

Capítulo 2

Trabalhar com macros, funções lógicas (Exemplos: Triângulo de Pascal, Domingo dê Páscoa)

Cópias relativas e absolutas

Nesta parte mostraremos outra vez a importância que tem o processo de **copiar** células no trabalho com o programa Excel. Nos lembramos: quando copiamos células, o resultado é muito diferente se copiamos dados do tipo texto e números, ou se copiamos fórmulas e funções. No primeiro caso, reproduzimos exatamente o conteúdo das células. No segundo caso, temos de ter mais cuidado, já que o Excel altera o endereço das células que intervêm nas fórmulas.

Exemplificando: se a célula A3 contiver a seguinte fórmula: $=A1+A2$, e a copiarmos para a célula B3, as referências da fórmula em B3 se ajustarão, ficando assim: $=B1+B2$. Isso acontece devido à fórmula original conter apenas *referências relativas* (sem o símbolo \$).

O caractere "\$" diferencia uma referência relativa de uma absoluta. Para que a referência não se altere quando é copiada, devemos incluir o "\$" ao endereço. No exemplo acima, a fórmula na célula A3 ficaria assim: $=\$A1+\$A2$. Mas copiando-a para A4 daria $=\$A2+\$A3$, ou seja, a proteção funciona só horizontalmente. O uso de um segundo "\$" protege a fórmula para qualquer movimento. Assim, podemos copiar a equação $=\$A\$1+\$A\2 para qualquer posição da planilha que ela não se alterará. É uma *referência absoluta*.

	A	B	C	D	E	F	G	H
1	1							
2	1	1						
3	1	2	1					
4	1	3	3	1				
5	1	4	6	4	1			
6	1	5	10	10	5	1		
7	1	6	15	20	15	6	1	
8								
9								
10								
11								
12	Triângulo de Tartaglia-Pascal							
13								
14								

O **Triângulo de Pascal** nos dá uma excelente oportunidade de aprender **copiar** fórmulas de uma célula a outras. A idéia é de produzir numa planilha Excel quase automaticamente este famoso triângulo formado por números inteiros. Vamos considerar *dois tipos de triângulo*.

Na figura acima, colocamos 1 nas células A1:A7, também um 1 em B2. Na célula B3 colocamos a fórmula =A2+B2 que copiamos até G3 e depois até G7. O triângulo dos zeros limpamos "manualmente", mais tarde vamos ver, como se podem ocultar os zeros com um simples comando.

(Para copiar manualmente uma fórmula em uma célula, colocamos o cursor sobre o canto inferior direito dela, e neste momento, o cursor muda sua forma de cruz branca para uma cruz preta. Mantendo-se pressionado o botão esquerdo do mouse, arrasta-se o cursor até cobrir todas as células para as quais se deseja copiar o conteúdo da célula de origem.)

O triângulo é formado pelos números binomiais:

$$\binom{0}{0} = 1$$

$$\binom{1}{0} = 1; \binom{1}{1} = 1$$

$$\binom{2}{0} = 1; \binom{2}{1} = 2; \binom{2}{2} = 1$$

$$\binom{3}{0} = 1; \binom{3}{1} = 3; \binom{3}{2} = 3; \binom{3}{3} = 1$$

.....

As propriedades desse triângulo, embora já fossem conhecidas desde o século XII ou XIII, foram sistematizadas somente no século XVII, por Blaise Pascal.

A soma dos números em células diagonais, como nas células coloridas, são números de Fibonacci: $1 + 4 + 3 = 8$; $1 + 5 + 6 + 1 = 13$...

(Os números de Fibonacci são 1, 1, 2, 3, 5, 8, 13, 21, 34, ... veja detalhes abaixo e na seção seguinte.)

A soma dos números binomiais de uma mesma linha é uma potência de base 2: $2^0, 2^1, 2^2, 2^3$...

Na seguinte figura, vemos **outra forma** do triângulo, que surgiu na China já em 1300 AC. Aqui colocamos um 1 na célula K1 e também um 1 em J2 e L2. K3 contém a soma =J2+L2. Copie o conteúdo da célula K3 (Ctrl-C e Ctrl-V) para a célula E3. Copie seguidamente, com a aba de preenchimento, o conteúdo de E3 para todas as células entre E3 e Q3. Agora é só copiar tudo isso até Q7.

