

AULA 01 – CONCEITO DE ALGORITMOS

A lógica e o raciocínio lógico estão presentes em nosso dia-a-dia, e eles nos ajudam a organizar nossas atitudes e ações, bem como, determina uma seqüência de pensamentos que nos permitem dialogar com pessoas e resolver problemas

O raciocínio lógico utilizado pelas pessoas pode sofrer alterações dependendo de alguns fatores como o momento vivido pela pessoa, seu estado emocional, calma, conhecimento, paciência, experiência, criatividade, responsabilidade, entre outras

Para o uso da lógica é necessário ter domínio sobre o pensamento. Neste sentido, Manzano (Manzano 1996) define raciocínio lógico como um conjunto de estudos que visa a determinar os processos intelectuais que são as condições gerais do conhecimento verdadeiro. Outros, preferem dizer que é a seqüência coerente, regular e necessária de acontecimentos, de coisas ou fatos, ou até mesmo, que é a maneira do raciocínio particular que cabe a um indivíduo ou a um grupo. Além disso, pode ser um esquema sistemático que define as interações de sinais no equipamento automático do processamento de dados, ou um computador científico com o critério e princípios formais de raciocínio e pensamento.

Assim, podemos dizer que a lógica é a ciência que estuda as leis e critérios de validade que regem o pensamento e a demonstração, ou seja, ciência dos princípios formais do raciocínio.

"LÓGICA: coerência de raciocínio, de idéia; seqüência coerente, regular e necessária de acontecimentos, de coisas" (AURÉLIO).

"Podemos relacionar a LÓGICA como a 'correção do pensamento', pois uma de suas preocupações é determinar quais operações são válidas e quais não são, fazendo análise das formas e leis do pensamento: " (FORBELLONE, 2005).

Segundo FORBELLONE (2005), a lógica estuda e ensina a colocar "ordem no pensamento", "arte de bem pensar", "ciência das formas de pensamento", e visto que a forma mais complexa de pensamento é o raciocínio a lógica estuda a "correção do raciocínio" ou "ordem da razão".

Usar a lógica é um fator a ser considerado pelos profissionais de informática (programadores, analistas, etc), pois o objetivo destes profissionais é a solução de problemas e atingir objetivos apresentados por seus usuários com eficiência e eficácia.

Este conhecimento permite a busca de soluções diárias, seja na área administrativa, comercial, financeira, planejamento e requer a definição de estratégias que construam o conhecimento para o processo de automação de atividades nas empresas.

Alguns conceitos importantes:

- Lógica de programação é a técnica de encadear pensamentos para atingir determinado objetivo.
- A lógica de programação é necessária para pessoas que desejam trabalhar com desenvolvimento de sistemas e programas, ela permite definir a seqüência lógica para o desenvolvimento
- Seqüência Lógica são passos executados até atingir um objetivo ou solução de um problema. Estes pensamentos, podem ser descritos como uma seqüência de instruções, que devem ser seguidas para se cumprir uma determinada tarefa.
- Na linguagem comum, entende-se por instruções "um conjunto de regras ou normas definidas para a realização ou emprego de algo". Em informática, porém, instrução é a informação que indica a um computador uma ação elementar a executar. Convém ressaltar que uma ordem isolada não permite realizar o processo completo, para isso é necessário um conjunto de instruções colocadas em ordem seqüencial lógica. Por exemplo, se quisermos fazer uma omelete de batatas, precisaremos colocar em prática uma série de instruções: descascar as batatas, bater os ovos, fritar as batatas, etc... É evidente que essas instruções tem que ser executadas em uma ordem adequada – não se pode descascar as batatas depois de fritá-las. Dessa maneira, uma instrução tomada em separado não tem muito sentido; para obtermos o resultado, precisamos colocar em prática o conjunto de todas as instruções, na ordem correta.
- Instruções são um conjunto de regras ou normas definidas para a realização ou emprego de algo. Em informática, é o que indica a um computador uma ação elementar a executar.

No exemplo abaixo estão descritas ações de quem vai a um banco; entretanto, veja se seria possível retirar o dinheiro obedecendo à seqüência descrita.

- 1 Sair do banco.
- 2 Aguardar a vez de ser atendido.
- 3 Entrar na fila do caixa.
- 4 Digitar a senha.
- 5 Entrar no banco
- 6 Informar o valor a ser retirado.
- 7 Procurar a fila do caixa.
- 8 Pegar o comprovante de saque, o cartão eletrônico e o dinheiro.
- 9 Entregar o cartão eletrônico ao funcionário que atende no caixa.
- 10 Conferir o valor em dinheiro.

