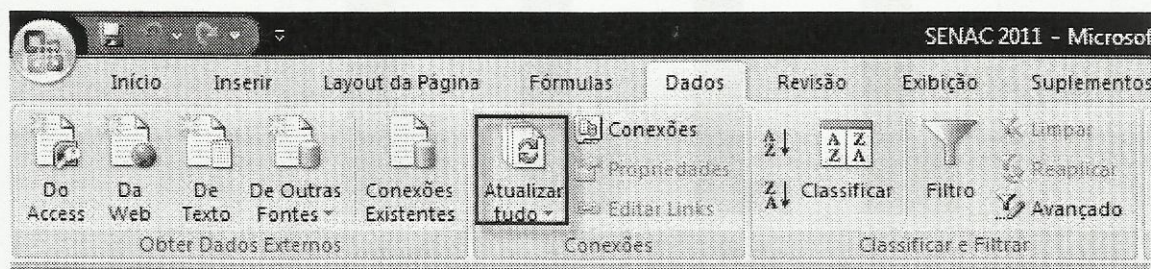


CUIDADO!

Ao se acrescentar algum dado na tabela dinâmica, não ocorrerá a atualização da tabela e gráfico no momento da inserção deste ou destes novos dados. Será preciso atualizar os dados da tabela dinâmica. Para isso devemos:

DADOS → ATUALIZAR TUDO



CONCLUSÃO DE TABELA E GRÁFICOS DINÂMICOS!

Tabelas e gráficos dinâmicos podem ou não estarem relacionadas. Se quiser ambos os relatórios, é possível que sua criação ocorra. Mas o importante é saber que o gráfico está ligado de certa forma com a Tabela dinâmica. Assim, ao alterarmos o layout da tabela dinâmica, mudaremos também o layout do gráfico.

Isso é um enorme facilitador, uma vez que não precisamos ficar criando gráficos separados para cada caso.

12. MACROS

→ INTRODUÇÃO

Macro é uma automatização de uma seqüência de tarefas. Ou seja, é uma seqüência de comandos e funções de forma lógica com a finalidade de automatizar tarefas ou ações do Microsoft Excel. Podemos concluir então, que uma macro pode ser considerada um programa.

→ TIPOS DE MACROS

Existem basicamente 3 tipos de macros: macros de comandos, de funções definidas pelo usuário e macros de sub-rotina.

A macro de comando é o tipo mais comum de procedimento, possui equivalência a comandos de menu e caixas de diálogo. Logo, são as macros que podemos gravar.

Quando se grava uma macro, o Excel grava a seqüência de ações que foram realizadas, à medida que você executa uma seqüência de comandos. Logo, como a macro grava tudo que você clica na barra de ferramentas, é bom que você saiba o que esta querendo executar para se evitar erros.

A macro de funções é um tipo de procedimento que trabalha de forma idêntica às funções existentes no Excel. A única diferença é que você é o criador das funções.

Por fim, a macro de sub-rotina é uma junção dos tipos anteriores. Caracteriza-se pela possibilidade de ser chamada de dentro de outras macros, permitindo com isso minimizar o uso de tarefas repetitivas, o que torna o código de programação mais leve.

→ EDITOR DE MACROS

Com o editor Visual Basic, você pode editar macros, copiá-las de um módulo para outro, bem como entre pastas de trabalho diferentes. Esse editor pode ser chamado pelo atalho **ALT + F11**.

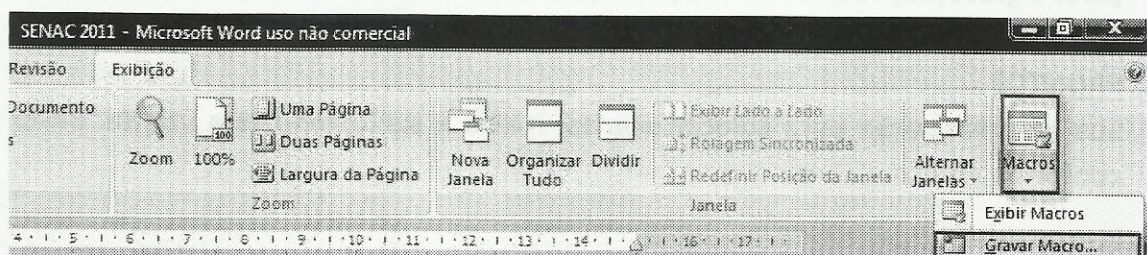
12.1 CRIANDO UMA MACRO

Para facilitarmos o aprendizado de macros e suas utilidades, vamos fazer exercícios práticos para que você entenda como a ferramenta se comporta.

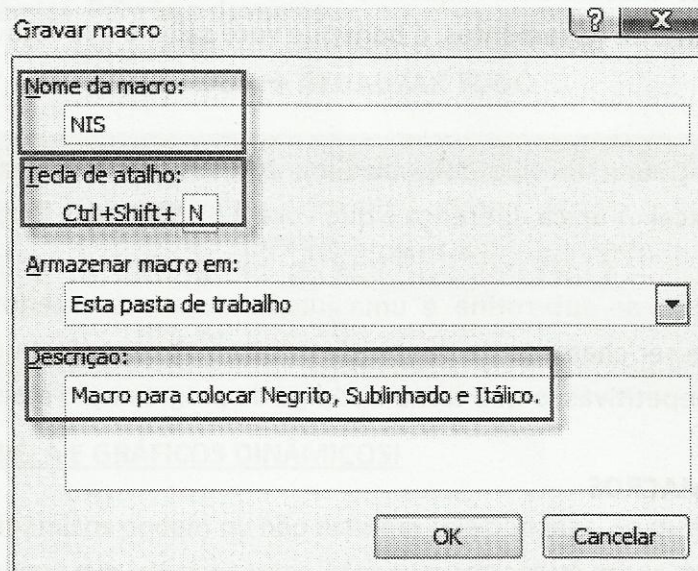
Caso 1: No decorrer do dia a dia em empresas, muitas vezes temos que fazer formatações repetitivas como colocar negrito, itálico e sublinhado em determinados campos da planilha. Para evitarmos ficar colocando manualmente esse tipo de formatação, clicando constantemente nos botões (**N I S**), podemos criar uma macro para fazer essa ação para nós. **Veja como proceder.**

