

**PROGRAMAÇÃO**

**EM**

**VBA**

# **PRÉ-TESTE DE EXCEL AVANÇADO**

01) No Excel, quando uma célula contém a sequência de caracteres "####", isto indica que:

- a) A fórmula utilizada para gerar o valor da célula contém um erro genérico.
- b) A operação utilizada contém divisão por um valor nulo.
- c) Um endereço de célula na fórmula faz referência a uma célula inexistente.
- d) A largura da célula é menor que o número de dígitos do valor nela contido.

02) O que é uma pasta de trabalho do Excel ?

- a) Um arquivo temporário.
- b) Uma parte de uma planilha.
- c) Um intervalo de células.
- d) Um conjunto de planilhas.

03) No Excel, uma lista ou banco de dados **NÃO** pode ser definido por:

- a) Um conjunto de linhas e colunas apenas com dados numéricos.
- b) Um conjunto de slides importado do Microsoft PowerPoint.
- c) Um conjunto de linhas e colunas com qualquer tipo de dados.
- d) Um banco de dados importado do Microsoft Access.

04) No Excel, o que são, respectivamente, endereços relativos e absolutos ?

- a) São endereços de células que variam ao serem copiados ou movidos para outras células e endereços de células que se mantêm fixos ao serem copiados ou movidos para outras células.
- b) São endereços de células em uma planilha e endereços de fórmulas.
- c) São endereços de células que se mantêm fixos ao serem copiados ou movidos para outras células e endereços de células que variam ao serem copiados ou movidos para outras células.
- d) São endereços de células em relação à planilha onde foram definidos e endereços de células em relação à pasta de trabalho onde foram definidos.

05) De um modo geral podemos dizer que a diferença entre os comandos Editar-Copiar e Editar-Recortar é:

- a) Editar-Copiar é tirar o texto de um lugar e colar em outro local.
- b) Editar-Recortar é tirar o texto de um lugar e colar em outro local.
- c) Editar-Copiar e Editar-Recortar tem a mesma função de mover o texto de um local para o outro.
- d) Editar-Copiar e Editar-Recortar tem a mesma função de copiar o texto de um local para o outro.

06) O que são funções aninhadas no Excel ?

- a) São funções cujos argumentos são formados por outras funções.
- b) São funções que contém mais de um argumento.
- c) São funções lógicas (SE, OU, E, etc) cujos argumentos são formados por outras funções lógicas.
- d) São funções vinculadas a funções de outras planilhas.

07) Considerando que você está trabalhando com o Excel, indique a afirmação ERRADA:

- a) Os textos são alinhados normalmente à esquerda das células, a menos que sejam formatados de outra forma.
- b) As fórmulas devem sempre ser iniciadas com o símbolo = (sinal de igual), com exceção das funções que não precisam do uso desse caractere.
- c) Os números são alinhados normalmente à direita das células, a menos que sejam formatados de outra forma.
- d) Tanto os textos como números podem ser alinhados no centro das células.

08) No Windows 95 o Windows Explorer é um

- a) dispositivo de edição de textos que permite explorar as vantagens do Word e do Excel
- b) *software* destinado a navegar pelos endereços da Internet

- c) dispositivo conversor de programas fonte em programas executáveis
- d) dispositivo para verificar a existência de arquivos infectados por vírus de computador
- e) dispositivo que permite a visualização de todos os arquivos em seu computador

09) Considerando que você está editando um texto no MS-Excel 97, versão português, assinale a opção que indica o resultado correto para o comando mencionado.

- a) Ctrl+T faz a transferência do texto para um arquivo a ser especificado.
- b) Alt+A exibe o menu de *Ajuda*.
- c) Alt+E abre o menu de *Exibir*.
- d) Ctrl+Z desfaz a última digitação.
- e) Ctrl+I abre a janela *Imprimir*.

10) Em relação ao MS Excel é correto afirmar que

- a) não pode haver relação entre o valor de uma célula e os valores de outras células
- b) uma fórmula de uma planilha pode fazer referência a uma célula de outra planilha
- c) todas as colunas de uma mesma planilha devem ter a mesma largura
- d) os comandos Dividir e Congelar Painéis têm a mesma função
- e) uma ou mais colunas ocultas são transferidas para um arquivo auxiliar devendo este ser acessado para a sua reexibição

