**Introdução**

Esse documento tem como objetivo conceituar e servir como consulta sobre os seguintes assuntos: estrutura condicional SE, operadores lógicos OU e E e operadores relacionais IGUAL(=), MAIOR(>), MENOR(<), MAIOR IGUAL(>=), MENOR IGUAL(<=) e DIFERENTE(<>). Qualquer dúvida que possuir consulte este documento e se não conseguir saná-la, consulte um orientador.

Além disso, o documento utilizará as seguintes cores como forma de facilitar o entendimento, a localização e sobre o que está sendo falado:

*Estrutura condicional SE :* 

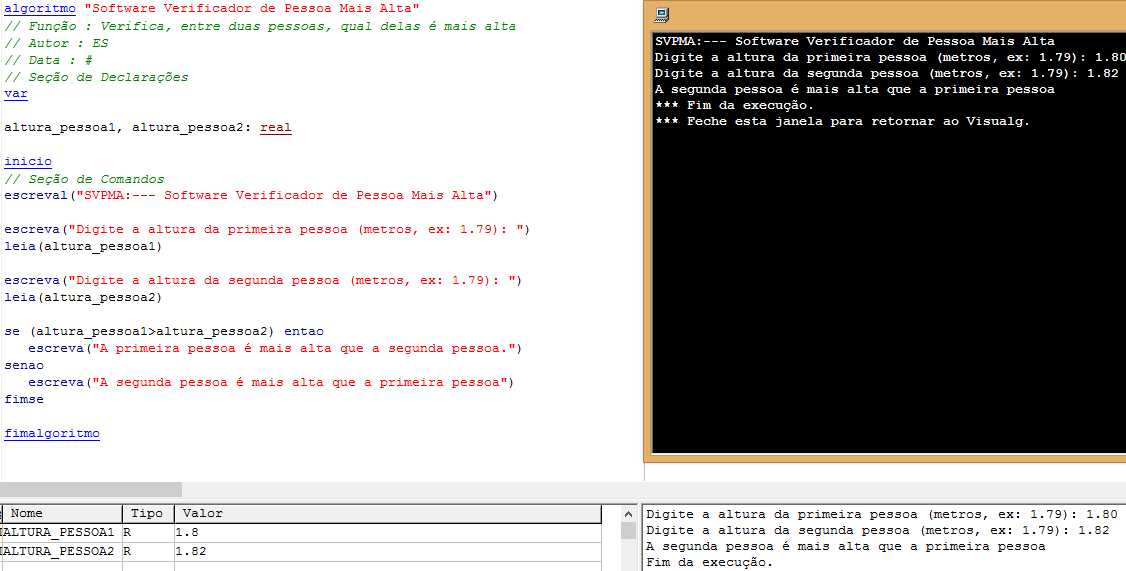
*Operadores Lógicos :*

*Operadores Relacionais :*

**Estrutura condicional SE**

Essa estrutura, como o próprio nome já indica, trabalha condições através das instruções SE, SENAO SE e SENAO. Isso serve para possibilitar caminhos diferentes dependendo de uma condição, seja ela verdadeira ou falsa. Trazendo isso para o mundo real, se sua idade for maior que 18, você é maior de idade, senão você é menor de idade.

Com o uso dessas instruções o programa consegue ter caminhos diferentes e realizar diferentes tarefas dependendo das informações e das condições. Porém, para conseguir identificar o caminho que o programa deve tomar, as condições precisam fazer algum tipo de comparação. Assim é que aparecem os operadores. Um exemplo prático simples: supondo que eu queira verificar se você é mais alto(a) que outra pessoa, eu preciso da sua altura, da altura da outra pessoa, e?... verificar se a sua altura é maior que a altura da outra pessoa. Se essa condição for verdadeira a resposta é: você é maior, senão (ou seja, se a condição se é falsa): a outra pessoa é maior. Isso é apenas um exemplo, dentro de um apelo didático, porém já dá para perceber que podem ser realizadas várias verificações através dessa instrução. Veja abaixo como ficaria escrito esse código no VisuAlg:



Crie esse código no VisuAlg para poder testar. Assim que terminar, vamos agora para a estrutura SE do código:

se (altura\_pessoa1>altura\_pessoa2) entao

escreva("A primeira pessoa é mais alta que a segunda pessoa.")

senao

escreva("A segunda pessoa é mais alta que a primeira pessoa")

fimse

Perceba que todo se tem um entao (sem acento), e que a condição vai entre parênteses. O uso do operador relacional é idêntico ao uso normal na matemática (apenas o operador de diferente [<>] que difere a escrita).