1º Passo → Devemos ir:

EXIBIÇÃO → MACROS → GRAVAR MACRO

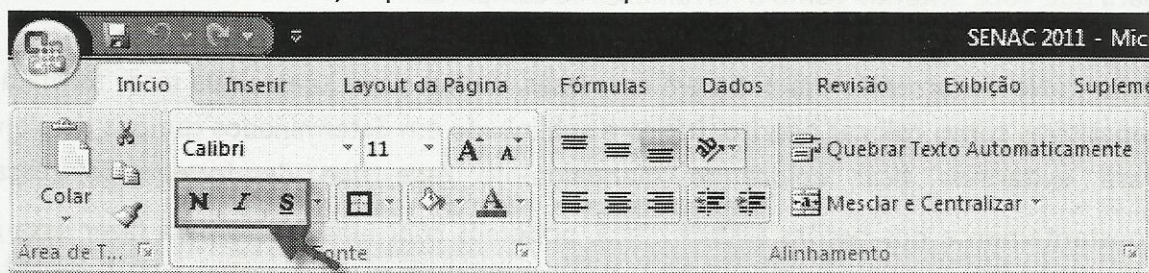


2º Passo → No momento que clicar em gravar macro, aparecerá uma tela para que você informe o nome da macro, uma tecla de atalho e também informar uma pequena descrição do que macro faz. Isso ajuda a organizar suas macros, que poderão ser muitas.



3º Passo → Após clicar em OK, a macro esta sendo gravada. Para verificar mesmo se sua macro esta gravando, vá no mesmo local onde você iniciou a gravação da macro (EXIBIR → MACROS) e verifique o botão ativo chamado (Parar gravação).

4º Passo → Como verificamos que a macro esta sendo realmente gravada, isso implica que onde você clicar na barra de ferramentas, o Excel estará registrando e gravando. Então, basta clicar em **NEGRITO**, depois no **ITÁLICO** e por fim no **SUBLINHADO**.



5º Passo → Como o objetivo de nossa Macro é colocar o negrito, itálico e sublinhado, e acabamos de fazer isso no passo anterior, devemos agora somente parar a gravação. Para parar a gravação, você dever ir a (EXIBIÇÃO → MACROS → Parar gravação).

PRONTO!

Sua automatização para colocar negrito, itálico e sublinhado esta pronta. Basta testá-la conforme abaixo.

→ **ANTES:** Nota que os nomes das pessoas não possuem negrito, nem itálico e sublinhado. Para testar sua automatização, selecione todos os nomes e digite a senha que você usou para gravar a macro (CTRL + SHIFT + N)

A	B	C	D	E
TESTE DA MACRO - NIS				
JOÃO	PEDRO	MATIAS	PAULA	JOANA
CRISTIANE	SAULO	GLAUBER	RAMON	VITOR
REGINA	JOANA	DIEGO	MARIA	JOSÉ

→ **DEPOIS:**

Veja o resultado final.

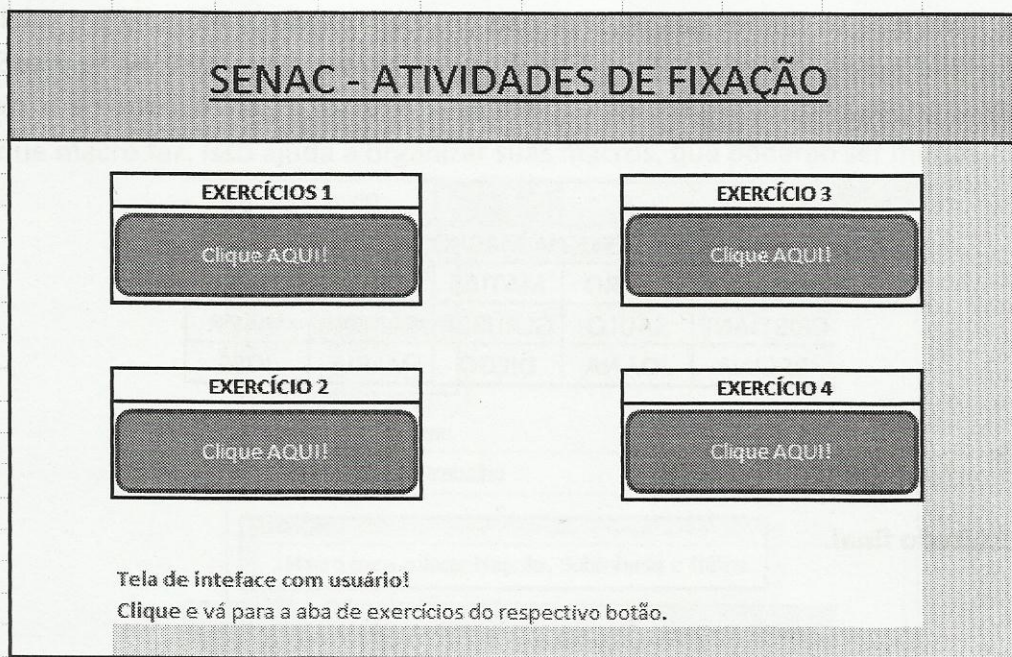
A	B	C	D	E
TESTE DA MACRO - NIS				
<u>JOÃO</u>	<u>PEDRO</u>	<u>MATIAS</u>	<u>PAULA</u>	<u>JOANA</u>
<u>CRISTIANE</u>	<u>SAULO</u>	<u>GLAUBER</u>	<u>RAMON</u>	<u>VITOR</u>
<u>REGINA</u>	<u>JOANA</u>	<u>DIEGO</u>	<u>MARIA</u>	<u>JOSÉ</u>

