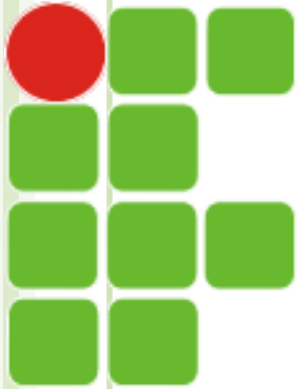


INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA

RIO GRANDE DO NORTE

Campus João Câmara

INTRODUÇÃO A ALGORITMOS

Prof. Alba Lopes

alba.lopes@ifrn.edu.br

http://docente.ifrn.edu.br/albalopes

Introdução

- Em nosso dia dia, utilizamos determinados procedimentos para resolver alguma situação.
  - Ex: trocar um pneu do carro

  - PASSO A PASSO:

1. Levantar o carro com o macaco
2. Remover os parafusos da roda
3. Remover os parafusos da roda
4. Retirar o pneu
5. Colocar o pneu reserva
6. Parafusar a roda
7. Baixar o macaco

Introdução

- Se quisermos fazer um bolo para o lanche da tarde, normalmente seguimos a receita. E isso vale para qualquer outra ação que formos realizar, sempre existem passos a serem seguidos

- Um algoritmo nada mais é do que um conjunto de passos (chamados comandos ou instruções) devem ser seguidos para conseguir resolver umdeterminado problema ou atingir um objetivo.

Introdução

- Um algoritmo para se vestir mal feito pode especificar que você deve primeiro vestir as meias e os sapatos antes de vestir a calça

Exercícios

- Crie um algoritmo que descreva:
  - Como trocar uma lâmpada queimada.

Formas de Representação

- Textual:
  - 1. Pegar uma escada
  - 2. Posicionar a escada
  - 3. Buscar nova lâmpada
  - 4. Subir na escada
  - 5. Remover lâmpada queimada
  - 6. Colocar nova lâmpada
  - 7. Descer da Escada
  - 8. Acionar o interruptor

Como o computador entende a Informação

- Apesar dos computadores de hoje em dia serem máquinas super poderosas que fazem praticamente tudo, eles ainda não conseguem compreender a linguagem do ser humano.

- Você pode estar se perguntando: “Mas como não entende se eu digito tudo em português?”

Como o computador entende a Informação

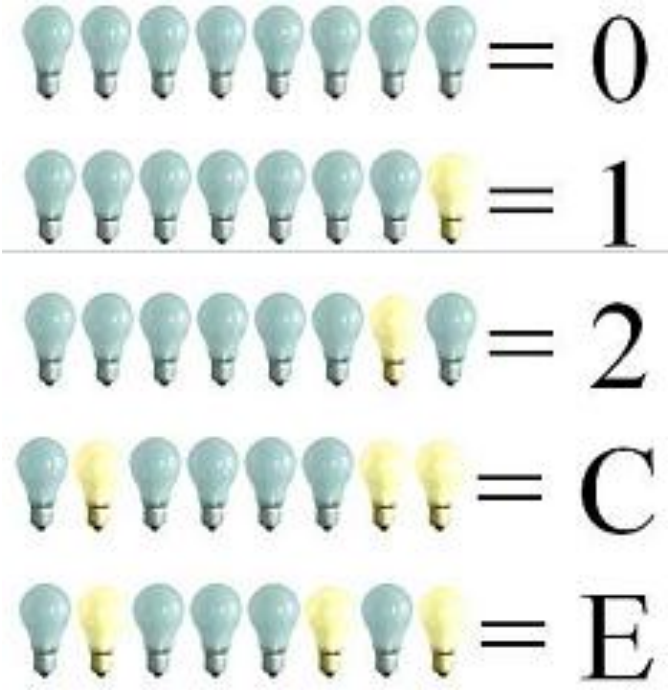
- Na verdade o computador é uma máquina que só compreende duas informações: LIGADO ou DESLIGADO

- É mais ou menos como se fosse uma lâmpada elétrica que você acende ou apaga através de um interruptor.

- Infelizmente com esses dois estados não podemos dizer muita coisa, apenas SIM ou NÃO, VERDADEIRO ou FALSO e isso não é suficiente para conversarmos com a máquina.

Como o computador entende a Informação

- Para conseguir representar outras informações, os cientistas da computação decidiram então agrupar 8 “lâmpadas” e usar as possíveis combinações de “aceso e apagado” para se comunicar com a máquina



Como o computador entende a Informação

- Usamos então os caracteres 0 e 1 para representar apagado e acesso respectivamente.
- É o chamado Sistema Binário.