E	F	G	H	I	J	K	L	M	N	O	P	Q
						1						
					1		1					
0	0	0	0	1	0	2	0	1	0	0	0	0
0	0	0	1	0	3	0	3	0	1	0	0	0
0	0	1	0	4	0	6	0	4	0	1	0	0
0	1	0	5	0	10	0	10	0	5	0	1	0
1	0	6	0	15	0	20	0	15	0	6	0	1

Finalmente podemos ocultar todos os zeros. Selecione todas as células de E1 até Q7 com Shift-<para baixo>. No menu *Formatar*, clique em *Células* e, em seguida, clique na guia *Número*. Na lista *Categoria*, clique em *Personalizado* e na caixa **Tipo**, digite **0;-0;;@**

Macros

Agora, vamos automatizar o processo de copiar a fórmula =J2+L2 com ajuda de uma **Macro**. (Uma macro é um processo automatizado que evita a repetição manual de comandos, fazendo com que eles sejam realizados de forma automatizada poupando tempo e esforço.)

Siga os seguintes passos para criar nossa macro:

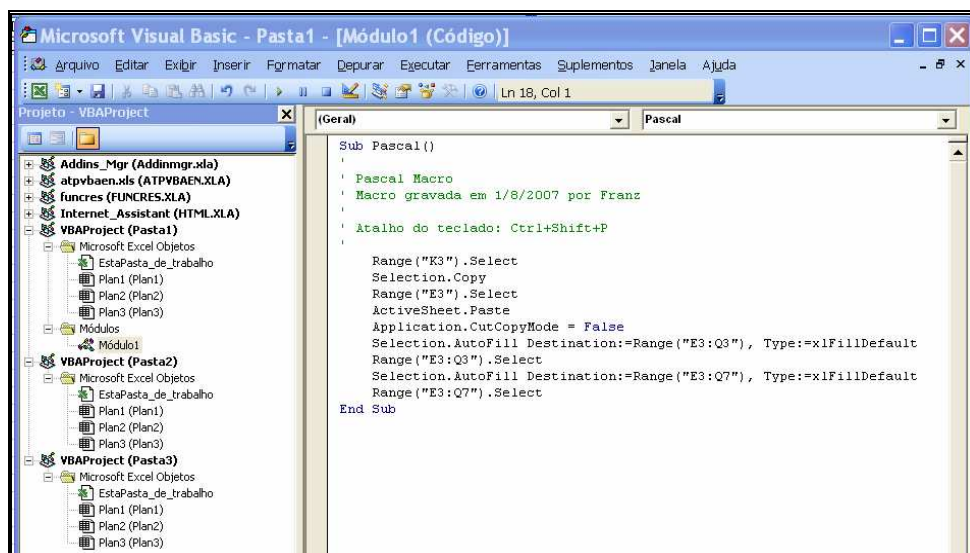
1. Digite 1 em K1, 1 em J1, 1 em L1: =J2+L2 em K3
2. Clique em *Ferramentas>Macros>Gravar nova macro* para ativar o gravador de macros. Na tela que surgirá, digite como *Nome da macro* Pascal, *Tecla de atalho* p (melhor Shift p, o que será na tela Ctrl+Shift p, ou seja P). Cuidado, pois se existir algum comando que utilize as mesmas teclas, ele será desabilitado, e a macro passará a funcionar em seu lugar.
3. Na caixa *Armazenar macro em* deixamos "Esta pasta de trabalho". Assim, a macro será armazenado juntamente com a planilha atual. Na caixa *Descrição*, você pode escrever o que quiser.
4. Ao pressionar *OK*, aparecerá uma pequena tela com dois botões. (Utilizamos o quadrado azul para **parar** a gravação do macro. O outro botão se usa para alterar entre gravação absoluta e relativa.) Depois de aparecer esta pequena tela, o gravador de macro registrará todos os passos que você vai fazer na planilha: posicione o cursor na célula K2, copie com Ctrl-C, vá na célula E3, Ctrl-V, arraste a aba de preenchimento até Q3, e depois até Q7. Pressione o botão azul, para parar a gravação.

Uma vez que você já tenha criado a macro, você deseja rodá-la. Você pode usar Ctrl+Shift-p ou as teclas Alt-F8 e pressionar **Executar**.

A sua planilha vai ter o seguinte aspecto (que depende de como você terminou o quarto passo):

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1											1	1						
2										1		1						
3					0	0	0	0	1	0	2	0	1	0	0	0	0	0
4					0	0	0	1	0	3	0	3	0	1	0	0	0	0
5					0	0	1	0	4	0	6	0	4	0	1	0	0	0
6					0	1	0	5	0	10	0	10	0	5	0	1	0	0
7					1	0	6	0	15	0	20	0	15	0	6	0	1	1
8																		
9																		

Para evitar erros, é aconselhável de bem pensar nos passos a fazer durante a gravação. Se errar, poderá ser mais fácil começar tudo de novo do que tentar consertar o erro. Se deseja ver o código gerado durante a gravação, pressione as teclas Alt-F11 e clique em *Ferramentas>Macros>Editar*. É muito provável que você verá um código parecido ao seguinte:



O seu código na janela de código pode ser diferente, isso depende dos detalhes da gravação. (Não se preocupe agora da janela da esquerda, mais tarde vamos trabalhar muito com ela.) O objeto *Range* seleciona a célula K3. Em seguida, o método de cópia do objeto *Selection* é usado e *Selection* representa a célula selecionada atualmente. Depois E3 é selecionada etc.