Agora analise o exemplo usado e faça um teste colocando as alturas iguais. O algoritmo informou que a segunda pessoa é mais alta, porém isso não é verdade. O que pode ser feito para corrigir esse problema? Após verificar se sua altura é maior que a segunda, verificar se a altura das pessoas é igual, e se for informar que as pessoas possuem a mesma altura, senao mostrar que a segunda pessoa é mais alta. Siga o script abaixo:

se (altura\_pessoa1>altura\_pessoa2) entao

escreva("A primeira pessoa é mais alta que a segunda pessoa.")

senao

se (altura\_pessoa1=altura\_pessoa2) entao

escreva("As duas pessoas são do mesmo tamanho")

senao

escreva("A segunda pessoa é mais alta que a primeira pessoa")

fimse

fimse

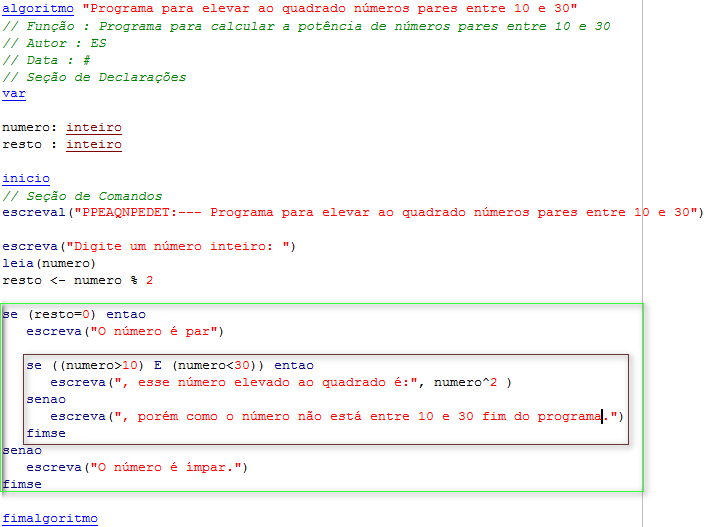
Teste e verifique se está funcionando. Agora, perceba que não verificamos se a segunda pessoa é maior que a primeira para poder afirmar na última condição, mas vamos usar a lógica…





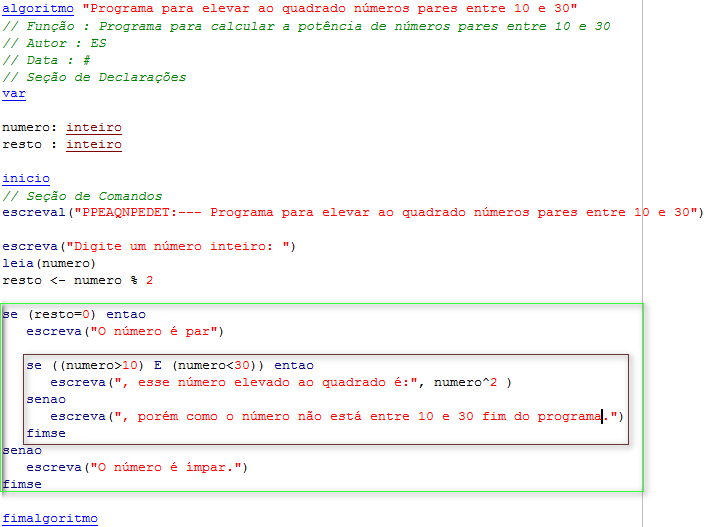
Portanto, agora sim podemos afirmar que na última condição a pessoa 1 é menor que a pessoa 2.