- Neste sistema de numeração as combinações são escritas como em 00000001, 10101010, 11111111 ou qualquer outra variação possível.

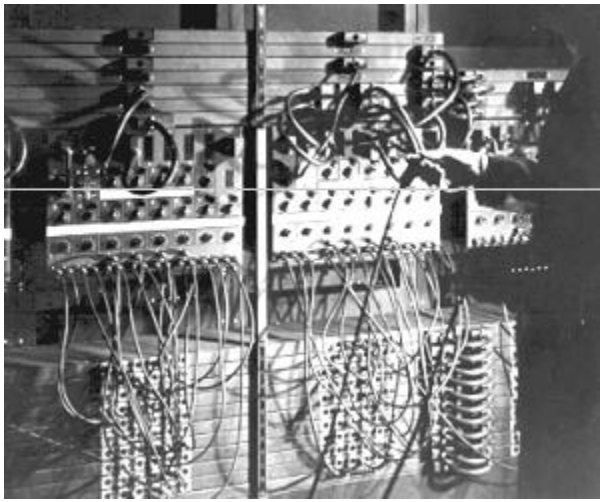
- Cada combinação representa uma letra, número, sinais de pontuação, etc

Como o Computador Entende a Informação

BINÁRIO	CARACTERE	BINÁRIO	CARACTERE
01000001	A	00110000	0
01000010	B	00110001	1
01000011	C	00110010	2
01000100	D	00110011	3
01000101	E	00110100	4
...	...	...	...

Como o computador entende a Informação

- Mas como fazemos para “dizer” essas combinações ao computador?
- Nos primeiros computadores (década de 40) a programação dos computadores era feita através da ligação de cabos entre os conectores disponíveis, algo que não era nada prático e exigia grande atenção e conhecimento técnico.



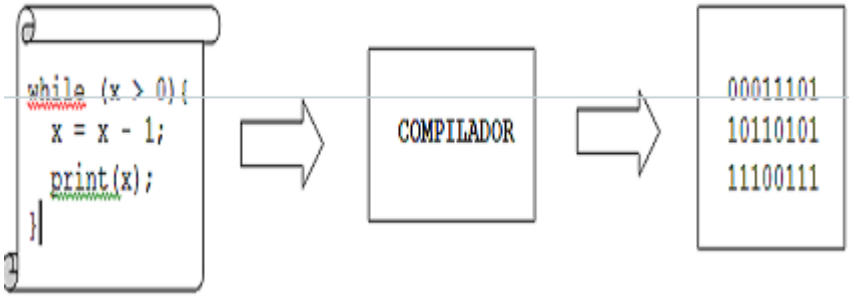
Como o computador entende a Informação

- Felizmente esse tempo já ficou para trás e hoje utilizamos o teclado para digitar os comandos que serão enviados aos computadores.

- Eles continuam entendendo apenas zeros e uns, a grande jogada é que atualmente utilizamos programas que traduzem a linguagem humana para linguagem de máquina, os chamados **compiladores** e **interpretadores**.

Como o computador entende a Informação

- As linguagens de programação são conjuntos de termos e regras que permitem a formulação de instruções para o computador. Geralmente essas instruções são escritas em formato de texto (em inglês na maioria das vezes) e ao carregarmos esses códigos no compilador obteremos um programa em formato binário.



