**Lógica de Programação**

* **Procedimento (Sub) e Função (Function)**
  + Diferença entre Sub e Function
  + Passagem de parâmetros
* **Tipos de dados no Visual Basic**
  + Boolean
  + Byte
  + Date
  + Double (ponto flutuante de precisão dupla)
  + Integer
  + Long (Inteiro longo)
  + String (tamanho fixo e variável)
  + Variant
* **Declaração de Variáveis**
  + Local
  + Private
  + Global
* **Forçando declaração de variáveis**
  + Option Explicit
* **Operadores aritméticos**
  + + ..... Soma
  + - ..... Subtração
  + \* ..... Multiplicação
  + / ..... Divisão
  + ^ ..... Potência
  + \ ..... Divisão inteira
  + ( ) ... Uso de parênteses para definir qual cálculo será realizado isoladamente e de forma prioritária
* **Entrada de dados com o teclado**
  + Inpubox
* **Ferramentas de depuração**
  + Debug F8
  + Ponto de parada F9
  + Assistir variáveis
  + Verificação imediata
  + Rodar programa direto na IDE F5
* **Comentários**
  + Uso de ‘ (apóstrofe simples) como comentários da linha
* **Operadores Relacionais**
  + = ..... Igual
  + <> ..... Diferente
  + > ..... Maior que
  + < ..... Menor que
  + >= ..... Maior ou igual
  + <= ..... Menor ou igual
* **Operadores Lógicos**
  + AND
  + OR
  + NOT
* **Estrutura de Controle de Decisão**
  + IF simples
  + IF <condições> THEN <instruções>
  + IF <condições> THEN <instruções se verdadeiro>
  + IF composto

IF <condições> THEN

...

...

ELSE

...

...

ENDIF

* + IF aninhado

IF <condição1> THEN

.....

ELSEIF <condição2> THEN

.....

ENSEIF <condição...> THEN

.....

ELSE

.....

ENDIF

* + SELECT CASE

SELECT CASE <expressao de teste>

CASE <condição1>

CASE <condição2>

CASE <condiçãon>

CASE ELSE

END SELECT

* **Estruturas de Repetição**
  + Estrutura de Repetição While

While condição

[ instruções ]

Wend

* + Estrutura de Repetição Do

Do While | Until [ condição ]

[instruções]

[ Exit Do ]

[instruções]

Loop

Ou

Do

[ instruções ]

[ Exit Do ]

[ instruções ]

Loop While | Until [ condição ]

* + Estrutura de Repetição For

For contador = início To fim [ Step etapa ]

[ instruções ]

[ Exit For ]

[ instruções ]

Next [ contador ]

* **Vetor**

'Criando o vetor

Dim aNotas(1 to 4) as integer

* **Matriz**

Dim aNotas(1 To 2, 1 To 4) As Integer

* **Ordenação de Vetor/Matriz**

Usar variável de transferência entre as posições do vetor/matriz. Exemplo:

Sub ORDENANDO\_VETOR()

Dim valor(1 To 6) As Integer

Dim n As Integer

Dim i As Integer

Dim j As Integer

Dim k As Double

For n = 1 To 6

Randomize

valor(n) = Int((Rnd \* 60) + 1)

Next

Debug.Print ""

Debug.Print "Os valores não ordenados são..."

Debug.Print ""

For n = 1 To 6

Debug.Print "Posição : " & n & " do vetor é: " & valor(n)

Next

For i = 1 To 6

For j = i + 1 To 6

If valor(i) > valor(j) Then

k = valor(i) 'K é a variável de transferência

valor(i) = valor(j)

valor(j) = k

End If

Next

Next i

Debug.Print ""

Debug.Print "Os valores ordenados são..."

Debug.Print ""

For n = 1 To 6

Debug.Print "Posição : " & n & " do vetor é: " & valor(n)

Next

End Sub

* **Vetor com diferentes tipos de dados**

Exemplo:

Sub VETOR\_DIVERSOS()

Dim parametro(1 To 5) As Variant

Dim n

parametro(1) = "A1"

parametro(2) = 1

parametro(3) = True

parametro(4) = #11/12/2020#

parametro(5) = 1.75

For n = 1 To 5

Debug.Print "Tipo de dado da posição " & Str(n) & " -> " & VarType(parametro(n))

Next

End Sub

* **Vetor com quantidade de elementos de forma flexível**

Sub VETOR\_DINAMICO()

Dim vetor() As Integer

Dim n As Integer

ReDim vetor(1 To 3) ‘Cria elementos, porém não mantém o conteúdo do vetor (apaga)

vetor(1) = 1

vetor(2) = 3

vetor(3) = 2

For n = 1 To 3

Debug.Print vetor(n)

Next

ReDim Preserve vetor(1 To 5) (cria mais dois elementos e mantém o conteúdo que já existia no vetor nas posições 1 a 3).

* **Quantidade de elementos de um vetor**

UBound (arrayName, [ dimensão ])

* **Atribuindo valores usando comando Array**

Exemplo:

Sub VETOR\_VARIOS\_VALORES1()

Dim vetor

Dim n

vetor = Array(1, 5, 3)

For n = 0 To 2

Debug.Print vetor(n)

Next

End Sub

* **Encadeamento de código**

Exemplo sem encadeamento

Sub SEM\_ENCADEAMENTO()

Dim notas(1 To 4) As Byte

Dim qtde\_notas

Dim media As Byte

Dim n

Dim lnotaok As Boolean

qtde\_notas = 4

For n = 1 To 4

lnotaok = False

While lnotaok = False

InputBox (""Informe a nota "" & n)

If notas(n) < 0 Or notas(n) > 10 Then

MsgBox ""Nota inválida"", vbCritical

Else

lnotaok = True

End If

Wend

Next

For n = 1 To 4

soma = soma + notas(n)

Next

media = Int(soma / 4)

If media > 7 Then

MsgBox ""aprovado, média = "" & media

Else

MsgBox ""reprovado, média = "" & media

End If

End Sub

Exemplo com encadeamento

Sub COM\_ENCADEAMENTO()

Dim notas(1 To 4) As Byte

Dim qtde\_notas

Dim media As Byte

Dim n

Dim lnotaok As Boolean

qtde\_notas = 4

For n = 1 To 4

lnotaok = False

While lnotaok = False

InputBox (""Informe a nota "" & n)

If notas(n) < 0 Or notas(n) > 10 Then

MsgBox ""Nota inválida"", vbCritical

Else

lnotaok = True

End If

Wend

Next

For n = 1 To 4

soma = soma + notas(n)

Next

media = Int(soma / 4)

If media > 7 Then

MsgBox ""aprovado, média = "" & media

Else

MsgBox ""reprovado, média = "" & media

End If

End Sub

* **Módulo**
  + Renomeando módulo
  + Criando módulos separados (functions, procedures, variables)
  + Protegendo módulo com senha

**Eventos de Planilha (Worksheet)**

* **Eventos**

Texto

Descrição gerada automaticamente

**Eventos da Pasta de Trabalho (Workbook)**

* **Workbook**

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

**Form & Controls**

* **Form**

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

* **Controles**
  + Rótulo
  + Caixa de Texto
  + Caixa de Combinação
  + Caixa de Listagem
  + Caixa de Seleção
  + Botão de Opção
  + Botão de Ativação
  + Quadro
  + Botão de Comando
  + TabStrip
  + Multi-Página
  + Barra de Rolagem
  + Botão de Rotação
  + Imagem
  + RefEdit

**Diversos**