NOTA:

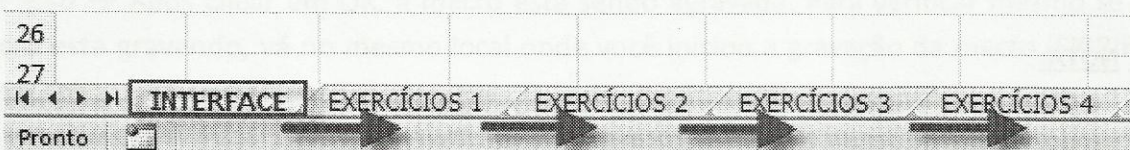
Perceba que fizemos uma automatização que usa elementos da barra de ferramentas do Excel. Assim sendo, você pode seguir o mesmo processo para construir macros que façam inserção de novas planilhas, excluir planilhas, apagar dados, formatar tabelas... Para isso, basta que você ao iniciar a gravação da macro clique nos botões que executam tais ações e mande parar depois a gravação.

Caso 2: Vamos agora criar botões como os da web, que ao serem acionados direcionam para outras planilhas. Veja o modelo abaixo.

→ **Interface:**



→ Planilhas:

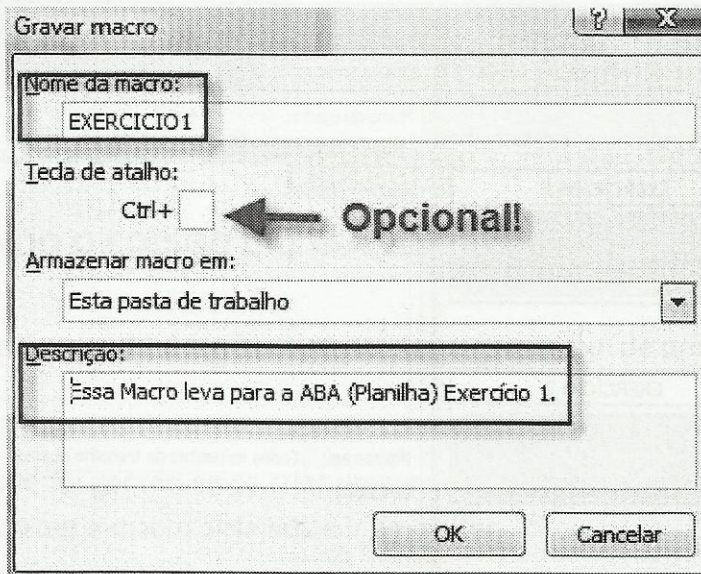


1º Passo → O processo de construção dessa macro é semelhante a outra que já criamos. A única diferença é que após a criação da macro, **NÃO** a acionaremos através de atalhos (CTRL + SHIFT + ...) e sim atribuiremos essa macro a um botão. Esse botão será responsável por chamar a macro e acionar o que ela é responsável por fazer. Tudo isso com o clique!

Então: Vá em

EXIBIÇÃO → MACROS → GRAVAR MACRO.

2º Passo → Grave a macro seguindo o mesmo processo do "Caso 1". De o nome para a macro de EXERCICIO1 para que você possa achá-la depois. Veja abaixo.



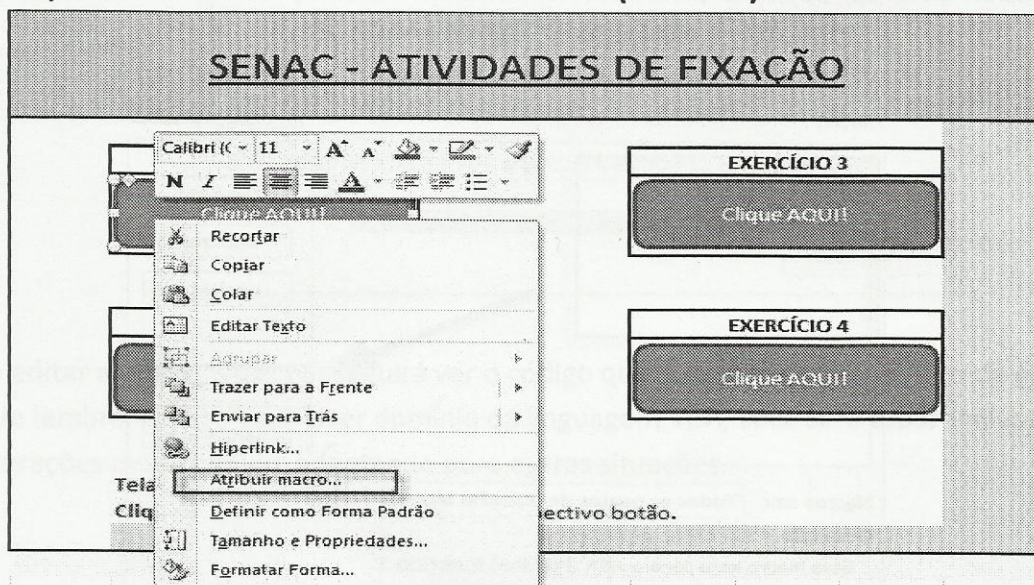
3º Passo → Após clicar em OK, como sabemos, a macro já esta sendo gravada e tudo que você fizer com o cursor do mouse estará sendo registrado. Então, basta clicar com o mouse na aba “EXERCÍCIO 1”, que é para onde o primeiro botão (Interface) o levará quando for acionado e depois **PARAR A GRAVAÇÃO**.