Como o computador entende a Informação

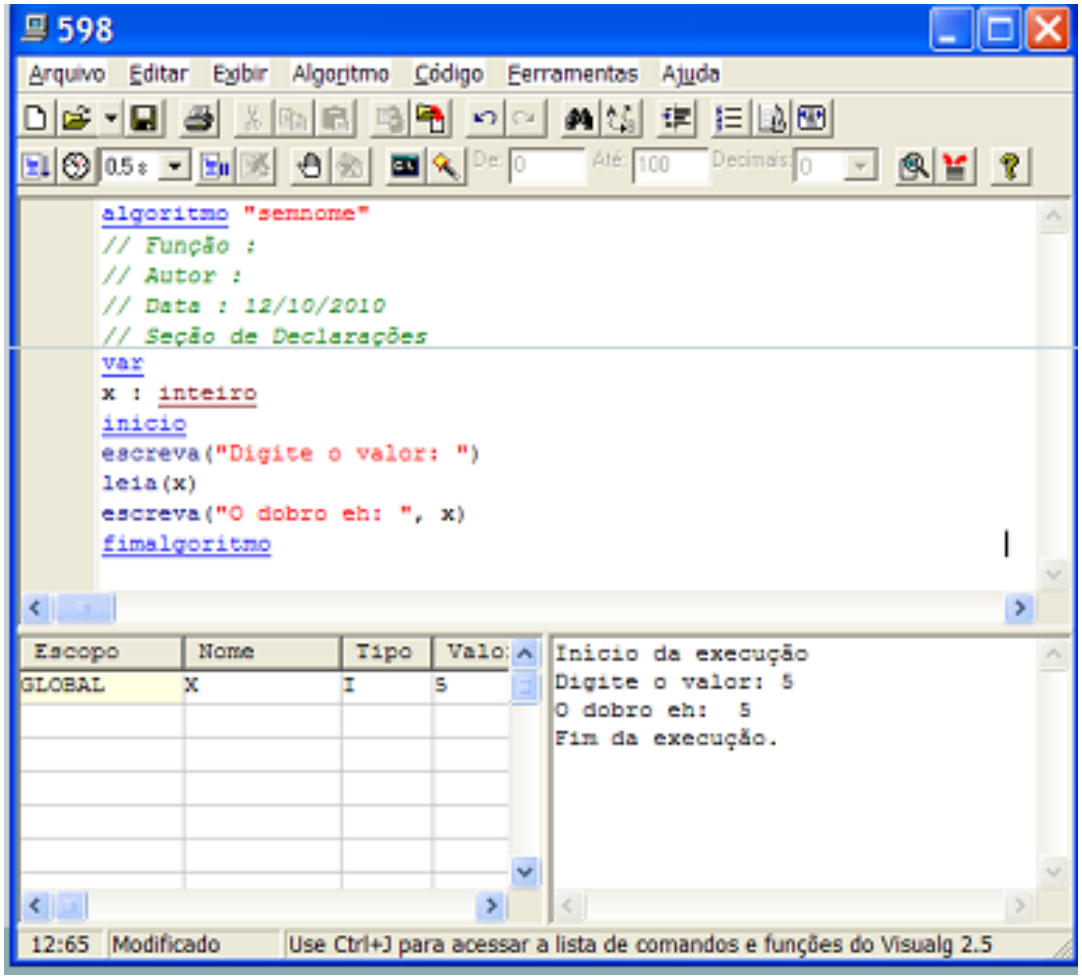
- Algumas outras linguagens que se destacaram na história da computação:
  - ADA, ALGOL, BASIC, CLIPPER, COBOL, FORTRAN, PASCAL, DELPHI, JAVA, VISUAL BASIC, C, C++, e muitas outras.

