Disciplina: Lógica de programação
Professora: Sandra Hedler de Amorim
E-mail: sandra-famorim@educar.rs.gov.br

Objetivos da Aula

Breve Revisão de Algoritmos

A seguir a estrutura de um algoritmo pseudocódigo ou metalinguagem.

```
Algoritmo "nome"

Var
    //declarações de variáveis

Inicio

    //Corpo do algoritmo

fimalgoritmo
```

A área da *declaração de variáveis* é onde vamos prever todas as variáveis que serão manipuladas pelo algoritmo.

Delimitado por **inicio** e **fimalgoritmo** é onde vamos ter o corpo principal do algoritmo, onde vamos colocar todas as instruções que serão executadas sequencialmente para a princípio solucionar o problema proposto.

Tipos primitivos de dados que podem ser representados pelas nossas variáveis:
- inteiros
- real
- caracter
- logico

Como vamos diferenciar quando vai ser de um determinado tipo um dado qualquer? Devemos saber interpretar dentro do problema proposto qual o

tipo de determinada informação ou buscar do teclado ou gerar a partir de determinada informação.

Por exemplo:
1) Idade de uma pessoa - tipo numérico inteiro
2) Quantas balas vou ter no fim de uma semana, se como 2/dia - tipo numérico inteiro.
3) Peso, altura, largura - todo valor de medida primitiva, valor monetário são do tipo real.
4) Caractere, é todo aquele dado tipo literal, seja ele alfanumérico, seja A a Z maiúsculo ou minúsculo, caracteres especiais. Exemplos: nome de uma pessoa, endereço, CPF, telefone, placa de veículo etc.
5) O tipo primitivo logico, é aquele biestável, que só armazena valores(0/1) ou seja, verdadeiro ou falso. Armazena resultados de expressões lógicas ou relacionais.

Sintaxe para declarações de variáveis

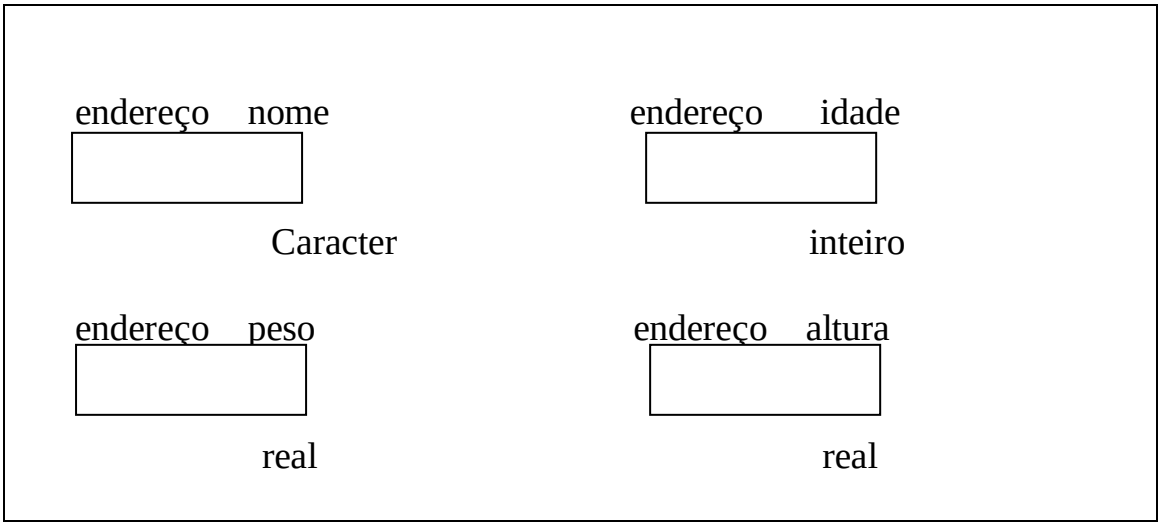
```
Var
    nome_da_variavel: tipoprimitivo
    lista_de_variaveis: tipoprimitivo
```

Exemplos:

```
var
    nome: caracter
    idade: inteiro
    peso, altura: real
```

Quando declaramos variáveis estamos alocando um endereço de memória onde a informação vai ficar efetivamente armazenada para ser utilizada posteriormente em tempo de execução do meu algoritmo.