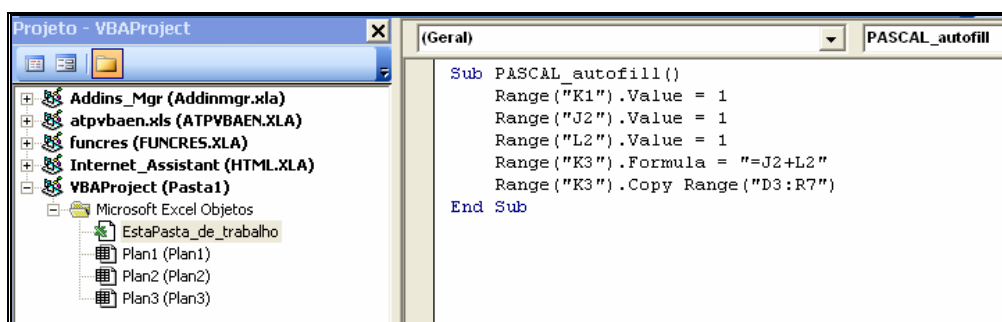
Quando você gravar uma macro, um *módulo* será criado e o código será gravado nele. O gravador de macros cria uma sub-rotina, por isso vemos o *Sub* no início do programa. Toda sub-rotina deve terminar com *End Sub*.

O que vemos, é um programa com o nome "Pascal" na linguagem **VBA** (Visual Basic for Applications). O VBA é uma linguagem de programação incorporada ao pacote Office e será uma ferramenta muito importante em nossos trabalhos com o Excel. Você vai gostar! (O editor de VBA do Excel 2007 é praticamente idêntico ao editor das versões anteriores de Excel. Também pode-se abrir com ALT-F11.)

Aqui seguem alguns **exemplos** em VBA. Para escrever o código utilizamos Alt-F11 e depois clique duplo sobre *EstaPasta_de_trabalho*.

1. Neste exemplo vemos que também é possível colocar os valores iniciais automaticamente na planilha:

Pressione Alt-F11, clique duas vezes sobre *EstaPasta_de_trabalho*, e entre o código na janela de código. Depois pressione F5, para executar a macro.



2. Em VBA precisamos só uma linha de código para produzir o triângulo:

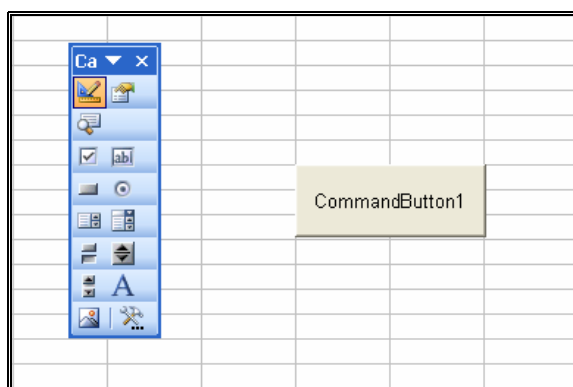
```

CommandButton1 Click
Private Sub CommandButton1_Click()
Range("K3").Copy Range("D3:R7")
End Sub

```

Este programa vem com um **botão** de *Exibir>Barra de ferramentas>Caixa de ferramentas de controle*.

(O Excel oferece dois tipos de botões: os **ActiveX** da *Caixa de ferramentas de controle* e os botões da coleção **Formulários**. Este segundo tipo usaremos no próximo capítulo.)



Clique no ícone do botão e desenhe-lo, sem pressionar nenhum botão do mouse, sobre a planilha. Em seguida clique duas vezes no botão, abre-se uma janela para escrever a linha do código. Salvar o programa com um nome apropriado (não necessariamente Pasta1!). Clique em Executar para ativar a macro (ou use F5). Depois volte para a planilha Excel e coloque um 1 em K1, J2, L2 e anote em K3 a fórmula =J2+L2. Dê um Click no botão – e tudo pronto! (Em **2007** se faz assim: *Desenvolver>Controles de ActiveX* etc.)