O VISUALG

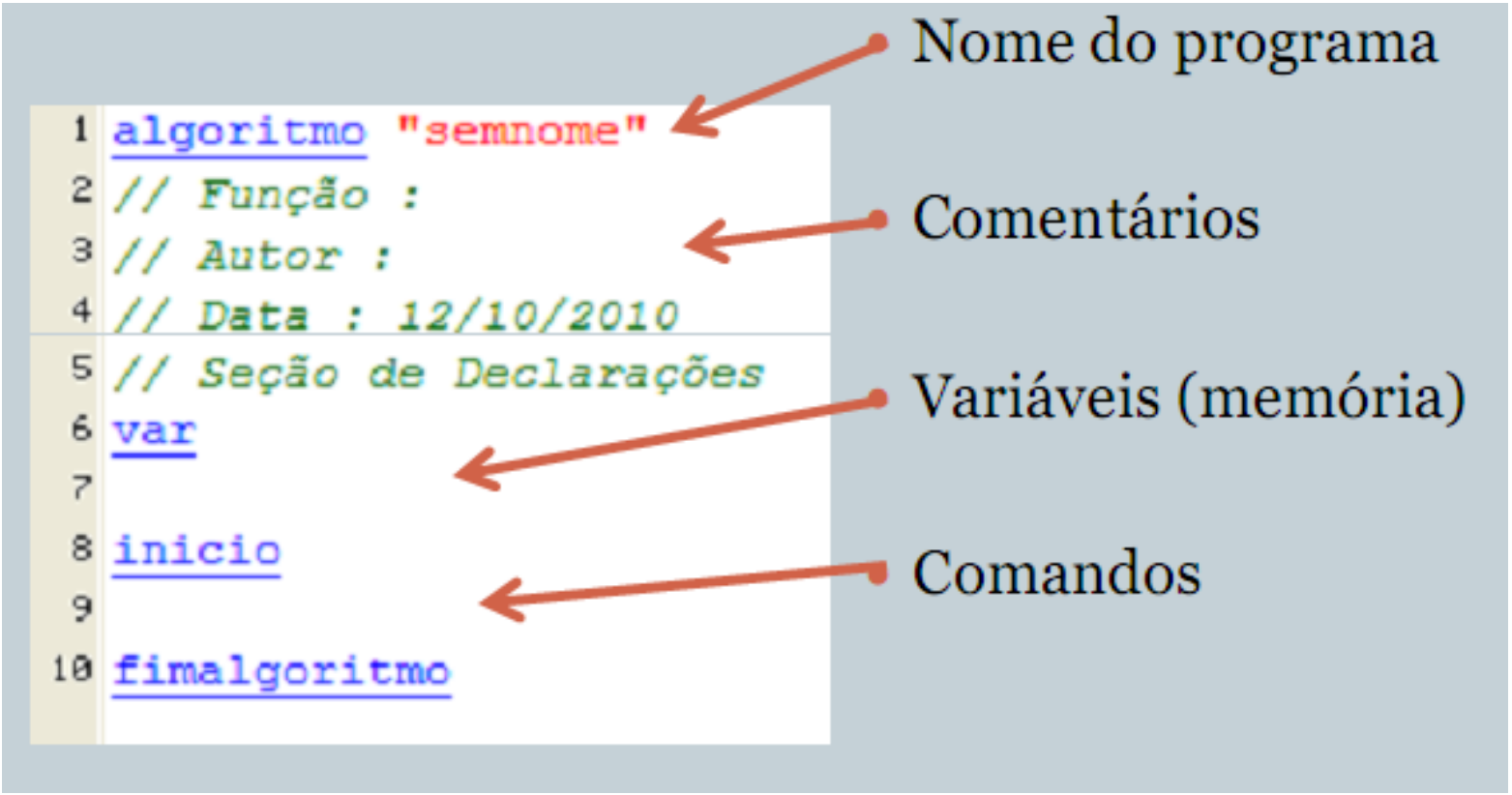
- Editor e interpretador de algoritmos criado pelo professor Cláudio Morgado de Souza.

- É uma ferramenta para os alunos iniciantes em programação exercitarem seus conhecimentos.
  - Possui interface simples.
  - Veja o link no site da disciplina de como baixar o programa.

O VISUALG


598
Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda
algoritmo "semnome"
// Função :
// Autor :
// Data : 12/10/2010
// Seção de Declarações
var
x : inteiro
inicio
escreva("Digite o valor: ")
leia(x)
escreva("O dobro eh: ", x)
fimalgoritmo
Escopo Nome Tipo Valo
GLOBAL X I 5
Inicio da execução
Digite o valor: 5
O dobro eh: 5
Fim da execução.
12:65 Modificado Use Ctrl+J para acessar a lista de comandos e funções do Visualg 2.5

Estrutura de um Algoritmo




```
1  algoritmo "semnome"
2  // Função :
3  // Autor :
4  // Data : 12/10/2010
5  // Seção de Declarações
6  var
7
8  inicio
9
10 fimalgoritmo
```