# Teclas de ATALHO

	<i>Tecla de Função</i>	<i>SHIFT</i>	<i>CTRL</i>	<i>ALT</i>	<i>CTRL+ SHIFT</i>	<i>ALT+ SHIFT</i>
<b>F1</b>	Ajuda	O que é isto ? (Explica o que um comando ou ícone faz)		Inserir uma nova folha de gráfico		Inserir uma nova planilha
<b>F2</b>	Edita a célula ativa	Inserir ou Editar um comentário de célula			Salvar Como	Salvar
<b>F3</b>	Cola um nome em uma fórmula	Colar função	Define um nome para uma célula ou região		Cria nomes usando rótulos de linha e de coluna	
<b>F4</b>	Repete a última ação	Localizar próxima (Repete o último Localizar)	Fechar (Fecha a janela aberta)	Sair (Fecha o Programa aberto)		
<b>F5</b>	Ir para (seleciona uma célula ou região)	Localizar (Exibe a caixa de diálogo Localizar)	Restaurar o tamanho da janela			
<b>F6</b>	Mover-se até o próximo painel	Mover-se até o painel anterior	Mover-se até a próxima janela de pasta de trabalho		Mover-se até a janela anterior da pasta de trabalho	
<b>F7</b>	Corretor Ortográfico (Corrigir erros de Ortografia)		Mover a janela			
<b>F8</b>	Estende uma seleção	Adiciona à seleção	Redimensiona a janela	Mostra as MACROS existentes		
<b>F9</b>	Recalcula as fórmulas das células das planilhas abertas	Calcula a planilha ativa	Minimiza a pasta de trabalho			
<b>F10</b>	MENU (Torna ativa a barra de menu)	Equivaler ao BOTÃO DIREITO do mouse	Maximiza ou restaura a janela da pasta			
<b>F11</b>	Cria um GRÁFICO INSTANTÂNEO	Inserir uma NOVA PLANILHA	Inserir uma FOLHA DE MACRO do Excel 4.0	VBA (Exibe o Editor do VBA)		
<b>F12</b>	Salvar como	Salvar	Abrir Arquivo		Imprimir	

# **OUTRAS TECLAS DE ATALHO**

ALT+A = Abre o Menu **Arquivo**  
ALT+D = Abre o Menu **Dados**  
ALT+E = Abre o Menu **Editar**  
ALT+F = Abre o Menu **Formatar**  
ALT+I = Abre o Menu **Inserir**  
ALT+J = Abre o Menu **Janela**  
ALT+M = Abre o Menu **Ferramentas**  
ALT+U = Abre o Menu **Ajuda**  
ALT+X = Abre o menu **Exibir**

CTRL+A = Abrir Arquivo  
CTRL+B = Salvar  
CTRL+C = Copiar  
CTRL+D = **Copia** o conteúdo de cima para a célula atual  
CTRL+F = **Copia** o conteúdo de cima para a célula atual permitindo a sua edição  
CTRL+I = Formata para **Itálico**  
CTRL+J = **Alterna** entre o resultado e a fórmula  
CTRL+K = Insere **Hyperlink**  
CTRL+L = Localizar  
CTRL+N = Formata para **Negrito**  
CTRL+O = Abre uma **Nova Pasta** de Trabalho  
CTRL+P = Imprimir  
CTRL+T = **Seleciona** toda a Planilha

Se tiver digitado "= nome de uma função", abre o assistente da função

CTRL+U = Substituir  
CTRL+V = **COLAR** o conteúdo da memória na célula ativa  
CTRL+X = Recortar  
CTRL+Y = **Vai para** uma célula ou região específica da Planilha (equivalente a **F5**)  
CTRL+Z = Desfazer  
CTRL+W = **Fechar** a janela da pasta de trabalho ativa (equivalente a **CTRL+F4**)  
CTRL+1 = Formatar Células  
CTRL+2 = Negrito  
CTRL+3 = Itálico  
CTRL+4 = Sublinhado  
CTRL+5 = Riscado  
CTRL+9 = Oculta as linhas atualmente ativas  
CTRL+PAGE DOWN = Vai para a próxima Planilha  
CTRL+PAGE UP = Vai para a Planilha Anterior  
CTRL+HOME = Vai para a célula A1  
CTRL+END = Vai para a última célula escrita na planilha  
CTRL+TAB = Anda pelas guias dos menus  
CTRL+[ = Marca todas as células que originam a fórmula da célula atual (antecedentes)  
CTRL+] = Marca todas as células que usam a célula atual em suas fórmulas (precedentes)  
CTRL+\* = Seleciona todas as células preenchidas, adjacentes à célula atual

# **TABELA DINÂMICA**

- Tabela dinâmica é uma tabela interativa que resume uma grande quantidade de dados rapidamente ou os combina de tabelas diferentes.
- É possível girar suas linhas e colunas para ver resumos diferentes dos dados de origem, filtrar os dados exibindo páginas diferentes ou exibir os detalhes das áreas de interesse.
- Você pode criar uma Tabela dinâmica a partir de uma lista ou banco de dados do **EXCEL**, de um banco de dados externo (Access, por exemplo), de várias planilhas do **EXCEL** ou de outra Tabela dinâmica.
- A Tabela dinâmica resume dados usando uma função de resumo que você especifica, como: **Soma**, **Contagem** ou **Média**. Você pode incluir subtotais e totais gerais automaticamente, ou usar suas próprias fórmulas adicionando campos e itens calculados.
- Para criar uma Tabela dinâmica, use o **Assistente da tabela dinâmica** como guia para localizar e organizar os dados que você deseja analisar.