Você conseguiria retirar o dinheiro? Com certeza não, pois, embora as ações estejam corretas, a seqüência está errada. A primeira ação é sair do banco. Como se pode retirar dinheiro de um banco estando fora dele? Agora observe as mesmas ações na seqüência correta.

- 1 Entrar no banco.
- 2 Procurar a fila do caixa.
- 3 Entrar na fila do caixa.
- 4 Aguardar a vez de ser atendido.
- 5 Entregar o cartão eletrônico ao funcionário que atende no caixa.
- 6 Informar o valor a ser retirado.
- 7 Digitar a senha.
- 8 Pegar o comprovante de saque, o cartão eletrônico e o dinheiro.
- 9 Conferir o valor em dinheiro.
- 10 Sair do banco

Neste caso, utiliza-se a lógica para ordenar e corrigir pensamentos ou ações voltados para a solução de problemas, e falar em **Lógica de Programação** significa apenas contextualizar a lógica na programação de computadores, buscando a melhor seqüência de ações para solucionar determinado problema.

Na programação de computadores, o nome dessa seqüência de ações é Algoritmo, que é, de maneira bem simples, a seqüência de passos ou ações para atingir um objetivo. Já a ação é um fato que, a partir de um estado inicial, após um período de tempo finito, produz um estado final previsível e definido. Por exemplo, abrir uma porta é uma ação, escrever uma palavra, pintar uma parede, etc..

Algoritmo não é a solução de um problema, pois, se assim fosse, cada problema teria um único algoritmo. Algoritmo é um caminho para a solução de um problema, e em geral, os caminhos que levam a uma solução são muitas. Neste caso, observa-se que o desenvolvimento da solução de um problema, inicialmente, temos que desenvolver um algoritmo e depois realizar a implementação.

O processo de implementar a solução de um problema sem desenvolver um algoritmo que o descreva é uma fase muito complicada e difícil.

Assim:

“Um algoritmo pode ser definido como uma seqüência de passos que visam atingir um objetivo bem

definido” (FORBELLONE 2005).

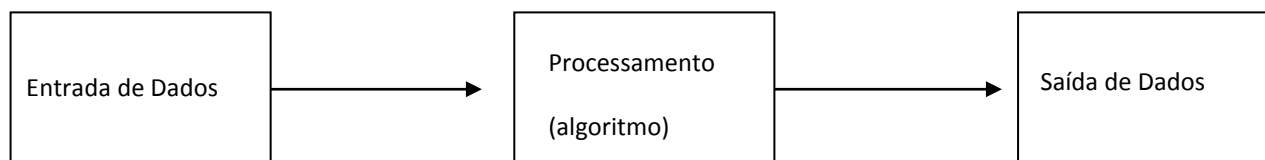
“Algoritmo é a descrição de um conjunto de comandos que, obedecidos, resultam numa sucessão finita de ações” (FARRER 1989)

Tipos de dados

Todo trabalho realizado pelo computador manipulam dados que podem ser:

- a) **Instruções** que comandam o funcionamento da máquina e determinam a maneira como devem ser tratados os dados.
- b) **Dados propriamente ditos**, que correspondem à porção das informações a serem processadas pelo computador.

Devemos ter em mente o diagrama abaixo, pois é a partir dele que devemos solucionar todos os problemas baseados em computador e alcançar os objetivos do curso:



Desta forma, temos por objetivo resolver problemas do mundo real que podem ser descritos como um conjunto de dados(premissas) e desenvolver algoritmos que tenham como resultado um outro conjunto(conclusão) de dados.

O que é um dado?

Dado pode ser visto como qualquer coisa que exista no mundo real, como por exemplo, o nome de uma pessoa, seu CPF, ou informações que representem um elemento (como os lados de um triângulo), código de um cliente, etc.

Assim, saber os dados de um problema e poder identificá-los já é um bom começo para se desenvolver um bom algoritmo. Além disso, devemos identificar quais são os dados de **entrada** do problema e quais serão os **resultados esperados**, designados na forma de dados de **saída**.

Podemos identificar a Lógica de Programação como sendo a identificação de um conjunto de passos que transformam dados de entrada de um certo problema em geração de resultados através do desenvolvimento de dados de saída.

Em alguns problemas a saída é representada por dados na forma de mensagem.