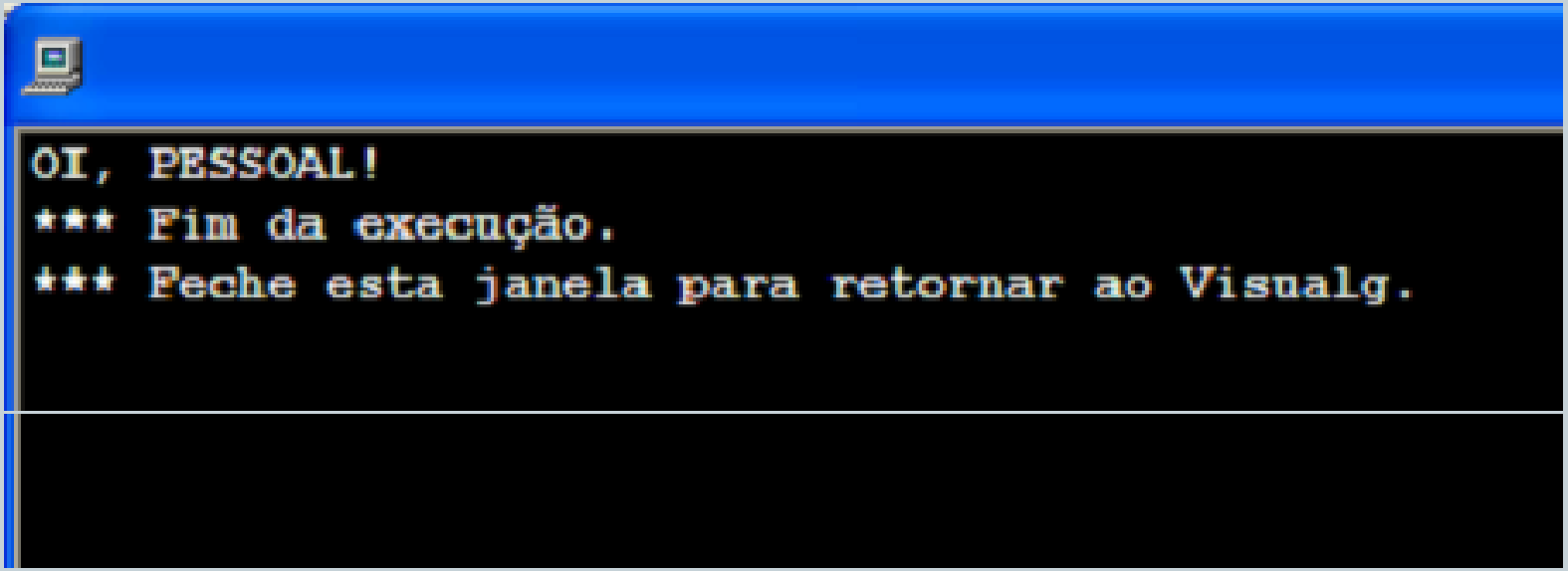
Escrevendo na Tela

- Primeiro programa

```
1 algoritmo "primeiro"
2
3 var
4
5 inicio
6
7 escreva ("OI, PESSOAL!")
8
9 fimalgoritmo
```

- O comando **escreva** mostra na tela o texto que estiver dentro do parênteses.
- O texto obrigatoriamente deve vir entre aspas duplas.
- Para executar pressione **F9**.

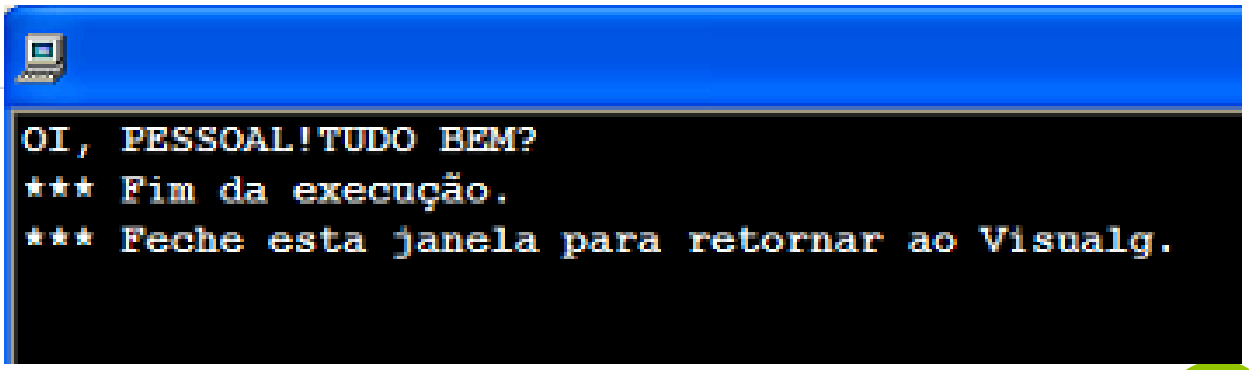
Resultado da Execução



Escrevedo Texto na Tela

- Podemos utilizar vários comandos **escreva** para mostrar diversas frases na tela