Memória RAM



Quando declaramos nome: caracter estamos dizendo ao computador para alocar um endereço de memória e referenciar ele pela variável *nome* e nesse endereçamento de memória só vai armazenar valor do tipo caracter.

Como vamos dar valores a essas variáveis?
Existem duas formas, ou damos a partir de *atribuição de dado* ou a partir de *comandos de entrada*.

Sintaxe para atribuição de dados

variavel := expressão
ou
variavel <- expressão

Exemplos:

nome := "Geltrudes"
idade := 90
peso := 56.500

Toda vez que chamar(referenciar) a variável *nome* dentro do meu algoritmo, estou manipulando diretamente o conteúdo desse espaço de memória que é Geltrudes.
Devemos lembrar que cada variável armazena um valor a cada momento, as chamadas variável simples, mais adiante vamos trabalhar com as chamadas variáveis compostas.

Fluxo de Dados

Vamos ver a partir de agora os dois comandos de entrada e saída de dados que nos permite fazer a comunicar do computador com o mundo externo e do mundo externo com o computador, ou seja, a interação homem máquina.
Dentro do fluxo de dados, temos dois comandos principais, o comando de entrada e o comando saída de dados.

Entrada de Dados:
Sintaxe:

```
leia(variavel)
```

Exemplos:

```
leia(nome)
leia(idade)
```

Saída de Dados:
Sintaxe:

```
escreva("CadeiadeCaracteres")
escreva("CadeiadeCaracteres",variavel)
```

Exemplos:

```
escreval("Qual seu nome? ")
escreva("Qual sua idade? ")

escreva("Seu nome é ",nome)
escreva(nome," você está com ",idade," anos")
```

O **comando de entrada**, possibilita usar uma variável ou uma lista. O ideal é ler uma variável a cada momento. Esse comando leia é o responsável por colocar um ponto intermitente (|) na tela do computador que indica que é a vez do usuário fornecer o valor e teclar ENTER para que o computador

capture e armazenar esse valor no endereçamento de memória que está sendo referenciado pela variável no comando leia.

O **comando escreva** é o responsável por mostrar na tela do computador do usuário as informações, mensagens, seja elas informações que o programa precisa ou informações com resultados de saída ou aleatórias.

Seja ele o comando escreva ou escreval, os dois tem essa funcionalidade, com a única diferença que o ***escreva*** dispara na tela a informação, porém não efetua a quebra de linha, já o comando ***escreval*** mostra mensagem e efetua ao final a quebra de linha, como se desse um ENTER ao final da impressão da mensagem na tela do dispositivo do usuário.

Quando colocamos a linha de instrução escreva("Qual seu nome? ") está disparando para a tela do usuário a mensagem "Qual seu nome? " questionando o usuário. E a partir desse comando de saída deve vir a seguir uma linha de instrução com comando de entrada de dados que captura esse dados que o usuário responderá e armazena em um endereço de memória que será utilizada posteriormente.

Quando colocamos no código do algoritmo a seguinte instrução,

escreval(nome," você está com ",idade," anos")

estamos disparando para tela do usuário e quebrando a linha ao final. A palavra nome dentro do comando escreval(), significa uma variável ou seja, um endereço reservado de memória onde estamos dizendo para o computador ir nesse endereço e trazer e imprimir na tela do usuário o conteúdo armazenado nesse endereço chamado nome. Ou seja, sempre que tiver dentro do comando escreva ou escreval uma palavra que *não* esteja entre aspas (" ") significa uma variável, que estamos referenciando esse endereço de memória, queremos que mostre na tela o conteúdo armazenado nesse endereço.
Então no comando acima estamos solicitando o conteúdo da variável nome, seguido da cadeia de caracteres, seguido do conteúdo da variável idade e seguido de outra cadeia de caracteres. E observem que cada variável e cadeia de caracteres é separado por vírgula, que significa concatenação, se não colocarmos está gerando um erro de sintaxe.