27

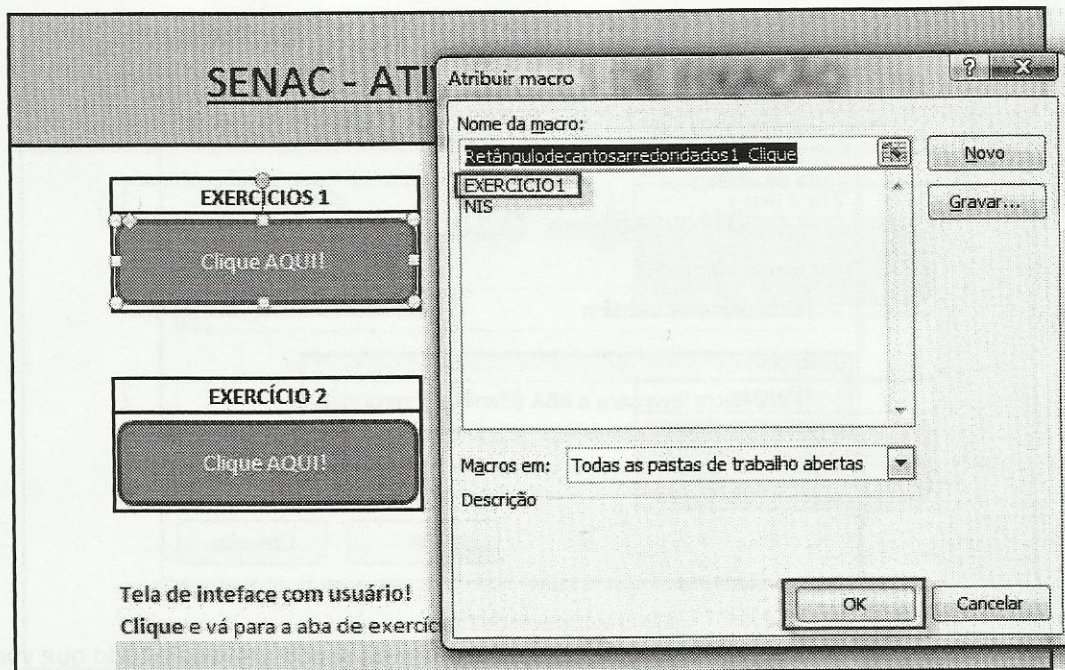


4º Passo → Com a automatização gravada, basta agora atribuir a sua macro ao botão que você tem criado na interface da seguinte forma.

- **Clique com o botão direito em cima do botão (Interface) e selecione ATRIBUIR MACRO.**



- **Escolha a macro “EXERCICIO1” que foi a que criamos para nos levar a planilha EXERCÍCIO 1 e depois em OK.**



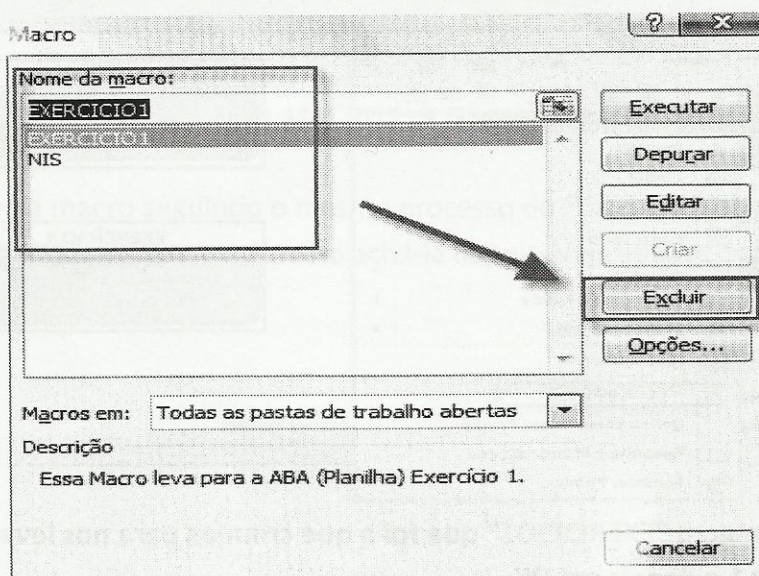
- **PRONTO!**

Feito isso, a macro que você criou já está pronta. Para saber se o botão agora está automatizado, basta passar o mouse em cima do mesmo. Se aparecer uma mãozinha, quer dizer que o botão da interface possui uma macro associada a ele. CLIQUE E VEJA O RESULTADO!

12.2 EXCLUINDO MACROS

Para excluir uma macro, basta que vá ao mesmo local onde é gravada automatização.

EXIBIÇÃO → MACROS → EXIBIR MACROS.



Basta agora selecionar qual macro deseja apagar e clicar em EXCLUIR.

PRONTO!