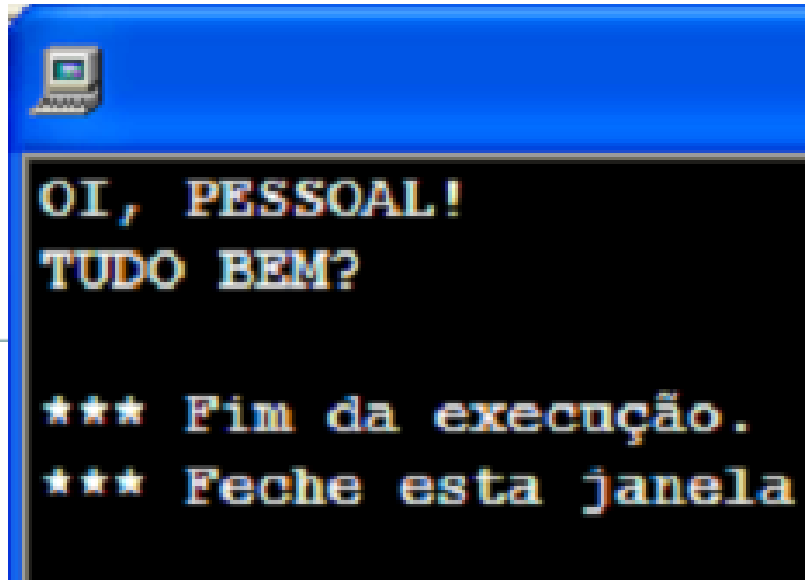
```
1  algoritmo "segundo"
2
3  var
4
5  inicio
6
7  escreva ("OI, PESSOAL!")
8  escreva ("TUDO BEM?")
9
10 fimalgoritmo
```



Escrevedo Texto na Tela

- Observe que escreva coloca o texto sempre na mesma linha. Se quisermos colocar cada frase numa linha diferente usamos escreval que salta para a próxima linha após exibir o texto:

```
1  algoritmo "terceiro"
2
3  var
4
5  inicio
6
7  escreval ("OI, PESSOAL!")
8  escreval ("TUDO BEM?")
9
10 fimalgoritmo
```



Estrutura Básica de um algoritmo

- Todo algoritmo tem um título, e ele deve estar entre aspas duplas na linha 1
- Da linha 2 a linha 5: comentários
  - Comentários são todo o texto que aparece após as duas barras "//" (sem aspas)
  - Os comentários servem para descrever algo no código
  - Os comentários não tem efeito sobre o código

Exercícios

1. Crie um programa que escreva seu nome em única linha.

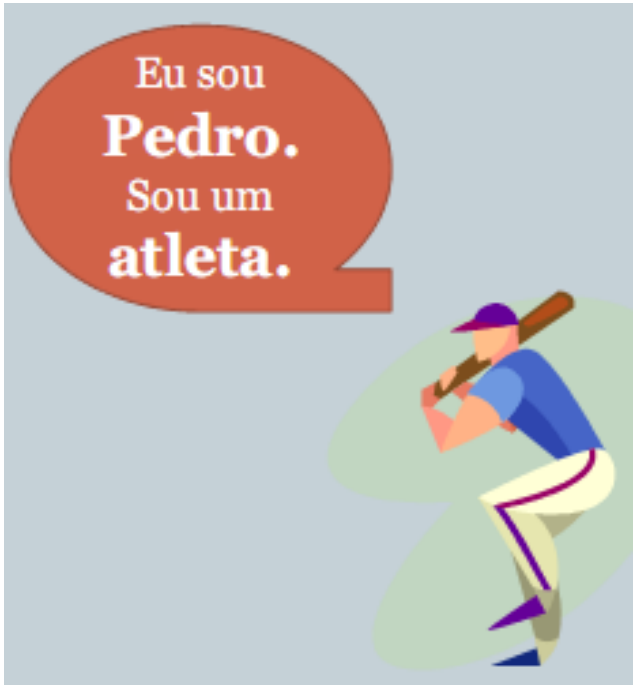
2. Crie um programa que escreva seu nome numa linha e o sobrenome em outra linha.

Variáveis

- Não tem graça um programa que só mostra texto.
- O ideal é que possamos digitar informações que serão
- processadas pelo computador.
- Para tal, precisamos de um lugar na memória onde as informações que digitamos sejam guardadas.
- A este lugar na memória damos o nome de variável.

Variáveis

- Cada variável precisa de um nome para identificá-la e de um tipo.