Além de usar a estrutura se em “cadeia” usando um senao se,a estrutura SE pode ser utilizada dentro de outra estrutura SE. Veja um exemplo abaixo:



Esse tipo de situação chama-se SE ANINHADO, ou seja, um SE dentro de outro SE. E quando deve-se usar um SE ANINHADO? R: quando precisar, isso depende muito da regra de negócio e da solução que você vai dar à ela. Além disso, na imagem é possível visualizar um E, que é um operador lógico, assim como o operador OU e o operador NAO.

Falando mais profundamente sobre os operadores lógicos, falaremos primeiro do operador E. Basicamente esse operador deve ser utilizado quando analisando DUAS condições, precisamos que as DUAS sejam VERDADEIRAS para que aconteça determinada rotina. Analisando novamente o exemplo acima, somente se a variável **numero** for maior do que 10 E também for menor do que 30, é que a instrução dentro do SE será executada:



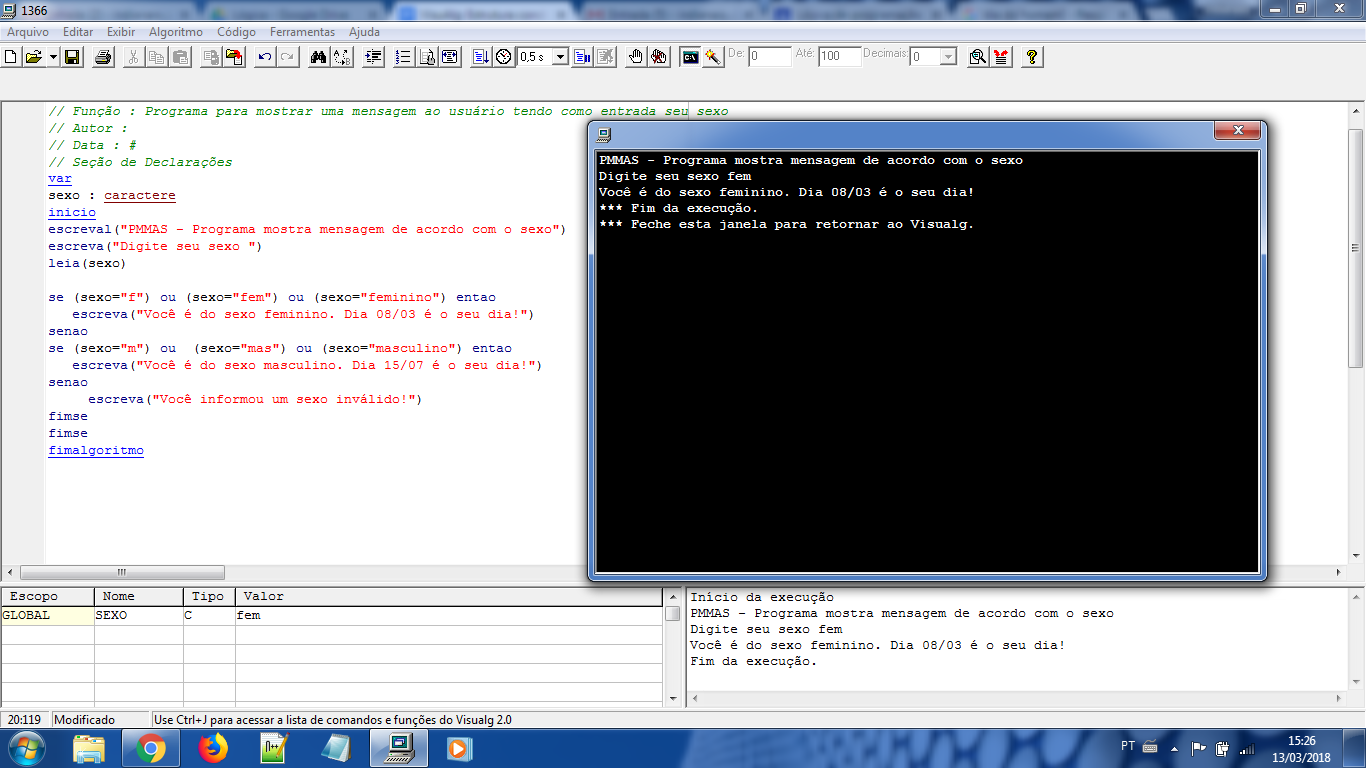
Não basta que apenas uma delas seja verdadeira para que o programa entre na condição. Observe que, as condições que testamos foram “envolvidas” em um parênteses (cada uma delas) e ainda, as duas envolvidas em outro. Na utilização de operadores lógico é imprescindível a utilização de parênteses para indicar o grupo de condições que devem ser analisadas.