E	F	G	H	I	J	K	L	M	N	O	P	Q	R
						1							
0	0	0	0	1	0	2	0	1	0	0	0	0	
0	0	0	1	0	3	0	3	0	1	0	0	0	
0	0	1	0	4	0	6	0	4	0	1	0	0	
0	1	0	5	0	10	0	10	0	5	0	1	0	
1	0	6	0	15	0	20	0	15	0	6	0	1	

3. Se queremos colorir todas as células > 0 de amarelo, podemos digitar as seguintes linhas:

```
Private Sub CommandButton1_Click()
Dim Cell As Range ' importante!
Range("K3").Copy Range("D3:R7")
For Each Cell In Range("D1:R7")
    If Cell.Value > 0 Then
        Cell.Interior.ColorIndex = 6 'amarelo
    End If
Next Cell
End Sub
```

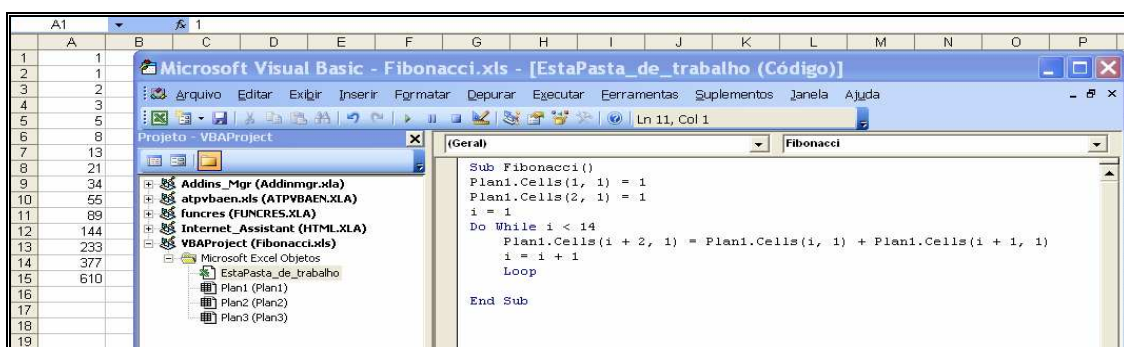
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
							1							
0	0	0	0	0	1	0	2	0	1	0	0	0	0	0
0	0	0	0	1	0	3	0	3	0	1	0	0	0	0
0	0	0	1	0	4	0	6	0	4	0	1	0	0	0
0	0	1	0	5	0	10	0	10	0	5	0	1	0	0
0	1	0	6	0	15	0	20	0	15	0	6	0	1	0

CommandButton1

Com *Editar>Limpar>tudo* podemos limpar a planilha.

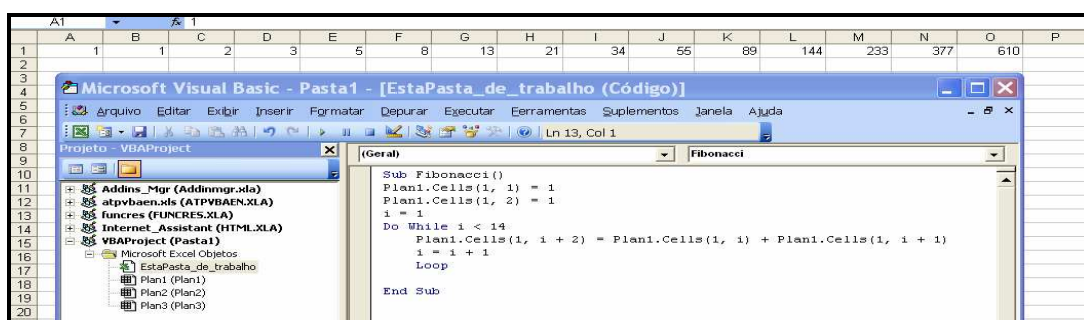
O código contém um `For...Next` loop com uma instrução condicional `IF...Then`. Afortunadamente, todo isso lê-se como se fosse português puro. Também as instruções condicionais múltiplas são simples e funcionam quase como uma frase em português ou inglês. Elas usam os operadores `And` e `Or` e significam exatamente o que querem dizer em inglês. Devido ao grande uso que vamos fazer dessas instruções, você vai facilmente acostumar-se a elas.

4. Facilmente podemos produzir uma planilha para criar os números de FIBONACCI. `Alt-F11`; 2x *EstaPasta_de_trabalho*; código; salvar; `F5`. O código pode ser o seguinte:



No código vemos o `Do While ... Loop` que continua sua execução enquanto a condição $i < 14$ estiver sendo atendida. Primeiro as células A1 e A2 obtêm cada uma o valor 1. Em seguida, a variável i é configurada com o valor 1. Já que $1 < 14$, calcula-se o valor para A3. Ele será $=A1+A2 = 2$. Depois, a variável i será aumentada em uma unidade, ou seja i vai ser igual a 2. O `Loop` manda o programa para ver se $2 < 14$... O `Loop` será executado por última vez quando $i = 13$, com $A15 = A13 + A14 = 233 + 377 = 610$.

