

## LIÇÃO 1 – CRIANDO FUNÇÕES DEFINIDAS PELO USUÁRIO

No VBA podemos criar procedimentos do tipo **sub** ou **function**. Basicamente a diferença entre estes dois tipos de procedimentos é que a função (function) retorna um valor. O VBA, como todas as linguagens de programação, oferece uma grande variedade de funções internas que auxiliam na codificação de procedimentos. São definidas a partir de sua sintaxe escrita no formato: **Nome (argumento1, argumento2,...)**. A função pode não ter argumento, os argumentos podem ser obrigatórios ou opcionais. Apesar do Excel/VBA já possuir uma série de funções pré-definidas, customizar funções poderá simplificar fórmulas e eliminar o aparecimento de código duplicados o que diminui a possibilidade de erro. As funções criadas pelo usuário podem ser utilizadas nos procedimentos ou diretamente em fórmulas inseridas na planilha.

**[Public | Private] [Static] Function Nome (argumento1 As tipo, [Optional] argumento2 As tipo...) As TipoRetorno**

**[instruções]**

**[nome = expressão]**

**[Exit Function]**

**[instruções]**

**[nome = expressão]**

**End Function**

- **Public:** função poderá ser utilizada em qualquer procedimento/módulo do projeto. Representa o escopo padrão caso não seja informado.
- **Private:** função acessível apenas para os procedimentos do mesmo módulo.
- **Static:** indica que os valores das variáveis declaradas no procedimento reterão seu valor após sua execução.
- **Nome:** nome da função (segue as mesmas regras para os nomes de variáveis).
- **Argumento:** representam os "valores" que serão repassados para a função.
- **Tipo:** especifica o tipo do argumento.
- **Optional:** define se o argumento será opcional ou não.
- **TipoRetorno:** define o tipo de dados que será retornado pela função.
- **Exit Function:** instrução que finaliza a função antes dela ser completada.

**Fonte: Ajuda do aplicativo**

### • Funções com argumentos obrigatórios

Função **ParOuImpar**: esta função deverá receber **um número inteiro** e retornar as strings **"par"** ou **"ímpar"**.

1. Abra o arquivo **VBA3** e habilite seu conteúdo.
2. Pressione **ALT+F11** para abrir o editor, expanda **Módulos** e dê um clique duplo sobre **Funcoes** para abrir a janela de código deste módulo.
3. Posicione o cursor e igite neste módulo o código da função **ParOuImpar** listado abaixo.

**'Função ParOuImpar recebe um número inteiro e retorna uma string**

**Function ParOuImpar(num As Integer) As String**

**'Se o resto da divisão de num por 2 for igual a 0**

**If num Mod 2 = 0 Then**

**'Função ParOuImpar retorna Par**

**ParOuImpar = "Par"**

**'Senão (resto da divisão de num por 2 diferente de 0)**

**Else**

**'Função ParOuImpar retorna Ímpar**

**ParOuImpar = "Ímpar"**

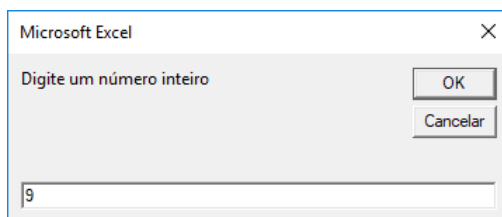
**End If**

**End Function**

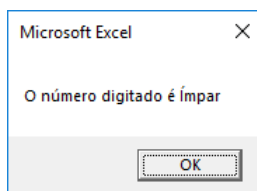
4. Complete o código do procedimento **Exemplo1** como listado abaixo.

```
Sub Exemplo1()
'Declarar a variável numero do tipo inteiro
Dim numero As Integer
'Atribuir a número o retorno da InputBox
numero = InputBox("Digite um número inteiro")
'Exibir msg com o valor retornado da função ParOuImpar
MsgBox "O número digitado é " & ParOuImpar(numero)
End Sub
```

5. Posicione o cursor no procedimento e pressione **F5** para executar o código digitado.  
6. Digite **um número** e clique em **OK**.



7. Clique em **OK**.



Função **Massa**: esta função deverá receber **dois números reais (peso e altura)** e retornar outro ( $\text{Massa} = \text{peso} / \text{altura}^2$ ).

1. Posicione o cursor e digite neste módulo o código da função **Acima** listado abaixo.

```
'Função Massa recebe dois números reais e retorna outro
Function Massa (peso As Single, altura As Single) As Single
'Função Massa retorna o resultado da expressão peso/altura^2
Massa = peso / altura ^ 2
End Function
```

2. Volte para o **Excel** e selecione a célula **D4** da planilha **IMC**.  
3. Digite nesta célula a fórmula **=Massa(B4;C4)**.

SOMA			
=Massa(B4;C4)			
	A	B	C
1	Tabela de cálculo do Índice de Massa Corporal		
2			
3	Nome	Peso (Kg)	Altura (m)
4	Maria	48	1,63
5	João	97,5	1,72

4. Tecle **ENTER**.

	A	B	C	D
1	Tabela de cálculo do Índice de Massa Corporal			
2				
3	Nome	Peso (Kg)	Altura (m)	IMC
4	Maria	48	1,63	18,06616783
5	João	97,5	1,72	

5. Posicione o mouse na alça de preenchimento da célula **D4** e dê um clique duplo para copiar a fórmula para as demais células da coluna.