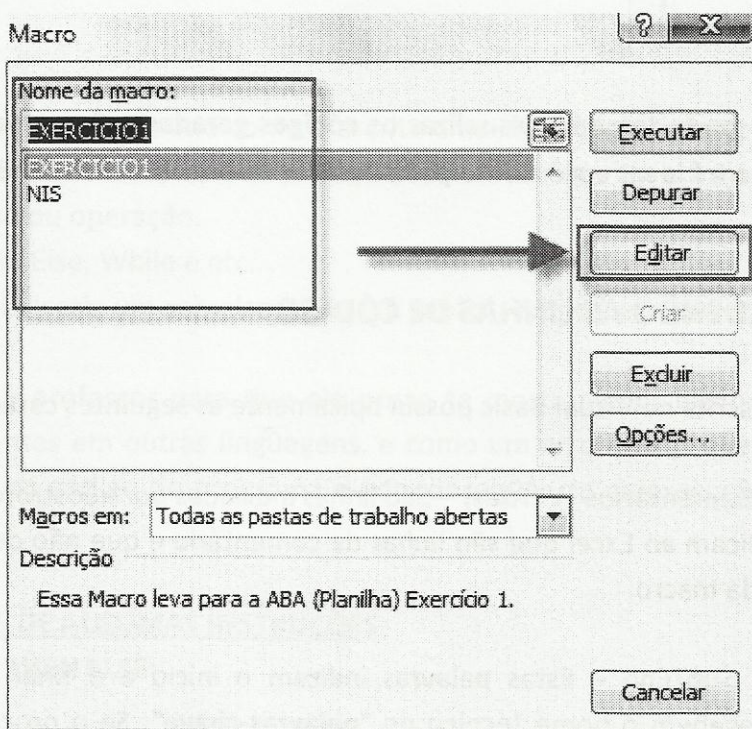
A exclusão da macro esta feita. Se tentar executá-la agora, verá que não será mais possível.

12.3 VISUALIZANDO O CODIGO DAS MACROS

Toda macro quando é gravada, é também escrita numa linguagem de programação chamada de VISUAL BASIC.

Para ver o código que foi gerado ao se criar uma macro, basta ir no mesmo local onde você gravou a macro, escolher a macro desejada e ir em editar.

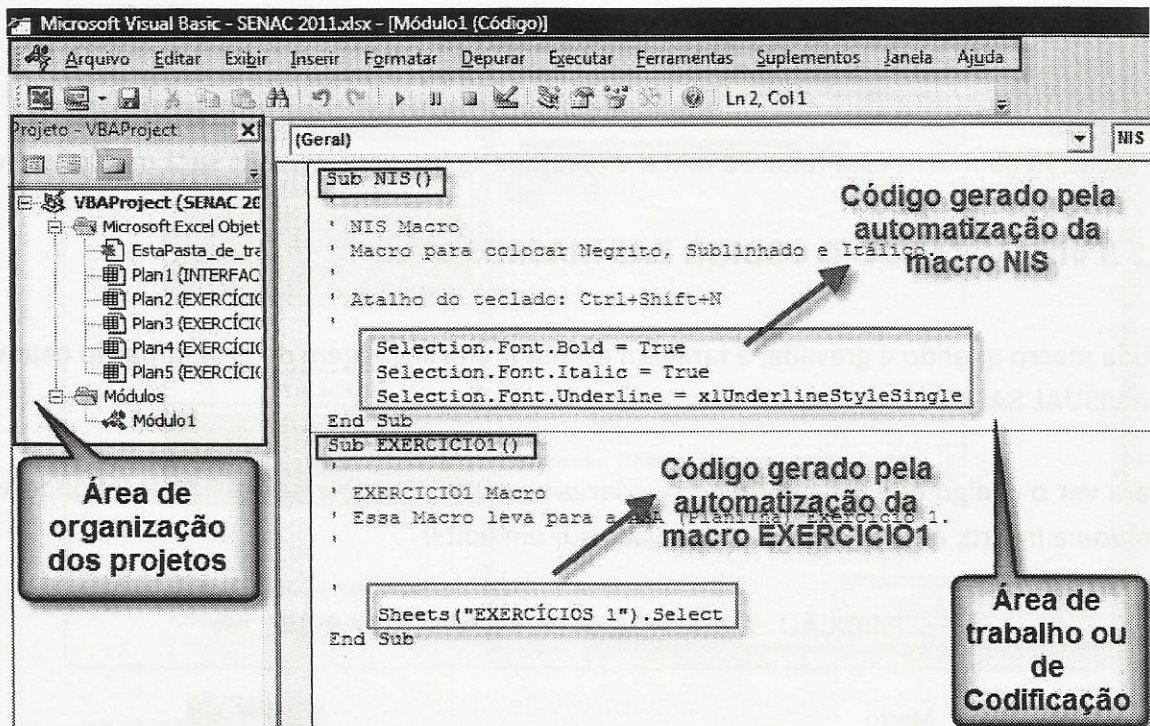
EXIBIÇÃO → MACROS → EXIBIR MACROS → EDITAR



Ao editar a macro, você conseguirá ver o código que foi gerado no momento da gravação.

Vale lembrar que se você tiver domínio da linguagem VBA, você será capaz inclusive de fazer alterações nesse código, adaptando para outras situações.

Veja o resultado:



NOTA: Você pode também visualizar os códigos gerados pela automatizações digitando **ALT + F11**. Isso irá levar você direto para o editor do Microsoft visual Basic (Print acima).

12.4 ENTENDENDO AS LINHAS DE CÓDIGO

Uma macro escrita no Visual Basic possui tipicamente as seguintes características:

- **Linhas de Comentários (Verdes)** – São linhas indicadas por apóstrofo e geralmente na cor verde. Ela indicam ao Excel que são linhas de comentário e que não devem ser processadas na execução da macro.
- **Comando Sub/End** – Estas palavras indicam o início e o final do procedimento de comandos. Recebem o nome técnico de “**palavras-chave**”. Se o procedimento for do tipo função definida pelo usuário, essas palavras serão definidas por “Function/End Function”.
- **Nome do Procedimento** – É a palavra na frente da palavra-chave. São obrigatórios e poderão ser usados para a definição de argumentos no caso de procedimentos e funções definidas pelo usuário.
- **Código Fonte** – É tudo que está inserida entre as palavras-chave e que não sejam comentários. Estas linhas de códigos representam toda a ação monitorada e gravada pela macro.

NOTA: Para sair do Visual Basic Application, feche a janela.