Os resultados vão aparecer na primeira **linha** em vez da primeira **coluna** se você trocar os índices das células:



Você pode definir a seqüência de Fibonacci também pelo seguinte conjunto de duas fórmulas:

1. $FIB(1) = FIB(2) = 1$
2. $FIB(n) = FIB(n-1) + FIB(n-2)$, para $n > 2$

Funções lógicas

Agora vamos ver alguns exemplos, nos quais a função lógica `SE` vai ter um papel decisivo.

Sintaxe: `SE(teste_lógico; valor_se_verdadeiro; valor_se_falso)`

1. Suponhamos que desejasse criar um Controle de Notas de Aluno, onde ao se calcular a média, ele automaticamente especificasse se o aluno fora aprovado ou não. Poderíamos fazer o seguinte:

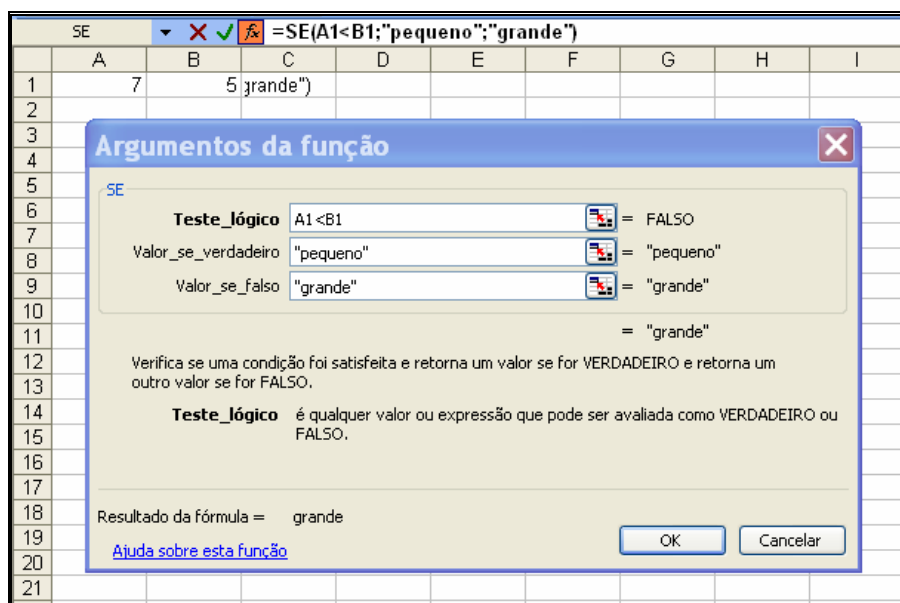
	A	B	C	D
1	Aluno	Média	Situação	
2				
3	Alonso	8	aprovado	
4				
5				
6				

É possível aninhar até sete funções SE. Na seguinte planilha utilizamos três:

	A	B	C	D	E
1	Aluno	Média	Situação		
2					
3	Alonso		4	insuficiente	
4	Carla		9	ótimo	
5	Desmond		8	bom	
6	Emilia		7	regular	
7	Franz		2	insuficiente	
8	Gerardo		5	insuficiente	
9	Hilda		10	ótimo	
10	Idomeneo		7	regular	
11					
12					

Para preencher a coluna da média automaticamente, colocamos em B3 a fórmula `=INT(ALEATÓRIO()*9)+2` que depois copiamos até B10. `ALEATÓRIO()*9` produz um número aleatório entre 0 e 9, a função `INT` arredonda um número para baixo até o número inteiro mais próximo. Para trabalhar com uma nova série de números aleatórios, é só preciso apertar a tecla de cálculo F9. Em vez de `ALEATÓRIO` podemos usar `ALEATÓRIOENTRE`, veja o próximo exemplo.

Você pode poupar trabalho, se clicar sobre o botão f_x e trabalhar com o recurso *Inserir Função*. Escolhe SE e utilize a janela que se apresenta. Veja a figura:



Compare as explicações seguintes:

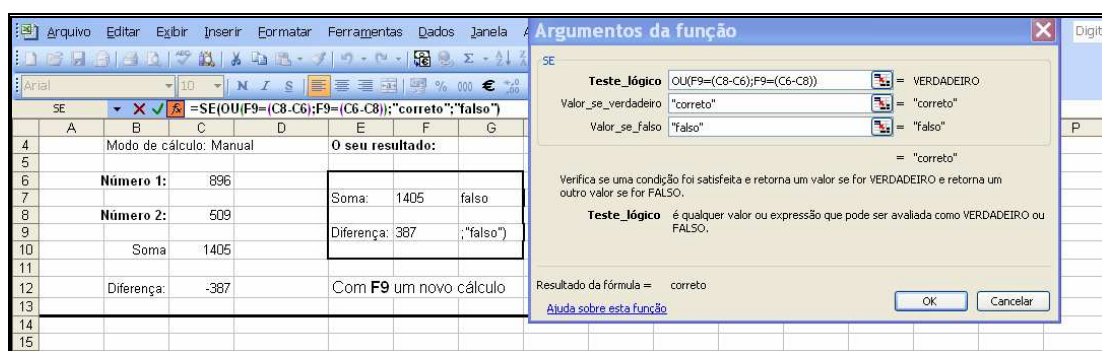
A	
1	Dados
2	15
3	9
4	8
Fórmula	Descrição (resultado)
=SE(A2=15; "OK"; "Não OK")	Se o valor na célula A2 for igual a 15, retornar "OK". (OK)
=SE(E(A2>A3; A2<A4); "OK"; "Não OK")	Se 15 for maior que 9 e menor que 8, retornar "OK". (Não OK)
=SE(OU(A2>A3; A2<A4); "OK"; "Não OK")	Se 15 for maior que 9 ou menor que 8, retornar "OK". (OK)

