome: caracter  
10 idade: inteiro  
11 nota: real
```

ATRIBUIÇÃO DE VALORES NAS VARIÁVEIS

- A atribuição de valores nas variáveis indica qual vai ser o valor que será armazenado naquele momento na variável.
- Por se tratar de variável este valor permanecerá até que um novo valor seja atribuído à variável.
- Para realizar a atribuição de valores, basta indicar a variável que receberá o valor, seguida da combinação de caracteres <- e por fim o valor a ser atribuído.
- Exemplo:

```
7 Var
8 // Seção de Declarações das variáveis
9 nome: caracter
10 idade: inteiro
11 nota: real
12 Inicio
13 // Seção de Comandos, procedimento, funções, operadores, etc...
14 nome <- "João"
15 idade <- 20
16 nota <- 8,5
17 Fimalgoritmo
```


EXIBIÇÃO DE VALORES NA TELA

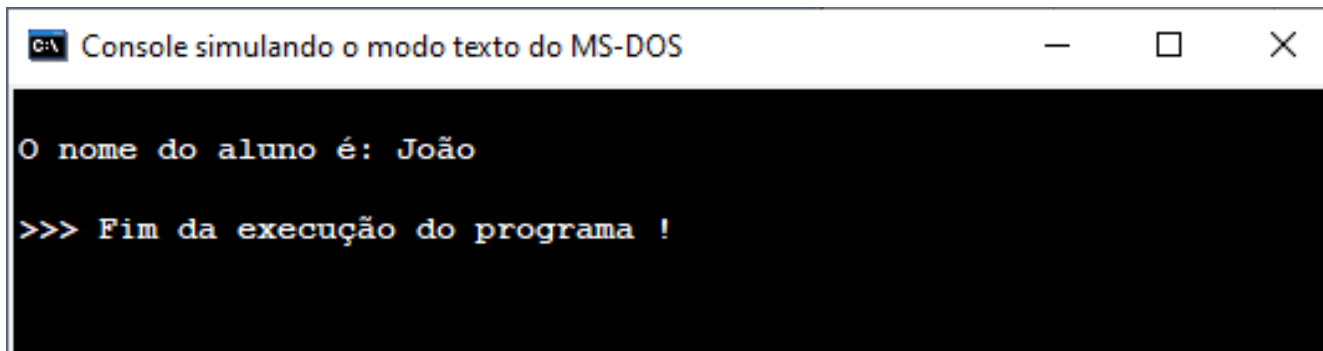
- Para exibir um valor em tela podemos usar 2 comandos:
 - **escreva(mensagem)** – exibirá a mensagem na mesma linha
 - **escreval(mensagem)** – exibirá a mensagem e pulará para a linha seguinte
- Pode-se utilizar para exibir um texto simples ou uma variável
 - **escreval(“Olá mundo”)** – exibirá a mensagem Olá mundo
 - **escreval(nome)** – exibirá o valor contido na variável nome
 - **escreval(“O nome do aluno é: ”+nome)** – exibirá a mensagem de texto seguida pelo valor contido na variável nome

EXIBIÇÃO DE VALORES NA TELA

Exemplo:

```
7 Var  
8 // Seção de Declarações das variáveis  
9 nome: caracter  
10 idade: inteiro  
11 nota: real  
12 Inicio  
13 // Seção de Comandos, procedimento, funções, operadores, etc...  
14 nome <- "João"  
15 idade <- 20  
16 nota <- 8,5  
17 escreval("O nome do aluno é: "+nome)  
18 Fimalgoritmo
```

Resultado



A screenshot of a DOS-style console window. The title bar reads "C:\ Console simulando o modo texto do MS-DOS". The window has standard minimize, maximize, and close buttons. The main area is black with white text. It displays the output of the program: "O nome do aluno é: João" followed by a blank line and then ">>> Fim da execução do programa !".

```
C:\ Console simulando o modo texto do MS-DOS  
  
O nome do aluno é: João  
  
>>> Fim da execução do programa !
```

TIPOS DE OPERADORES

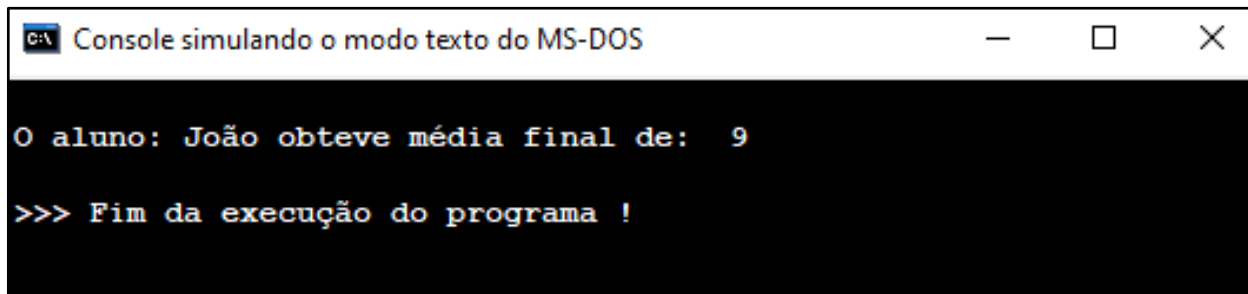
- Os operadores são importantes para realizar as operações entre as diferentes variáveis, criando cálculos ou expressões que resultem em verdadeiro ou falso.
- **Operadores aritméticos:** +, -, *, /, ^ e MOD
- **Operadores relacionais:** =, <, >, <=, >=, <>
- **Operadores lógicos:** e, ou

TIPOS DE OPERADORES

Exemplo:

```
1 Algoritmo "Operadores aritméticos"
2 Var
3 nome: caracter
4 nota1, nota2, nota3, media: real
5 Inicio
6 nome <- "João"
7 nota1 <- 8.5
8 nota2 <- 9
9 nota3 <- 9.5
10 media <- (nota1 + nota2 + nota3)/3
11 escreval("O aluno: ",nome," obteve média final de: ",media)
12 Fimalgoritmo
```

Resultado:



The screenshot shows a window titled "C:\> Console simulando o modo texto do MS-DOS". The window has standard Windows window controls (minimize, maximize, close). The content of the window is a black background with white text. The first line of output is "O aluno: João obteve média final de: 9". The second line is ">>> Fim da execução do programa !".

```
C:\> Console simulando o modo texto do MS-DOS

O aluno: João obteve média final de: 9

>>> Fim da execução do programa !
```

ATRIBUIÇÃO DE VALORES POR MEIO DA INSERÇÃO DO USUÁRIO

- Ao invés de atribuir um valor para uma variável manualmente, é possível fazer com que o usuário insira os valores.
- Para isso precisamos utilizar o seguinte comando:
 - **leia(variável)** – leia seguido no nome da variável que vai receber o valor indicada entre parênteses.
- Exemplo:

```
1 Algoritmo "Boletim Aluno"
2 Var
3 nome: caracter
4 nota1, nota2, nota3, media: real
5 Inicio
6 escreva("Digite o nome do aluno: ")
7 leia(nome)
8 escreva("Digite o nota 1: ")
9 leia(nota1)
10 escreva("Digite o nota 2: ")
11 leia(nota2)
12 escreva("Digite o nota 3: ")
13 leia(nota3)
14 media <- (nota1 + nota2 + nota3)/3
15 escreval("O aluno: ",nome," obteve média final de: ",media)
16 Fimalgoritmo
```

TESTE DE MESA

- Teste de mesa costuma ser uma técnica simples onde até mesmo com um papel e caneta são anotados os valores que são atribuídos às variáveis durante a execução do algoritmo para identificar possíveis problemas de lógica.
- No caso do VisualG, já existe um ambiente integrado que mostra estes valores chamado de Área das variáveis de memória.
- Com isso, é possível acompanhar em tempo de execução do programa, quais são os valores que estão sendo atribuídos a cada variável.

Áreas das variáveis de memória (Globais e Locais)			
Escopo	Nome	Tipo	Valor
GLOBAL	NOME	C	"asdf"
GLOBAL	NOTA1	R	8,500000000000000
GLOBAL	NOTA2	R	9,000000000000000
GLOBAL	NOTA3	R	9,500000000000000
GLOBAL	MEDIA	R	9,000000000000000

COMO EXTRAIR AS INFORMAÇÕES DO PROBLEMA – EXERCÍCIO I

■ Enunciado:

Desenvolva um algoritmo que receba dois números digitados pelo usuário. Posteriormente a isso, realize as 4 operações entre estes números e exiba os resultados na tela juntamente com os números digitados.

■ 1º passo: identificar quais são as possíveis variáveis do problema

- Número 1 e 2;
- Soma, subtração, multiplicação, divisão

■ 2º passo: declarar as variáveis de acordo com seu tipo:

- num1, num2, soma, sub, mult, divi: real

■ 3º passo: pensar nas etapas

- 1) Solicitar os números para o usuário
- 2) Realizar os cálculos
- 3) Exibir os resultados na tela

SOLUÇÃO DO EXERCÍCIO I

■ Resolução

```
1 Algoritmo "Operadores Básicos"
2 Var
3 num1, num2, soma, sub, mult, divi: real
4 Inicio
5 //solicitar números ao usuário
6 escreva("Digite o primeiro número: ")
7 leia(num1)
8 escreva("Digite o primeiro número: ")
9 leia(num2)
10 //operações
11 soma <- num1+num2
12 sub <- num1-num2
13 mult <- num1*num2
14 divi <- num1/num2
15 //exibição das mensagens
16 escreval("Os números digitados foram: ",num1," e ",num2)
17 escreval("A soma entre estes dois números é: ", soma)
18 escreval("A subtração entre estes dois números é: ", sub)
19 escreval("A multiplicação entre estes dois números é: ", mult)
20 escreval("A divisão entre estes dois números é: ", divi)
21 Fimalgoritmo
```


EXERCÍCIO 2

- 2) Faça um algoritmo que receba o preço de custo de um produto e mostre o valor de venda do mesmo. Sabe-se que o preço de custo receberá um acréscimo de acordo com um percentual informado pelo usuário.