	A	B	C	D
1	Tabela de cálculo do Índice de Massa Corporal			
2				
3	Nome	Peso (Kg)	Altura (m)	IMC
4	Maria	48	1,63	18,06616783
5	João	97,5	1,72	

	A	B	C	D
1	Tabela de cálculo do Índice de Massa Corporal			
2				
3	Nome	Peso (Kg)	Altura (m)	IMC
4	Maria	48	1,63	18,06616783
5	João	97,5	1,72	32,95700455
6	Fernando	117	1,8	36,1111145
7	Luisa	86,5	1,74	28,57048416
8	Ana	65	1,85	18,99196434
9	Francisco	148	1,87	42,32320023

6. Volte para o editor.

### • Funções com argumentos opcionais

Função **Semanas**: esta função deverá receber **uma data** (final do período) e retornar a **quantidade de semanas** que existe entre esta **data** e a **data de hoje**. Caso a **data** não seja informada, o dia **31/12/2018** será utilizado como **data final**.

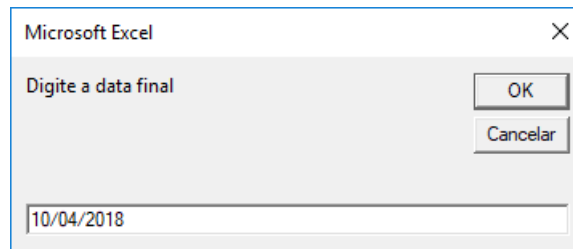
1. Digite neste módulo o código da função **Semanas** listado abaixo.

```
'Função Semanas recebe um argumento opcional do tipo variant
Function Semanas(Optional Final)
'Se Final não informado
If IsMissing(Final) Then
'Função Semanas retorna a quantidade de semanas entre hoje e 31/12/2018
    Semanas = DateDiff("ww", Date, #12/31/2018#)
'Senão (Final informado)
Else
'Função Semanas retorna a quantidade de semanas entre hoje e final
    Semanas = DateDiff("ww", Date, Final)
End If
End Function
```

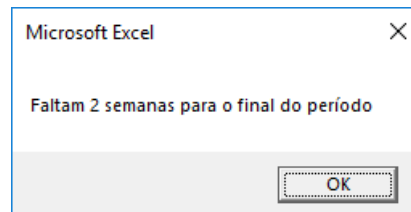
2. Complete o código do procedimento **Exemplo2** listado abaixo.

```
Sub Exemplo2()
'Declarar variável fim do tipo variant
Dim fim
'Atribuir a variável fim o retorno da InputBox
fim = InputBox("Digite a data final")
'Caso fim não seja informado
'Enviar msg com o valor retornado pela função Semanas
'sem informar nenhum argumento
If fim = Empty Then
    MsgBox "Faltam " & Semanas() & " semanas para o final do ano"
'Senão enviar msg com o valor retornado pela função Semanas
'informando um argumento
Else
    MsgBox "Faltam " & Semanas(fim) & " semanas para o final do semestre"
End If
End Sub
```

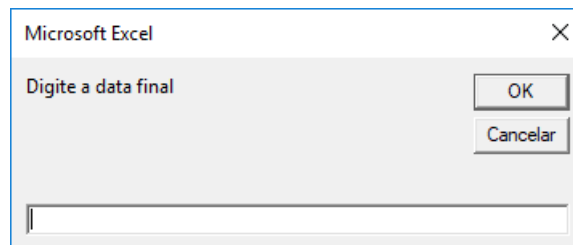
3. Pressione **F5** para executar o código digitado, digite **10/04/2018** e clique em **OK**.



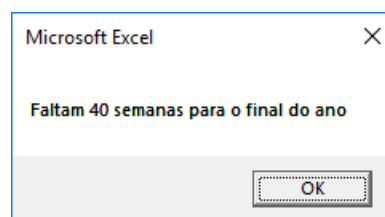
4. Clique em **OK**.



5. Execute novamente e clique em **OK** sem informar a data final.



6. Clique em **OK**.



Função **Area**: esta função deverá receber **dois números reais** (**comprimento** e a **largura**) e retornar outro **número real** (**área**). Caso a **largura** não seja fornecida, a função deverá utilizar apenas o **comprimento** para calcular a **área**. Neste caso o argumento opcional será testado por um valor padrão.

1. Posicione o cursor e digite neste módulo o código da função **Area** listado abaixo.

```
'Função Area recebe dois número sendo o segundo opcional com padrão e retorna outro
Function Area(C As Single, Optional L As Single = 0)
'Se L não informado o valor padrão será utilizado
If L = 0 Then
'Função Area retorna C²
    Area = C ^ 2
'Senão L informado
Else
'Função Area retorna C * L
    Area = C * L
End If
End Function
```

- Volte para o **Excel** e selecione a célula **C2** da planilha **Medidas**.
- Digite a fórmula **=Area(A2;B2)** e tecla **ENTER**.