2. Para mostrar um exemplo com SE(OU...) vamos desenvolver um programa para praticar as técnicas de somar e restar dois números.

Criamos aleatoriamente os dois números entre 0 e 1000 nas células C6 e C8 com a nova função =ALEATÓRIOENTRE(0;1000) na forma =INT(ALEATÓRIOENTRE(0;1000)). Em G7 temos =SE(F7=(C6+C8);"correto";"falso"). Na célula F7 escrevemos nossa soma. Se for diferente de C6+C8, aparecerá em G7 "falso". A fórmula para a diferença contém SE e OU na combinação =SE(OU(F9=(C8-C6);F9=(C6-C8));"correto";"falso"). Isso significa que vamos considerar como sendo correta a diferença calculada como C8-C6 ou como C6-C8. (Os resultados em C10 e C12 podemos ocultar colorindo as células em branco.)

Para usar o programa, escolhemos primeiro o **modo manual** de cálculo. Isso fazemos com *Ferramentas>Opções>Cálculo>Manual*, depois pressione F9, para efetuar um novo cálculo.

Em **2007** pressione o botão do Microsoft, depois *Opções do Excel>Fórmulas>Opções do cálculo>Manual*.



Para fazer o teste lógico, colocamos o cursor na célula G7 e com o botão da mouse clicamos no símbolo f_x . Como mostrado na figura, veremos a janela da função SE. Clique em *OK*. Em G7 aparecerá "falso" ou "correto". Em seguida clicamos na G9 e depois clicamos com o cursor no ícone f_x . Dê um clique no *OK* da janela do SE. Usando F9, efetuamos um novo cálculo.

Para simplificar tudo, vamos registrar os passos do teste lógico numa macro.

	A	B	C	D	E	F	G	H	I	J
1										
2		Treinador de Aritmética								
3										
4		Modo de cálculo: Manual			O seu resultado:					
5										
6		Número 1:	636							
7					Soma:	1618	correto			
8		Número 2:	982							
9					Diferença:	45	falso			
10		Soma	1618							
11										
12		Diferença:	346		Com F9 um novo cálculo					
13										
14										
15										
16										
17										
18										
19										

O código da macro do teste lógico pode ser assim:

```

CommandButton1
Click

Private Sub CommandButton1_Click()
Range("G7").Select
ActiveCell.FormulaR1C1 = _
"=IF(RC[-1]=(R[-1]C[-4]+R[1]C[-4]),""correto"", ""falso"")"
Range("G9").Select
ActiveCell.FormulaR1C1 = _
"=IF(OR(RC[-1]=(R[-1]C[-4]-R[-3]C[-4]),RC[-1]=(R[-3]C[-4]-R[-1]C[-4])), ""correto"", ""falso"")"
End Sub

```

Como acima descrito, conectaremos a macro também com um Command-Button. (Clique no ícone do botão e desenhe-lo, sem pressionar nenhum botão do mouse, na planilha. Em seguida clique duas vezes no botão, e abre-se uma janela para escrever as linhas do código. Salvar o programa com um nome apropriado (por exemplo "Calculadora"). Clique em Executar ou F5 para ativar a macro. Em seguida regresse na planilha.) R = row (linha), C = column (coluna); isso explicaremos mais adiante.

Páscoa

Agora vamos determinar a data do domingo de Páscoa para qualquer ano depois de 1582. Será interessante ver a aplicação das três funções lógicas SE, OU, E na célula B11 da seguinte planilha.

O algoritmo para nossa planilha foi desenvolvido por Aloysius Lilius e Christoph Clavius. (Páscoa é o primeiro domingo depois da primeira lua cheia, o que sucede no 21 de Março ou depois no mês de Abril.)