12.5 CODIFICANDO MACROS

Para que você consiga criar macros sem o recurso de gravação do Excel, você deve dominar pelo menos o básico da linguagem VBA. QUE NÃO É A ENFASE DESTE CURSO!

Mas para ajudar ao aluno a ter noção do que vem a ser programação VBA, e até mesmo para que ele se sinta motivado ao aprendizado desta parte da ferramenta, vamos passar algumas instruções de como programar codificando no Excel.

Para isso, temos que conhecer algumas sintaxes e suas respectivas semânticas da linguagem do Excel.

NOTA:

Sintaxe, na informática, é o modo como é escrita uma instrução em determinada linguagem de programação. Já a semântica, é o que aquela instrução representa no momento que ela entra em execução ou operação.

Ex: **Sintaxe** → If, Else, While e etc.

Ex: **Semântica** → IF (seria um se), else (seria um senão), While (seria um enquanto) e etc.

Converse com seu professor para que ele possa te mostrar mais exemplos de sintaxes e semânticas existentes em outras linguagens, e como um erro nessas questões pode causar sérios problemas no código do programa e por consequência na execução do programa ou macro.

NOÇÕES RÁPIDAS DE ALGUMAS INSTRUÇÕES:

→ INSTRUÇÃO IF THEN ELSE:

If <condição> Then

 <se condição verdadeira >

Else

 <se condição falsa>

End If

→ INSTRUÇÃO FOR NEXT:

For <inicialização do Contador> to <Valor> [Step <Valor a incrementar>]

<Instruções a realizar em cada iteração>

Next

→ **INTRUÇÃO DO LOOP:**

Do While <condição>

<instruções a realizar em cada iteração>

<Lembrar de fazer o incremento da variável>

Loop

Caso 1: Vamos resolver um problema simples com o uso de VBA. Suponha que queiramos gerar os resultados de uma turma de alunos que fizeram duas provas e temos assim que gerar a média desses alunos e também a sua situação.

Vamos resolver.

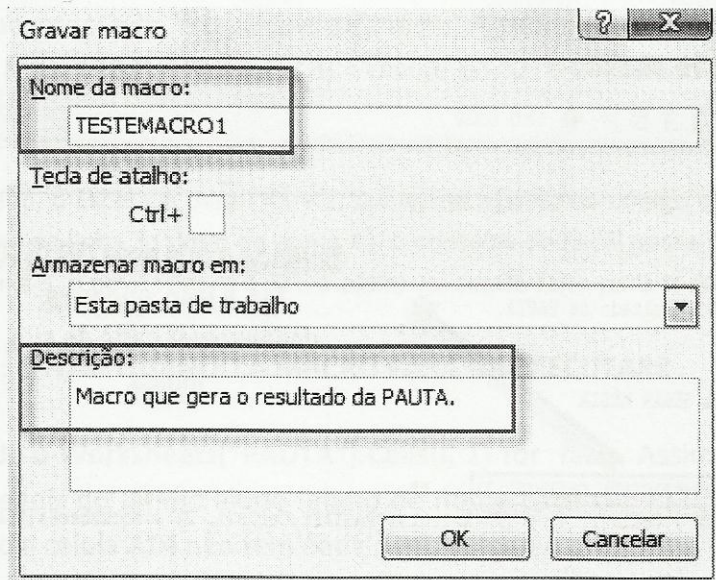
Seja a planilha abaixo:

	A	B	C	D	E
1	PAUTA				
2	ALUNO	NOTA 1	NOTA 2	MÉDIA	SITUAÇÃO
3	A	9,0	10,0		
4	B	5,0	7,0		
5	C	6,0	8,0		
6	D	9,0	6,0		
7	E	8,0	7,0		
8	F	3,0	9,0		
9	G	4,0	6,0		
10	H	4,0	5,0		
11	I	6,0	8,0		
12	J	9,0	8,0		
13	L	5,0	8,0		

1º Passo → Para que facilite pra você, faça da seguinte forma: Grave uma macros com o nome (TesteMacro1) mas não coloque nenhuma instrução nessa macro. Somente vá em:

EXIBIÇÃO → MACROS → GRAVAR MACRO

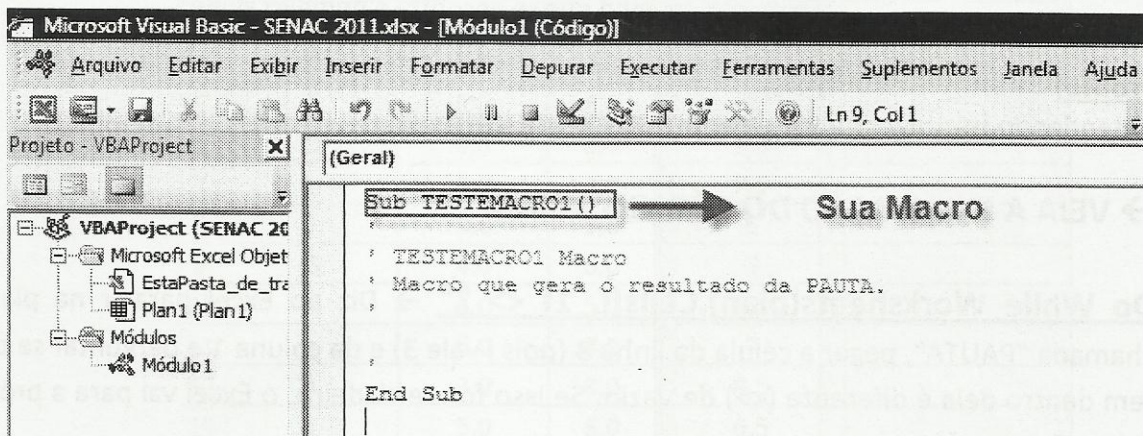
2º Passo → Preencha conforme abaixo (Você já sabe executar esse passo a passo, pois ele não difere do modo como uma macro é criada).