O VBA trabalha com vários tipos de dados, como números, datas, textos e objetos. Boas práticas de programação sugerem que além de que você declarar todas as variáveis que utiliza, também especifique seu tipo de dados num programa. Na verdade, isso não é somente considerada uma boa prática, mas quase uma obrigatoriedade.

Para melhor entendimento, podemos dizer que o VBE trabalha com dois tipos de dados: os numéricos e os não numéricos.

A tabela abaixo mostra os tipos de dados numéricos:

Tipos numéricos em VBA		
Tipo de dado	Alocação	Intervalo permitido
Byte	1 byte	0 a 255
Integer	2 bytes	-32.768 a 32.767
Long	4 bytes	-2.147.483.648 a 2.147.483.647
Single	4 bytes	+/-3,402823E+28
Double	8 bytes	+/-1,79769313486232E+308
Currency	8 bytes	+/-922.337.203.685.477,5808
Decimal	12 bytes	+/-79.228.162.514.264.337.593.543.950.335

A tabela abaixo mostra os tipos de dados não numéricos que o VBA suporta:

Tipos não-numéricos em VBA		
Tipo de dado	Alocação	Intervalo permitido
String	1 byte por caractere	1 a 65.400 caracteres
String (tamanho variável)	1 byte por caractere + 10 bytes	0 a 2.000.000.000 de caracteres
Date	8 bytes	-657434,9... (01/01/100) a 2958465,9... (31/12/9999)
Boolean	2 bytes	True ou False
Object	4 bytes	Qualquer objeto
Variant (numérica)	16 bytes	Qualquer valor tão grande quanto um Double
Variant (texto)	Length+22 bytes	O mesmo que um Strig de tamanho variável

Estrutura de um programa em VBA

Como foi falado anteriormente uma macro é um programa escrito em VBA que realiza alguma ação bem definida. Neste sentido, uma macro é visto como um procedimento ou subrotina no VBA e desta forma possui a seguinte estrutura:

Sub <nome>(<parâmetros>)

End Sub

Assim:

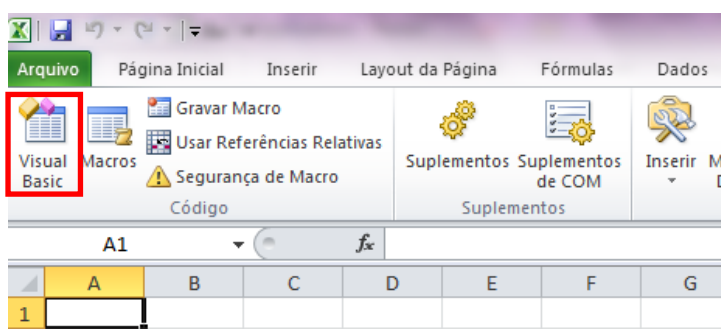
<nome> - refere-se ao nome da macro ou do programa a ser executado

<parâmetros> são canais de comunicação entre um programa chamador e subrotina(será visto posteriormente)

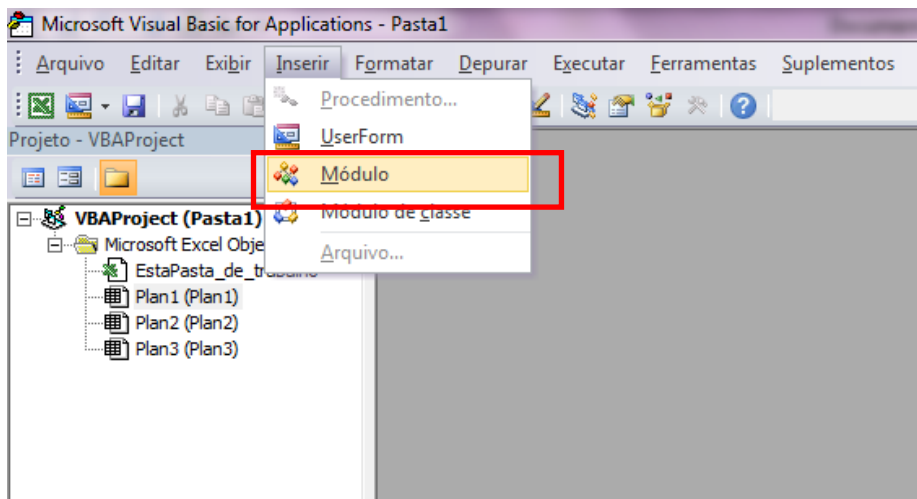
Para que possamos criar um programa devemos inicialmente acessar o ambiente VBA e abrir um Módulo, que é o local onde todas as macros(programas) estarão armazenados.