Variáveis

- Assim como não o nome de uma pessoa não pode ser “132&$!”, as variáveis também possuem algumas regras para sua nomeclatura.
  - Devem sempre iniciar com uma letra ou _
  - Não devem ter caracteres especiais (*, +, !, #, etc...)
  - Não devem possui espaço
  - Devem representar o valor nelas contidos
  - Tem limite de tamanho (não pode ultrapassar 30 caracteres)
- **Exemplos válidos:** nome, valor1, total, x2
- **Exemplos inválidos:** 1x, $legal, tot*al

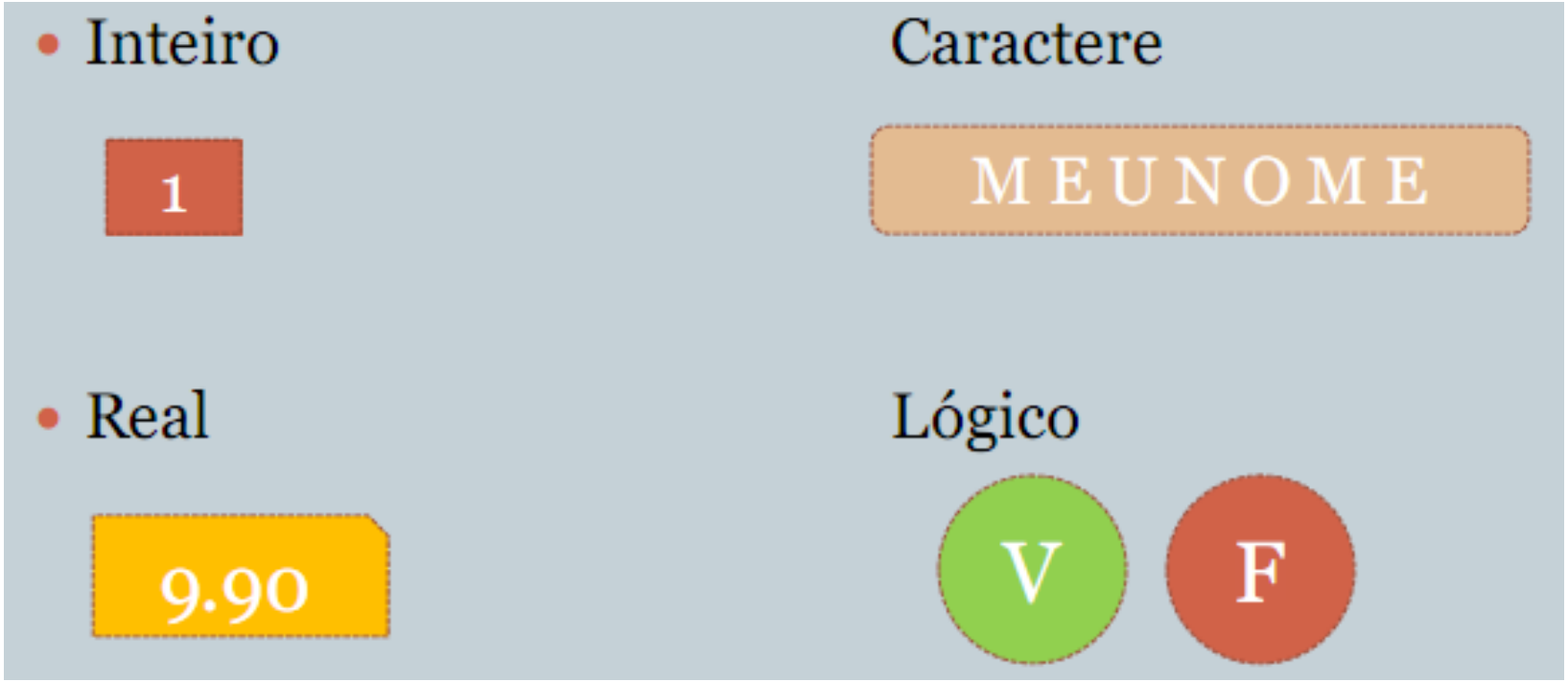
Exercício 2

- Identifique nomes válidos de variáveis:

1. $Salário
2. Salário$
3. A[1]
4. a>b
5. a+b
6. xKH
7. Alfa2
8. 2Vizinhos
9. Val0r
10. valOr
11. Qwert
12. guarda_chuva
13. U.F.
14. Diaadia
15. betateste

Variáveis

- Poderíamos representar visualmente assim:



Variáveis

- Observe que cada tipo é diferente, então não devemos tentar colocar um valor real dentro de um inteiro, ou somar duas variáveis caractere.
- No Visualg, as variáveis devem ficar na seção var
- são declaradas da seguinte maneira:

```
nome_da_variavel : tipo
```

```
3  var
4
5  valor1: inteiro
6  total: real
7  nome: caractere
8  achou: logico
9
10 valor2, valor3 : inteiro
```