A seguir um exemplo de um algoritmo:

```
algoritmo "exemplo"

var
        nome: caracter
        idade: inteiro

inicio
        escreva("Qual seu nome? ")
        leia(nome)
        escreva("Qual sua idade? ")
        leia(idade)
        escreval(nome," você está com ",idade," anos")

fimalgoritmo
```

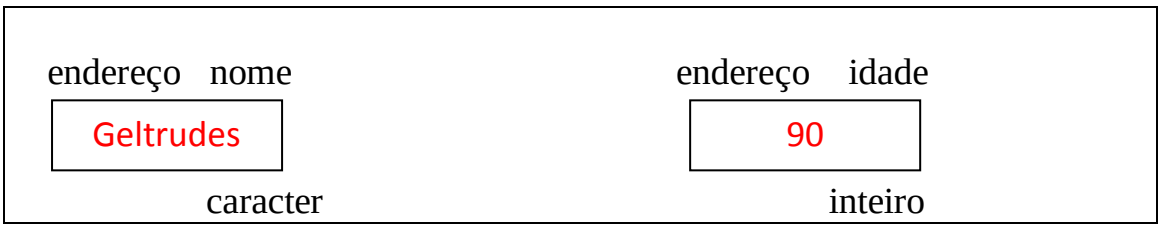
A estrutura básica do algoritmo são todas as palavras que estão impressas em letra azul negritadas (algoritmo, var, inicio e fimalgoritmo) no exemplo acima.
Um algoritmo sempre começa pela palavra reservada *algoritmo* seguido pelo nome para diferenciar dos demais.
O nome do algoritmo devemos sempre começar com uma letra ou uma sublinha e não colocarmos caracteres especiais. E se for composto por duas ou mais palavras devemos unificar essas palavras para compor o nome do algoritmo.
Essa regra para dar nome ao nosso algoritmo também se enquadra para dar nome as nossas variáveis.
Após a palavra reservada **var** temos declarados 2 variáveis:
nome do tipo caracter e idade do tipo inteiro.
Após a palavra reservada **inicio** vamos nos comunicar com o usuário. E como realizamos essa comunicação? Com o comando escreva e na sequencia o comando leia.

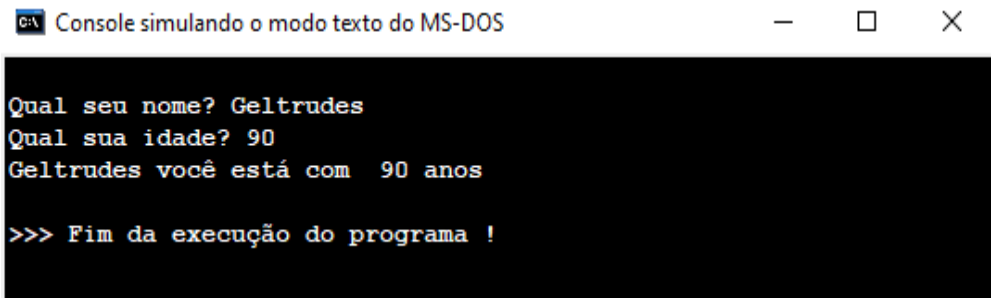
```
        escreva("Qual seu nome? ")
        leia(nome)
```

O comando dispara a mensagem para o usuário responder a pergunta e ao digitar e dar ENTER o comando leia recebe e armazena na variável nome que foi declarada para receber, armazenar esse valor informado pelo usuário.
Agora como vamos saber se está certo esse nosso algoritmo?

Para saber se está correto existe o chamado teste de mesa, que é uma simulação do que acontece na tela do usuário e na memória do computador enquanto esse o algoritmo estiver em tempo de execução.
Exemplo:

Memória Ram





Quando o algoritmo chega ao final, na palavra reservada fimalgoritmo, esses endereços de memória são excluídos, ou seja, não existem mais e podemos fechar o nossa tela de exibição.