Passo 1: Abra o Excel e selecione a barra desenvolvedor.

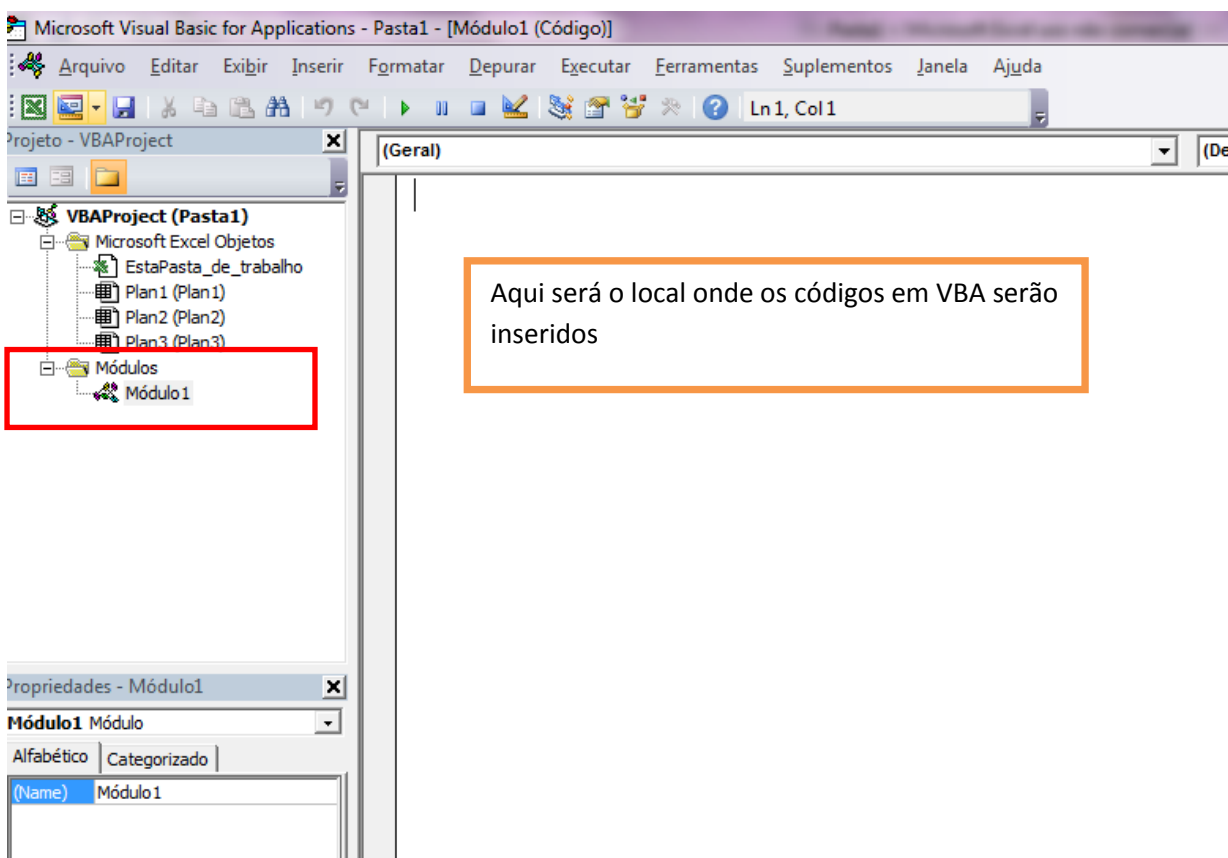
Passo 2: Na Barra desenvolvedor, selecione a opção Visual Basic



Passo 3: Abrir o ambiente do VBA, escolha a opção Inserir->módulo



Passo 4: ao clicar em módulo, o ambiente VBA abrirá uma janela do lado direito, que será o local onde os códigos serão inseridos:



Conceito de Variáveis

Uma **variável** se refere a uma entidade destinada a armazenar uma informação na memória do computador. Este processo é importante porque em todos os problemas temos dados de entrada e saída. A questão é: Onde e como vamos armazenar estes dados?

A resposta para esta questão é o conceito de **variável** que foi criado para manter estes dados armazenados dentro do computador de tal forma que, possam ser usadas para fazer operações, guardar informações de entrada e de saída, resolver expressões matemáticas, etc.