	A	B	C	D	E	F	G	H	I	
1										
2	Domingo de Páscoa						Ano:	2008		
3										
4										
5										
6	G	C	X	Z	D	E1	E			
7	14	21	3	1	2497	172	22			
8										
9										
10	E<0	Epact	N	N<21?	N1		Domingo de Páscoa:			
11	22	22	22	22	23		Abril	Março		
12								23		
13										

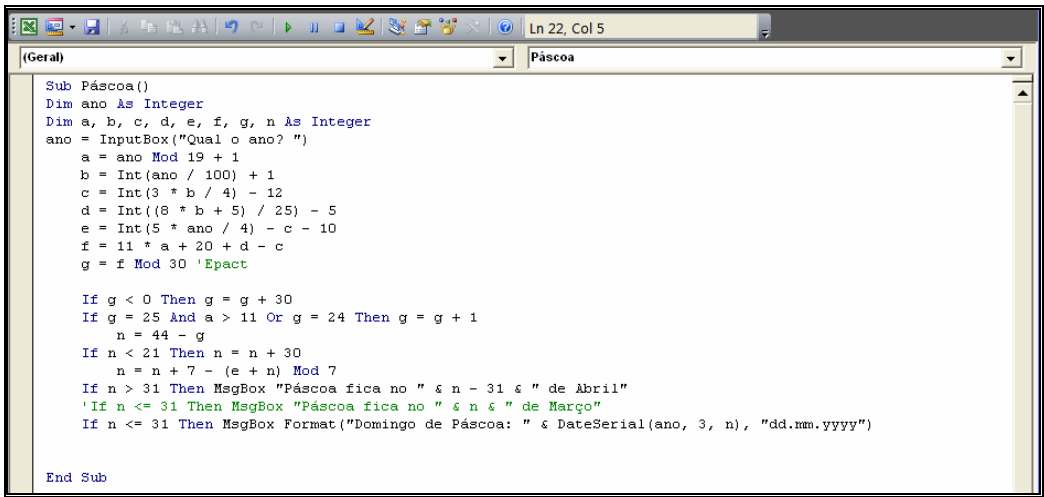
Precisaremos das seguintes informações:

1. O ano em questão J
2. O número de ouro G: $(J \text{ MOD } 19)+1$
3. O número do século C: $\text{INT}(J/100)+1$
4. Correções X: $\text{INT}(3C/4)-12$
" Z: $\text{INT}(8C+5)/25)-5$
5. Número de domingo D: $\text{INT}(5J/4)-X-10$
6. Número do "Epact" E: $(11G+20+Z-X) \text{ MOD } 30$
Se $E = 25$ e $G > 11$, ou se $E = 24$, então aumente E em 1
(esta condição composta fica em B11)
7. Número da lua cheia N: $44 - E$
Se $N < 21$, então aumente N em 30
8. Critério N1: $N + 7 - ((D + N) \text{ MOD } 7)$
9. Se $N > 31$, então Páscoa fica em N-31 de Abril, senão em N de Março

Sem entrar numa discussão deste algoritmo, levaremos todas essas informações numa planilha, compare também a figura acima.

H2: J
A7: =MOD(H\$2;19)+1
B7: =INT(H\$2/100)+1
C7: =INT(3*B7/4)-12
D7: =INT((8*B7+5)/25)-5
E7: =INT(5*H\$2/4)-C7-10
F7: =11*A7+20+D7-C7
G7: =MOD(F7;30)
A11: =SE(G7<0;G7+30;G7)
B11: =SE(OU((E(G7=25;A7>11));G7=24);G7+1;G7)
C11: =44-G7
D11: =SE(C11<21;C11+30;C11)
E11: =D11+7-MOD(E7+D11;7)
G11: =SE(E11>31;(E11-31);"
H11: =SE(E11<=31;E11;")

Esse exemplo chama para ser transformado numa sub-rotina de VBA. Precisaremos de uma série de instruções If para processar as diferentes condições do algoritmo. (Outra instrução disponível no VBA para tais situações é a instrução **Select Case**, compare o capítulo 3 sobre *Procedimentos*, p. 10.) Uma instrução IF dentro de uma linha só, não termina com End IF, já que assim não forma nenhum bloco. Os resultados são exibidos dentro de uma caixa de mensagem. A versão com a função **DateSerial** é especialmente interessante, pois ela mostra a data de Páscoa no formato dia/mês/ano



```

Sub Páscoa()
Dim ano As Integer
Dim a, b, c, d, e, f, g, n As Integer
ano = InputBox("Qual o ano? ")
a = ano Mod 19 + 1
b = Int(ano / 100) + 1
c = Int(3 * b / 4) - 12
d = Int((8 * b + 5) / 25) - 5
e = Int(5 * ano / 4) - c - 10
f = 11 * a + 20 + d - c
g = f Mod 30 'Epact

If g < 0 Then g = g + 30
If g = 25 And a > 11 Or g = 24 Then g = g + 1
n = 44 - g
If n < 21 Then n = n + 30
n = n + 7 - (e + n) Mod 7
If n > 31 Then MsgBox "Páscoa fica no " & n - 31 & " de Abril"
'If n <= 31 Then MsgBox "Páscoa fica no " & n & " de Março"
If n <= 31 Then MsgBox Format("Domingo de Páscoa: " & DateSerial(ano, 3, n), "dd.mm.yyyy")

End Sub