	A	B	C
1	Comprimento	Largura	Duas medidas
2	3	4	=Area(A2;B2)
3			

- Selecione a célula **D2** da planilha **Medidas**.
- Digite a fórmula **=Area(A2)** e tecla **ENTER**.

	A	B	C	D
1	Comprimento	Largura	Duas medidas	Uma medida
2	3	4	12	=Area(A2)
3				

	A	B	C	D
1	Comprimento	Largura	Duas medidas	Uma medida
2	3	4	12	9
3				

- Volte para o editor.

### • Funções sem argumentos

Função **Situacao**: esta função deverá retornar se **hoje** é um dia de **trabalho** (segunda a sexta) ou de **descanso** (sábado ou domingo).

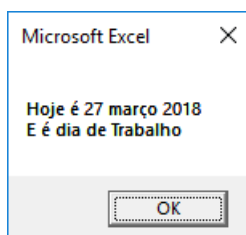
- Digite neste módulo o código da função **Situacao** listado abaixo.

```
'Função Situacao
Function Situacao()
'Se dia da semana diferente de domingo e sábado
If Weekday(Date) <> 1 And Weekday(Date) <> 7 Then
'Função Situação retorna Trabalho
    Situacao = "Trabalho"
'Senão (dia da semana igual à domingo ou sábado)
'Função Situação retorna Descanso
Else
    Situacao = "Descanso"
End If
End Function
```

- Complete o código do procedimento **Exemplo3** listado abaixo.

```
Sub Exemplo3()
'Exibir uma msg com a data atual formatada
'E na outra linha o valor retornado pela função Situacao
MsgBox "Hoje é " & Format(Date, "d mmmm yyyy") _
    & Chr(13) & "E é dia de " & Situacao()
End Sub
```

- Posicione o cursor no procedimento e pressione **F5** para executar o código digitado.
- Clique em **OK**.



**Função Dias:** esta função retornará quantos dias faltam para o final do ano.

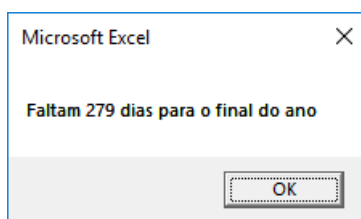
1. Digite neste módulo o código da função **Dias** listado abaixo.

```
'Função Dias
Function Dias()
'Função Dias retorna a quantidade de dias entre hoje e 31/12/2018
Dias = DateDiff("d", Date, #12/31/2018#)
End Function
```

2. Complete o código do procedimento **Exemplo4** listado abaixo.

```
Sub Exemplo4()
'Exibe quantos dias faltam para o final do ano utilizando
'o valor retornado pela função Dias
MsgBox "Faltam " & Dias & " dias para o final do ano"
End Sub
```

3. Execute este procedimento e clique em **OK**.



4. Feche o editor, salve as alterações e saia do **Excel**.

## FUNÇÕES INTERNAS (FONTE: AJUDA DO APLICATIVO)

**Date:** retorna a data do sistema.

Sintaxe: **Date**

**Weekday:** retorna um número inteiro que represente o dia da semana.

Sintaxe: **Weekday(data[,dia])**

- **data:** argumento obrigatório identificado por uma combinação que possa representar uma data.
- **dia:** argumento opcional que especifica o primeiro dia da semana. Se não for especificado será considerado **vbSunday** (1).

CONFIGURAÇÕES PARA O ARGUMENTO DIA		
Constante	Valor	Descrição
vbSunday	1	Domingo (padrão)
vbMonday	2	Segunda-feira
vbTuesday	3	Terça-feira
vbWednesday	4	Quarta-feira
vbThursday	5	Quinta-feira
vbFriday	6	Sexta-feira
vbSaturday	7	Sábado

**Chr:** retorna o caractere associado ao código de caractere especificado.

Sintaxe: **Chr(codigo)**

- **codigo:** argumento obrigatório que identifica o caractere.

**Format:** retorna a string formatada segundo o formato informado.

Sintaxe: **Format(Expressão, Formato)**

**DateDiff:** retorna o intervalo de tempo entre duas datas especificadas.

Sintaxe: **DateDiff(Intervalo, Data1, Data2)**

- **Intervalo:** argumento obrigatório que representa o intervalo de tempo utilizado para calcular a diferença entre as datas.
- **Data1 e Data2:** argumento obrigatório que representam as datas que serão utilizadas no cálculo.

INTERVALO	
Definição	Descrição
yyy	Ano
q	Trimestre
m	Mês
y	Dia do ano
d	Dia
w	Dia da semana
ww	Semana
h	Hora
n	Minuto
s	Segundo

**Obs.:** Caso data1 se refira a um ponto mais recente no tempo a data2, a função DateDiff retornará um número negativo. Ao comparar 31 de dezembro com 1º de janeiro do ano imediatamente seguinte, **DateDiff** para Ano ("yyyy") retornará 1, mesmo que tenha se passado apenas um dia.