Existem **três atributos** básicos para a variável (SALIBA 1993):

- Nome:** tem por função diferenciá-la das demais. Deve necessariamente começar por letra; não pode conter símbolos especiais exceto a sublinha(_). É permitido dígitos e letras. Obviamente, é interessante adotar

nomes de variáveis relacionadas às funções exercidas pelas mesmas dentro de um programa. O nome de variável é considerado um **identificador**.

- b) **Tipo de dado:** Se refere a natureza de dados por ela armazenados (tipos básicos: numérico, lógico e caractere). É aqui que devemos designar o tipo de dado que deverá ser usado por cada variável.
- c) **Informação:** é o dado propriamente dito, dependendo do tipo definido no item acima. Assim, a informação pode ser o número 5, ou o caractere 'a', etc.

Todas as variáveis em algoritmos devem ser definidas antes de serem utilizadas. Isto é necessário para que o **compilador** (tradutor de código fonte em linguagem entendida por máquina) reserve um espaço na memória para as mesmas. O processo de definição das variáveis a serem utilizadas pelo algoritmo é chamada de declaração de variáveis.

Todas as variáveis deverão ser declaradas em VBA, por meio do seguinte comando:

`DIM < NOME_DA_VARIAVEL> as < TIPO_DA_VARIÁVEL>;`

- ✓ **DIM** - representa a declaração de variáveis
- ✓ **<NOME_DA_VARIÁVEL>** - representa a lista contendo o nome de uma ou mais variáveis. Se forem representadas vários nomes, cada um deles é separado por vírgula e deve acompanhar de seus respectivos tipos de dados.
- ✓ **<TIPO_DAS_VARIÁVEIS>** - representa o tipo de dado da variável. Por exemplo, um integer, double, string, etc

Exemplos:

```
Dim x1 as Integer
```

```
Dim casa as String, nro as Double
```

Observação:

Algumas observações importantes devem ser feitas em relação ao nome do programa e nomes de variáveis. Tanto os nomes dos programas quanto de variáveis devem seguir a seguinte regra de nomeação:

- a) Não pode começar com NÚMERO

Exemplo: 1prog, 10xpe, 2123cal, 1aula

- b) Não pode ter espaços em branco

Exemplo: aula 1, prog comp

- c) Não pode ter acentos, cedilha, ou qualquer pontuação

Exemplo: aulãol, cçwqr

- d) PODE misturar letras e números

Exemplo: progl pr23rtu

- e) No VBA, não é obrigatório a declaração de variáveis. Porém é recomendável que declaremos todas as variáveis, de tal forma que o código fique mais claro e de fácil compreensão.

Comando de Entrada de Dados.

Em um sistema computacional é necessário obter dados para realizar o processamento. Desta forma, os dispositivos de entrada de dados permitidos são: teclado, mouse, scanner, leitora de disco, etc.

Nesta primeira aula vamos utilizar como entrada de dados uma posição na planilha que seja apontada por uma referência relativa.

Vamos utilizar o comando `ACTIVECELL`

Comando de Saída de Dados.

Os comandos de saída de dados têm como objetivo, mostrar mensagens ou o conteúdo das variáveis para o monitor de vídeo (tela), impressora ou mesmo um arquivo.

Nesta primeira aula vamos utilizar como entrada de dados uma posição na planilha que seja apontada por uma referência relativa.

Vamos utilizar o comando `ACTIVECELL`

Exemplo de um programa em VBA

Assim considere o seguinte exemplo: *Suponha sejam dados 02 notas e se deseja calcular a soma destas notas, a sua média aritmética e a sua média ponderada com pesos 3 e 7.*

Para a solução deste pequeno problema é necessário inicialmente identificar os dados de entrada:

Dados de Entrada (as duas notas do aluno):

n1 que é do tipo numérico real

n2 que é do tipo numérico real

Dados de Saída(Resultados): (o que é pedido ou solicitado):

- Soma das notas
- O Cálculo da média aritmética
- O Cálculo da média ponderada

Processamento: As operações que foram solicitadas:

$Soma = n1 + n2$

$Media_a = (n1 + n2) / 2$

$Media_p = n1 * 0,3 + n2 * 0,7$

Passos a serem seguidos para o desenvolvimento da macro:

Considere a planilha como mostrada abaixo:

	A	B	C	D	E
1	NOTA1	NOTA2	SOMA	MEDIA ARITMETICA	MEDIA PONDERADA
2	9,4	5,7			
3	7,8	8,3			
4	6,9	9,5			
5	8,5	8,2			
6	7,8	9,1			
7	9,3	8,5			
8					

Diagram illustrating the VBA macro development steps for calculating the sum, arithmetic mean, and weighted mean of two notes (NOTA1 and NOTA2) using the `ACTIVECELL` command.