```

Exemplos: 1793 (31 de Março); 1818 (22 de Março); 2007 (8 de Abril)

Nas seguintes linhas construímos um *bloco* para a instrução If ... Then ... Else, a segunda condição "IF n<=31" foi absorvido pelo Else

```

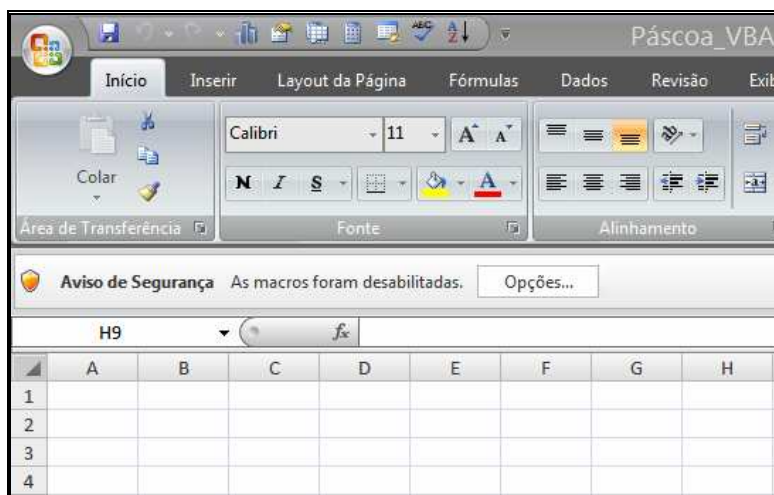
If n > 31 Then
    MsgBox "Páscoa fica no " & n - 31 & " de
    Abril"
Else
    MsgBox "Páscoa fica no " & n & " de Março"
End If

```

O algoritmo foi tirado do livro *The Art of Computerprogramming* p. 155 de D.E. Knuth. O comentário de Knuth deve ser lido: There are many indications that the sole important application of arithmetic in Europe during the Middle Ages was the calculation of Easter date, and so such algorithms are historically significant.

Veja outro algoritmo e muitos exemplos no site
<http://www.inf.ufrgs.br/~cabral/Pascoa.html>

Segurança de Macros, Depurar o Código



1. Quando carregar uma macro, Excel **2007** mostra um Aviso de Segurança que deve ser respondido clicando sobre *Opções*. (Clique, se for preciso, sobre o "Botão Office" e escolhe *Opções do Excel*, onde encontra a *Central de Confiabilidade*. Em *Configurações da Central ...* pode marcar *Mostrar a Barra de Mensagens*.)
2. Existem diversas maneiras para controlar e **depurar** o código.

```

(Geral) Páscoa
Sub Páscoa()
Dim ano As Integer
Dim a, b, c, d, e, f, g, n As Integer
ano = InputBox("Qual o ano? ")
a = ano Mod 19 + 1
b = Int(ano / 100) + 1
c = Int(3 * b / 4) - 12
d = Int((8 * b + 5) / 25) - 5
e = Int(5 * ano / 4) - c - 10
f = 11 * a + 20 + d - c
g = f Mod 30 'Epact

```

Na planilha **Domingo de Páscoa** temos todos os resultados intermediários indicados. Eles nos ajudam a controlar o nosso código, pois Excel nos deixa executar o código uma linha por vez. Coloque o cursor em algum lugar do código. Depois, pressione a tecla F8 uma vez. Continue teclando F8 até passar o ponto de interesse. Quando você executa o código, até a linha amarela chegar até a variável *b*, veja a figura, você já pode ver o valor de *a*, movendo o cursor sobre esta variável. O valor de *a* vai ser 14, se colocou o ano 2008.

(Com F9 pode-se colocar e remover pontos vermelhos de interrupção. Com Ctrl+Shift+F9 pode-se limpar todos os pontos de interrupção.)

Outra possibilidade de verificar o código é com *Depurar>Compilar VBA-Project >Executar até o Cursor*. Se você colocou o cursor diante da variável *g*, a compilação do código vai parar em *g*, e com o cursor pode-se inspecionar todos os valores anteriores. Você também pode ..., pois os métodos de depuração são numerosos. A melhor maneira de conhecê-los, é experimentar sem olhar no relógio.

Com *Verificação Imediata* (Ctrl+G) podemos ver os resultados intermediários escrevendo `Debug.Print`.