Variáveis

- Com a variável criada, é possível dar um valor para ela. Para atribuir um valor para uma variável, use um dos sinais de atribuição: ":=" ou "<-"

```
inicio
idade := 26
nome := "Fernando Carneiro"
saldo <- 204.3
sinal <- VERDADEIRO
```

VARIÁVEIS

- Com a variável criada, é possível dar um valor para ela de duas formas:
  - **Por atribuição**: através dos sinais ":=" ou "<-"

```
var
   idade: inteiro
   nome: caractere
   saldo: real
   sinal: logico
inicio
   idade <- 26
   nome <- "Fernando"
   saldo <- 204.3
   sinal <- verdadeiro
fimalgoritmo
```

OU

```
var
   idade: inteiro
   nome: caractere
   saldo: real
   sinal: logico
inicio
   idade := 26
   nome := "Fernando"
   saldo := 204.3
   sinal := verdadeiro
fimalgoritmo
```

VARIÁVEIS

- O comando **escreva** pode também mostrar o **valor** da variável que está dentro dos parênteses. Neste caso, não usamos aspas

```
inicio
   idade <- 16
   escreva(idade)
fimalgoritmo
```

- Podemos misturar e escrever texto e valores de variáveis. Nesse caso, devemos separar por vírgula

```
inicio
   idade <- 16
   escreva("A idade é: ", idade)
fimalgoritmo
```

VARIÁVEIS

- **Atribuição de valor à variável**
  - **Por digitação do usuário**: através do comando **leia** podemos digitar qualquer valor na variável

```
inicio
    leia(idade)
    escreva("A idade é: ", idade)
fimalgoritmo
```

O valor não foi digitado no código. Cada vez que o algoritmo executar, podemos digitar valores diferentes

Exercício 2

1. Faça um programa que peça para digitar o nome completo e depois mostre-o na tela

2. Faça um programa que peça para digitar primeiro nome, depois o sobrenome separadamente e mostre o nome completo de uma vez só.

Operadores Matemáticos

- Podemos fazer cálculos com valores **inteiro** ou **real**
- Os operadores usados são

Sinal	Descrição
+	Operador aritmético tradicional de soma.
-	Operador aritmético tradicional de subtração.
*	Operador aritmético tradicional de multiplicação.
/	Operador aritmético tradicional de divisão. O resultado pode ser um número real se a divisão não for exata.
\	Operador de divisão inteira. O resultado sempre é um número inteiro. Exemplo: 5 \ 2 = 2
%	Operador de módulo ou resto da divisão. Exemplo: 8 % 3 = 2
^	Operador de potenciação. Exemplo: 5 ^ 2 = 25

Operadores Aritméticos

- O cálculo se dá da seguinte forma:

```
algoritmo "operadores"
var
   total: inteiro
inicio
   total <- 5 + 5
   escreva(total)

   total <- 5 - 5
   escreva(total)

   total <- 5 * 5
   escreva(total)
fimalgoritmo
```

Operadores Aritméticos

- Assim como na matemática, o Visualg também considera a precedência dos operadores, ou seja, multiplicação e divisão são resolvidos antes de soma e subtração:
- Ex:
  - Qual o resultado de 2 + 2 * 5 ?
- Resposta: 12, pois 5 * 2 é resolvido antes de somar 2 + 2.

Operadores Aritméticos

- Podemos usar parênteses para resolver ambiguidades.
- A expressão anterior ficaria mais clara da seguinte forma

```
total <- 2 + (2 * 5)
```

Exercício

- Qual o resultado das seguintes operações?
  a) 2 * 3 + 5
  b) 2 + 6 / 2
  c) 5 * 8 + 4 / 2

Operadores Matemáticos

- Podemos usar o comando leia para digitar números que serão usados nos nossos cálculos.
- Observe o exemplo de um somador de dois números:

```
algoritmo "somar_dois_numeros"
var
   valor1, valor2, total: inteiro
inicio
   leia(valor1)
   leia(valor2)
   total <- valor1 + valor2
   escreva(total)
fimalgoritmo
```

Referências

- Slides de aula do Prof. Abrahão Lopes
  - http://docente.ifrn.edu.br/abrahaolopes
- Slides de aula do Prof. Jalerson Lima
  - http://www.jalersonlima.com.br