Red arrows indicate the relative cell references used in the macro code:

- `Activecell.offset(0,-1).value` points to the cell above the current cell (e.g., A2 from A3).
- `Activecell.offset(0,-2).value` points to the cell two rows above the current cell (e.g., A4 from A6).
- `Activecell.value` points to the current cell (e.g., A6).
- `Activecell.offset(0,1).value` points to the cell below the current cell (e.g., A7 from A6).
- `Activecell.offset(0,2).value` points to the cell two rows below the current cell (e.g., A8 from A6).

Observe que as células em verde representam os dados de entrada, ou seja, o que é dado pelo problema proposto e as células em branco representam os dados que são solicitados pelo algoritmo. Neste caso, devemos resolver o problema identificando esses elementos e preenchendo o programa com o código adequado.

Para a manipulação de dados utilizando referencia relativa, vamos utilizar o comando Activecell que representa a “célula ativa”. Assim:

- a) `Activecell.select` – seleciona uma célula como célula ativa
- b) `Activecell.value` – obtém-se o valor da célula selecionada.
- c) `Activecell.offset(linha,coluna).Value` – obtém-se o valor da célula ativa que é deslocada por *linha* e *coluna*.
- d) `Activecell.offset(0,1).Value = 20` – a partir da célula ativa avance uma coluna à direita e insira o numero 20.
- e) `Activecell.offset(linha,coluna).Select` –seleciona uma célula como ativa que esteja deslocada por *linha* e *coluna*.
- f) `Activecell.offset(1,0).Select` – a partir da célula ativa avance uma linha para baixo e selecione a célula.
- g) `ActiveCell.Offset(0, 1).Select` a partir da célula ativa caminha uma coluna à direita
- h) `Selection.FormulaR1C1 = "10"` Insere o numero 10 na célula ativa
- i) `ActiveCell.Value= 100` - Na célula ativa coloque o numero 100

Para o exemplo dado acima teremos:

Para a declaração das variáveis de entrada de dados::

```
Dim n1 As Double, n2 As Double
```

Para representar as variaveis de saida de dados:

```
Dim soma As Double, media_a As Double, media_p As Double
```

Para representar a leitura de dados, vamos obter os dados da planilha que estejam na coluna A para n1 e os dados da coluna B para n2 da seguinte forma.

```
n1 = ActiveCell.Offset(0, -2).Value  
n2 = ActiveCell.Offset(0, -1).Value
```

Observe que este comando é executado tendo como celula ativa os elementos da coluna C.

As operações a serem realizadas podem ser mostradas abaixo:

```
soma = n1 + n2  
media_a = (n1 + n2) / 2  
media_p = (n1 * 0.3 + n2 * 0.7)
```

Para mostrar os resultados da operação, vamos utilizar as colunas C, D e E da seguinte forma:


```
ActiveCell.Value = soma
ActiveCell.Offset(0, 1).Value = media_a
ActiveCell.Offset(0, 2).Value = media_p
```

Assim, o programa completo em VBA:

```
Sub aula01()
'declaração de variaveis
Dim n1 As Double, n2 As Double

Dim soma As Double, media_a As Double, media_p As Double

'leitura de dados
n1 = ActiveCell.Offset(0, -2).Value
n2 = ActiveCell.Offset(0, -1).Value

'operações a serem realizadas
soma = n1 + n2
media_a = (n1 + n2) / 2
media_p = (n1 * 0.3 + n2 * 0.7)

'dados a serem inseridos na planilha: resultado esperado
ActiveCell.Value = soma
ActiveCell.Offset(0, 1).Value = media_a
ActiveCell.Offset(0, 2).Value = media_p

End Sub
```

Se não utilizarmos as variáveis de saída, podemos mostrar os dados diretamente, mas não é considerada boa prática de programação:

```
Sub aula02()
'declaração de variaveis
Dim n1 As Double, n2 As Double

'leitura de dados
n1 = ActiveCell.Offset(0, -2).Value
n2 = ActiveCell.Offset(0, -1).Value

'dados a serem inseridos na planilha: resultado esperado
ActiveCell.Value = n1 + n2
ActiveCell.Offset(0, 1).Value = (n1 + n2) / 2
ActiveCell.Offset(0, 2).Value = n1 * 0.3 + n2 * 0.7

End Sub
```