**Matrizes Triangulares**

**Mário Leite**

...

Como já coloquei aqui, em tom de brincadeira: “*A codificação é apenas um avatar da programação*”. O que eu quis dizer com esta frase é que, embora a codificação seja a etapa final do processo, a solução do problema é dada na etapa da programação, pela lógica do algoritmo que a implementa. Nas aplicações práticas, como também já enfatizei em outras publicações, as matrizes são fundamentais; e um exemplo disto é a “triangulação de matrizes”, que pode ajudar na solução de sistemas de equações (escalonamento), cálculo de determinantes, eliminação de Gauss, etc.

Apenas uma rápida explicação: *matriz triangular é aquela em que os elementos que estão fora da diagonal principal são todos zeros*. Se todos esses elementos *zeros* estiverem acima da diagonal, ela é uma “Matriz Triangular Inferior”; se os *zeros* estiverem todos abaixo da diagonal é uma “Matriz Triangular Superior”. E no caso particular, de todos os elementos acima e abaixo da diagonal forem todos *zeros* a matriz é dita “Matriz Diagonal”, se houver, pelo menos um elemento da diagonal for diferente de zero.

Então, decidi criar uma solução em que o usuário pudesse escolher e imprimir as ***n*** primeiras matrizes triangulares: *inferiores* ou *superiores*.

Como eu fiz!!?? Pequei lápis e papel (nada de computador) e comecei a rabiscar os quatro primeiros exemplos de matrizes triangulares superiores quadradas: 2*x*2, 3*x*3, 4*x*4 e 5*x*5. Então, observando os *layouts* delas, notei que havia um padrão sequencial na quantidade de elementos *zeros* abaixo da diagonal, como mostrado abaixo:

**2 \* 2 ==> 1**

**3 \* 3 ==> 3**

**4 \* 4 ==> 6**

**5 \* 5 ==> 10**

Observem que o número de elementos *zeros* **X** (1,3,6,10,...n) abaixo da diagonal estava ligado diretamente ao número de elementos da diagonal. Então, foi fácil notar que:

**Matriz 2x2: X = (2\*2 - 2)/2 = 1**

**Matriz 3x3: X = (3\*3 - 3)/2 = 3**

**Matriz 4x4: X = (4\*4 - 4)/2 = 6**

**Matriz 5x5: X = (5\*5 - 5)/2 = 10**

A **figura 1** mostra o rabisco (planejamento) da solução do programa, onde cheguei à fórmula gera; em seguida o pseudocódigo da solução. Deste modo, a fórmula geral que deduzi foi a seguinte:

**X = (m^2 - m)/2**, onde **m** é o número de linhas da matriz e **X** o número de elementos *zeros* abaixo ou acima da diagonal principal. A **figura 2a** mostra a saída do programa para três matrizes triangulares inferiores e a **figura 2b** para três matrizes triangulares superiores. Desse modo, pude criar uma solução geral mostrar todas as ***n*** primeiras matrizes triangulares sem a necessidade de lançar mão de qualquer linguagem de programação, apenas usando a LÓGICA. pois, só a LOGICA tem esse poder! E, DEPOIS, só DEPOIS de ter criado lógica da solução, testei o programa **"MatrizesTriangulares"** em Visualg, cujo código-fonte é mostrado antes das figuras.

--------------------------------------------------------------------------------------------------------------------------------------

**Algoritmo** **"MatrizesTriangulares"**

//Mostra as n primeiras Matrizes Triangulares.

//Em Visualg

//Autor: Mário Leite

//------------------------------------------------------------------------------

**Const** MAXQTE=20 //limita a quantidade de matrizes

MAXELE=10 //limita as dimensões de cada matriz

**Var** MatTriang: **vetor**[1..MAXELE,1..MAXELE] **de** **inteiro**

i, j, k, m, n, p, q, EleRand, ColMat: **inteiro**

NEleD: **real**

Op, ms: **caractere**

**Inicio**

**Repita**

**Escreva**("Digite o número de matriz a serem mostradas [min 1 - max-",MAXQTE,": ")

**Leia**(n)

**Ate**((n>=1) **e** (n<=MAXQTE))

**Escreval**("")

**Repita**

**Escreva**("Matrizes triangulares do tipo Superior ou Inferior[S/I]: ")

**Leia**(Op)

Op <- Maiusc(Op)

**Ate**((Op="S") **ou** (Op="I"))

**LimpaTela**

**Se**(Op="S") **Entao**

**Escreval**("As",n," primeiras matrizes do tipo Triangular Superior:")

**Senao**

**Escreval**("As",n," primeiras matrizes do tipo Triangular Inferior:")

**FimSe**

**Escreval**("------------------------------------------------------------")

**Escreval**("")

{Gera uma matriz [**MAXELE** *x* **MAXELE**] randomicamente}

m <- 1 //contador de linhas de uma das n matrizes

**Para** i **De** 1 **Ate** n **Faca** //**i**: contador de matrizes

m <- m + 1 //inicia pela matriz 2x2

{Cria uma matriz mxm}

**Para** j **De** 1 **Ate** (i+1) **Faca**

**Para** k **De** 1 **Ate** (i+1) **Faca**

EleRand <- **Randi**(10) //considera cada elemento menor que **10**

**Enquanto** (EleRand=0) **Faca**

EleRand <- Randi(10)

**FimEnquanto**

MatTriang[j,k] <- EleRand

**FimPara**

**FimPara** //fim do *loop* de criação da matriz mxm

{Exibe a Matriz Triangular}

**Para** p **De** 1 **Ate** (i+1) **Faca**

**Para** q **De** 1 **Ate** (i+1) **Faca**

**Se**(Op="S") **Entao** //matriz triangula superior

**Se**(q<p) **Entao** //elemento abaixo da Diagonal

MatTriang[p,q] <- 0

**FimSe**

**Senao** //matriz triangula inferior

**Se**(q>p) **Entao** //elemento acima da Diagonal

MatTriang[p,q] <- 0

**FimSe**

**FimSe**

**Escreva**(MatTriang[p,q], " ")

**FimPara**

**Escreval**("")

**FimPara**

NEleD <- (m\*m-m)/2 //usa a fórmula: Elementos fora da Diagonal = **m(m-1)/2**

ms <- NumpCarac(m)

**Se**(Op="S") **Entao**

**Escreval**("Número de elementos zero da Matriz Triangular Superior ",ms,"x",ms,": ",NEleD)

**Senao**

**Escreval**("Número de elementos zero da Matriz Triangular Inferior ",ms,"x",ms,": ",NEleD)

**FimSe**

**Escreval**("")

**FimPara** //fim do *loop* contador de matrizes

**Escreval**("")

**FimAlgoritmo**