3º Passo → Após clicar em “OK”, já sabemos que a macro começou a sua gravação, e esta somente esperando com que você use a barra de ferramentas para informa o macro o que ela deverá fazer. Mas nesse caso, como o nosso propósito é codificar manualmente, basta que você PARE A GRAVAÇÃO sem fazer nada. Vá em:

EXIBIÇÃO → MACRO → PARAR GRAVAÇÃO

4º Passo → Feito isso, temos uma macro chamada TESTEMACRO1 e sem nada dentro. Que é justamente o que queríamos para codificar. Clique no atalho ALT + F11 e abra o editor do Visual Basic e veja sua macro. É nela que trabalharemos. Veja a tela abaixo.



5º Passo → Neste momento que o aluno deve conhecer as sintaxes e semânticas da linguagem. Pois começaremos a codificar.

NOTA: Não daremos ênfase a todas as regras de programação nem da forma como se escreve todas as sintaxes. Demonstraremos umas que já ajudarão seu inicio nesse caminhar rumo a programação VBA. O CORRETO É VOCÊ FAZER UM CURSO ESPECÍFICO DE VBA!

Segue a resolução abaixo:

```

Sub TESTEMACRO1()
'
' TESTEMACRO1 Macro
' Macro que gera o resultado da PAUTA.
'
'----> CÓDIGO PARA GERAR MÉDIA

i = 3
Do While Worksheets("PAUTA").Cells(i, 1) <> ""
    Worksheets("PAUTA").Cells(i, 4) = (Worksheets("PAUTA").Cells(i, 2) + Worksheets("PAUTA").Cells(i, 3)) / 2
    i = i + 1
Loop
End Sub
    
```

Entenda o que esse OBJETO faz, e isso ajudará a desenvolver parte de sua lógica.

Vimos acima um OBJETO chamado **WORKSHEETS ("PAUTA").CELLS(i, 1)**. Esse objeto cria uma relação entre a PLANILHA que esta sendo usada **PAUTA**, e as linhas dessa planilha.

Observe que: **WORKSHEETS (PLAN)** → Representa a planilha que o Excel deve usar (PLAN).
CELLS (i, 1) → Representa que o Excel deve iniciar seu trabalho a linha i (no nosso caso 3) e da coluna 1 (escrito ao lado). Note que "i" é uma variável que começa recebendo 3 pois é a linha que se encontra o primeiro aluno.

Pronto!

Entendendo isso, agora você deve praticar a sua lógica.

→ **VEJA A EXPLICAÇÃO DO CÓDIGO ACIMA**

Do While Worksheets(plan).Cells(i, 1) <> "" → Diz ao Excel para ir na planilha chamada "PAUTA", pegar a célula da linha 3 (pois i vale 3) e da coluna 1 e perguntar se o que tem dentro dela é diferente (<>) de vazio. Se isso for verdadeiro, o Excel vai para a próxima instrução, que é:

$$\underbrace{\text{Worksheets(plan).Cells(i, 4)}}_{D3} = \underbrace{(\text{Worksheets(plan).Cells(i, 2)} + \text{Worksheets(plan).Cells(i, 3)})}_{(B3 + C3)} / 2$$

Então o que esta acontecendo, é o cálculo da média. A primeira célula da coluna média **D3**, recebe a soma da Nota 1 (**B3**) + Nota 2 (**C3**) e divide por 2. O resultado fica armazenado em **D3**.

Depois que isso é feito, o Excel vai para $i = i + 1$, conforme escrito no código acima. Isso ocorre para que feita para o aluno da linha 3 (aluno de nome A), o ponteiro do Excel passa para o aluno B na linha 4 ($i = i + 1$, como i valia 3, o novo i agora é 4)... e assim vai sendo feito para todos os alunos.

→ A PERGUNTA É: ENTÃO QUANDO A MACRO PARA DE EXECUTAR?

Ela vai para quando o `Worksheets("PAUTA").Cells(i, 1)` for vazia. Assim, quando o ponteiro do Excel chegar depois do último aluno (aluno de nome L, célula **A13**), ele perguntará se é igual a vazio. Como a célula **A14** não tem nada dentro, logo o programa finaliza.

→ EXECUTE A MACRO

Para executar a macro que você criou, basta ir em:

EXIBIÇÃO → MACRO → EXIBIR MACROS → SELECIONE "TESTESMACRO1" → EXECUTAR

PRONTO! Veja o resultado final.

	A	B	C	D	E
1	PAUTA				
2	ALUNO	NOTA 1	NOTA 2	MÉDIA	SITUAÇÃO
3	A	9,0	10,0	9,5	
4	B	5,0	7,0	6	
5	C	6,0	8,0	7	
6	D	9,0	6,0	7,5	
7	E	8,0	7,0	7,5	
8	F	3,0	9,0	6	
9	G	4,0	6,0	5	
10	H	4,0	5,0	4,5	
11	I	6,0	8,0	7	
12	J	9,0	8,0	8,5	
13	L	5,0	8,0	6,5	

CUIDADO!

Pratique resolvendo o campo **SITUAÇÃO** do aluno acima. Provavelmente, você terá dúvidas uma vez que não conhece todas as instruções da linguagem. Mas peça ao professor para resolver juntamente com você assim ele te explicará novas instruções.

Ex: IF, ELSE, DO WHILE, WHILE...

Tente trabalhar usando o objeto **WORKSHEETS().CELLS()**. Ele ajuda na resolução de problemas pois trabalha fazendo a relação entre a planilha e as células dela.

13. MACROS DE INTERAÇÃO COM O USUÁRIO

Essas macros são muito usadas quando queremos realizar uma interação com o usuário, ou seja, precisamos saber informações que somente eles tenham para que possamos prosseguir com a execução da automatização. Veja os casos a seguir.