***Um exemplo do mundo real:*** se te derem a tarefa de ir até o supermercado e comprar leite E ovos, você deve trazer ambos. Caso você traga apenas leite OU apenas ovos, a tarefa não foi concluída/atendida.

Para falar sobre o próximo operador, o operador lógico OU, podemos já continuar na linha de raciocínio do mundo real: se te derem a tarefa de ir até o supermercado e comprar leite OU ovos, você pode trazer apenas leite, OU apenas ovo, que sua tarefa estará concluída/atendida. No entanto, se você não trouxer nenhum dos dois, a tarefa também não estará concluída.

Trazendo o uso do operador OU para a resolução de um problema computacional, podemos solicitar que o usuário digite seu sexo e devolvamos uma mensagem no caso do sexo ser feminino e outra mensagem no caso do sexo ser masculino. No entanto, você concorda que o usuário poderá digitar seu sexo de variadas maneiras? Por exemplo, para o sexo feminino: “f”, “F‘, “fem‘, “FEM”, “feminino”, “FEMININO”. São 06 possibilidades diferentes somente no exemplo em questão!

O VisuAlg não diferencia letras maiúsculas e minúsculas, porém, várias outras linguagens de programação o fazem e é sempre uma boa prática prevermos as mais variadas entradas de dados que o usuário pode nos fornecer para um mesmo caso. No exemplo dado, podemos apenas considerar as variações: *“f”, “fem” e “feminino”*, uma vez que *“F”, “FEM”, ou “FEMININO”* são respectivamente equivalentes às três primeiras formas descritas. Veja na prática esse exemplo:



Nesse caso, se o usuário digitar uma OU outra condição testada, a condição toda...

**(sexo="f") ou (sexo="fem") ou (sexo="feminino")**

...já é verdadeira. Ainda, observe que nesse caso, não faz sentido utilizarmos o operador E, de modo que a condição fosse:

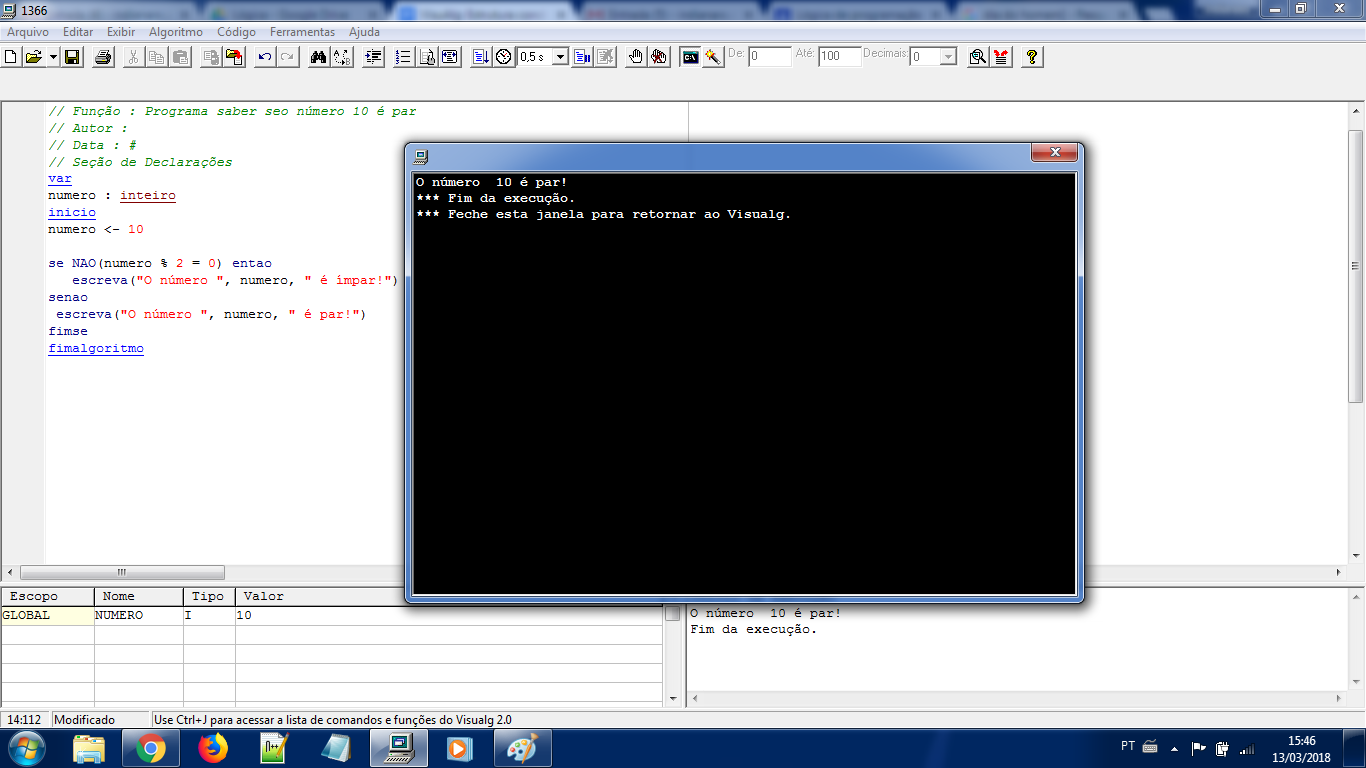
**(sexo="f") e (sexo="fem") e (sexo="feminino")**

Não há como o usuário entrar com um valor que seja ao mesmo tempo “f” e ao mesmo tempo “feminino”, certo? OU é um, OU é outro, concorda? Diferente do exemplo que utilizamos para o operador lógico E, pois, o número pode ser ao mesmo tempo maior do que 10 e menor do que 30, certo? Como por exemplo o número 15.

Por fim, vamos falar do operador NAO, talvez não tão utilizado quanto os demais, e você já entenderá o motivo. O operador NAO simplesmente inverterá o resultado de uma expressão lógica para o seu contrário. Ou seja, se expressão for verdadeira e inserirmos o operador NAO em frente a ela, automaticamente invertemos o retorno para FALSO.

Contextualizando, imagine que você deseja saber se um número **é par**, você poderá testar se o resto da divisão dele por dois é IGUAL A zero, correto? Normalmente fizemos esse teste, assumindo que se o resultado de tal expressão for VERDADEIRO, o programa fará determinada rotina.

Mas, você também pode testar se resto da divisão dele por dois NAO é IGUAL A zero e aí, descobrir que ele **não é par**, e sim, que ele é **ímpar**, concorda? Vamos ver o exemplo prático dessa última situação:



Percebeu o porquê dissemos que esse operador não é tão recorrentemente utilizado? Nós normalmente testamos se as condições das quais esperamos que sejam verdadeiras, sejam de fato verdadeiras. SENAO forem, fizemos a rotina pertinente. Trazendo para um **exemplo do mundo real,** se você quer saber se um homem já pisou na lua, você normalmente pergunta “Um homem já pisou na lua?” ao invés de “Nunca um homem pisou na lua?” para obter como resposta “não” e inverter essa resposta para descobrir que “sim”, um homem já pisou na lua.

Portanto, pode-se perceber que o uso de estruturas SE são muito importantes e com elas podem-se definir vários caminhos, indicando condições, para trilhá-los. E como essas condições precisam de fatores para comparação, entram os operadores lógicos e relacionais, que fornecem uma grande gama de possibilidades para comparações.

**Obs.: para saber mais sobre os operadores disponíveis no VisuAlg:** [**http://www.apoioinformatica.inf.br/produtos/item/09-operadores**](http://www.apoioinformatica.inf.br/produtos/item/09-operadores)