Caso 1: Suponha que você queira fazer uma brincadeira com seu amigo. Então você cria uma macro que faz perguntar para que seu amigo responda. Se ele responder corretamente aparecerá uma mensagem informando que ele acertou, caso contrário, Aparece a mensagem informando que ele errou.

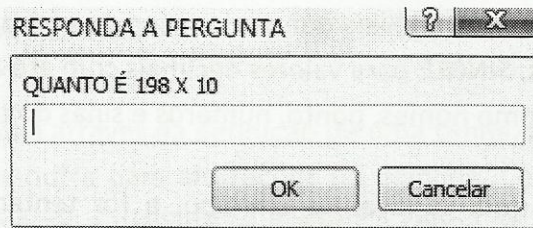
1º Passo → Crie a macro e coloque o nome de "RESPONDA". Caso não lembre como criar, consulte o capítulo de macro codificada.

2º Passo → Abra o Visual Basic usando **ALT + F11** e digite o código abaixo.

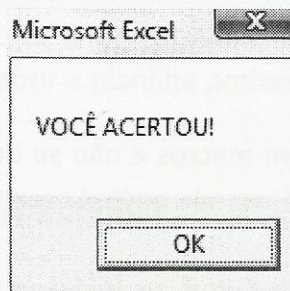
```
resposta = Application.InputBox("QUANTO É 198 X 10", "RESPONDA A PERGUNTA")
  If resposta = 1980 Then
    MsgBox ("VOCÊ ACERTOU!")
  Else
    MsgBox ("QUE PENA! VOCÊ ERROU")
  End If
End Sub
```

The diagram shows two arrows pointing downwards from the two MsgBox calls in the code above. The left arrow points to the text "Texto 1" and the right arrow points to the text "Texto 2".

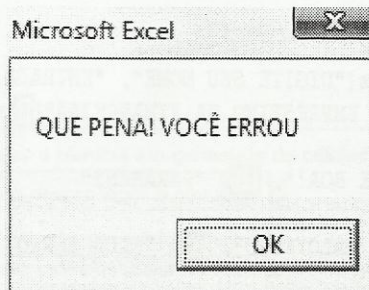
3º Passo → O **Application.InputBox**, é o responsável por criar a caixa de diálogo abaixo. Nota que é passada dois parâmetros, o texto 1 e o texto 2 mostrado acima.



4º Passo → Respondendo corretamente, aparecerá a seguinte mensagem. Essa mensagem foi criada pelo MSGBOX, onde foi passado somente como parâmetro de exibição a mensagem “Você Acertou”.



5º Passo → Respondendo e forma incorreta, aparecerá a seguinte mensagem. Essa mensagem também foi criada pelo MSGBOX, onde foi passado somente como parâmetro de exibição a mensagem “QUE PENA! VOCÊ ERROU”.



CONCLUSÃO:

Note que resposta é uma variável que recebe o que for respondido pelo usuário no momento da pergunta. A pergunta em si é gerada pelo `Application.InputBox` e as respostas de certo ou errado geradas pelo `MSGBOX`. Então, vimos aqui que é possível criar macros com interação com o usuário.

13.1 UTILIZANDO VARIÁVEIS DOTIPO STRING

Os procedimentos e funções criadas no Excel poderão necessitar de valores que venham a ser fornecidos durante a execução do programa (Macro). Quando temos essa necessidade,

precisamos trabalhar com variáveis que podem ser de vários tipos, como DATE para datas, INTEGER para inteiros, SINGLE para valores decimais com até 8 casas e também STRING para valores alfabéticos (Como nomes, ponto, números e sinas do teclado).

Caso seja declarada uma variável de um tipo, e for tentado armazenar nesta um valor diferente do estabelecido, será retornada uma mensagem de erro. O Visual Basic possui a característica de estabelecer um caracter para ser utilizado na declaração do tipo, no final do nome da variável em uso. Por exemplo, uma variável para receber um nome, poderá ser indicada como "NOME\$", onde o caracter "\$" estabelece que a variável NOME é do tipo string.

IMPORTANTE:

Ao se fazer uso de variáveis em macros e não se definir o tipo das mesmas, o Visual Basic automaticamente estabelecerá que ela poderá receber qualquer tipo de dados.

E vale lembrar que ao se declarar uma variável com um determinado nome, quando for usá-la você não pode digitar o nome da mesma de forma diferente. O Excel entenderá que você esta tentando criar uma nova variável ou tentando usar algo que não foi declarado.

Vamos a um exemplo prático:

Agora você já possui capacidade de criar suas macros, crie essa macro abaixo e teste seu resultado.

```
nome$ = Application.InputBox("DIGITE SEU NOME", "ENTRADA DE DADOS", , , , , 2)
resp% = MsgBox("VOCÊ FEZ UM EMPRÉSTIMO NA FINANCEIRA" & nome$ & Chr(13) & "VAI PAGAR", 4)
If resp% = vbYes Then
    MsgBox "VOCÊ É GENTE BOA!", 48, "PARABÉNS"
Else
    MsgBox "VOCÊ PARECE CALOTEIRO", 16, "FALA SÉRIO"
End If
End Sub
```


→ ENTENDENDO SEU SIGNIFICADO:


nome\$ - É uma variável criada para receber uma resposta do usuário.

resp% - É uma variável criada para receber uma resposta do usuário.

& - É o que chamamos de concatenar. É um elemento que liga o nome da pessoa a mensagem

4 - Mostra na caixa de diálogos os botões de SIM ou NÃO

48 - Mostra o ícone de Atenção ()

16 - Mostra o ícone de X ()

Chr(13) - É responsável pela quebra de linhas.