## **CRIANDO UMA TABELA DINÂMICA DO EXCEL**

1. Abra a pasta de trabalho onde você deseja criar a Tabela dinâmica. Se você estiver baseando a Tabela dinâmica em uma **lista** ou **banco de dados** do **EXCEL**, certifique-se de estar numa célula da **lista** ou do **banco de dados**.
2. Clique em **Dados**, na barra de menus.
3. Clique em Relatório da tabela dinâmica.

## **EXCLUINDO UMA TABELA DINÂMICA**

1. No menu **Tabela dinâmica** da barra de ferramentas **Tabela dinâmica**, aponte para **Selecionar** e, em seguida, certifique-se de que **Ativar seleção** esteja pressionado.
  2. Clique em uma célula na **Tabela dinâmica** que você deseja excluir.
  3. Na barra de ferramentas **Tabela dinâmica**, aponte para **Selecionar** no menu **Tabela dinâmica** e, em seguida, clique em **Tabela inteira**.
  4. No menu **Editar**, aponte para **Limpar** e, em seguida, clique em **Tudo**.
- Observação: Quando você exclui uma Tabela dinâmica os dados de origem não são afetados.

## **USANDO TOTAIS**

- Você pode incluir totais gerais para os dados contidos em linhas e colunas da Tabela dinâmica.
- O **EXCEL** exibe automaticamente os subtotais para o campo de linha ou coluna mais externo, quando você cria dois ou mais campos de linha ou de coluna em uma Tabela dinâmica. Para os campos de linha ou de coluna mais internos, o **EXCEL** exibe subtotais somente se você somá-los. Você pode especificar a função de resumo a ser usada para os subtotais.

## **FORMATANDO UMA TABELA DINÂMICA**

- Para manter as alterações da formatação quando você atualizar ou modificar o layout em uma **Tabela dinâmica**, você deve apontar para **Selecionar** no menu **Tabela dinâmica** da barra de ferramentas **Tabela dinâmica** e, em seguida, certifique-se de que **Ativar seleção** esteja pressionado antes de selecionar os dados que deseja formatar.
- 1. Selecione a parte da Tabela dinâmica que você deseja formatar.
- 2. Para aplicar as alterações de formatação desejadas, use os botões na barra de ferramentas **Formatação** e os comandos do menu **Formatar**.
- 3. Para definir opções que afetem a formatação da Tabela dinâmica, clique em uma célula na Tabela dinâmica, clique em **Opções** no menu **Tabela dinâmica** da barra de ferramentas **Tabela dinâmica** e selecione as opções desejadas.
- Para obter **Ajuda** sobre uma opção, clique no **ponto de interrogação** na barra de ferramentas e, em seguida, clique na opção.
- Para formatar uma Tabela dinâmica automaticamente, certifique-se de selecionar a opção **AutoFormatação de tabela** em **Opções de formatação**. Toda formatação manual que você aplicar à Tabela dinâmica, antes ou depois de for-

matar automaticamente, terá prioridade sobre qualquer formatação aplicada com o comando AutoFormatação.

- Para remover toda a formatação de uma Tabela dinâmica, desmarque a caixa de seleção AutoFormatação de tabela, selecione a Tabela dinâmica inteira e aplique o estilo Normal.
- Para remover a formatação sempre que a Tabela dinâmica for atualizada, desmarque a caixa de seleção **Preservar formatação**.
- Para aplicar a formatação a todas as páginas de uma Tabela dinâmica com campos de página, selecione Tudo para cada campo de página e, em seguida, aplique a formatação. Se a Tabela dinâmica possuir campos de página que consultam dados externos à medida que você seleciona cada item, exiba cada item de campo de página separadamente e, em seguida, aplique a formatação.
- Você não pode aplicar formato condicional nem configurar uma validação de dados para as células em uma Tabela dinâmica.
- O **EXCEL** não mantém as alterações introduzidas em bordas de células quando você altera o layout ou atualiza uma Tabela dinâmica.
- Para usar células mescladas para todos os rótulos de coluna ou de linha mais externos na Tabela dinâmica, clique em Opções no menu Tabela dinâmica da barra de comandos Tabela dinâmica e marque a caixa de seleção Mesclar rótulos.
- Antes de formatar uma Tabela dinâmica em um arquivo criado em uma versão anterior do **EXCEL**, salve a pasta de trabalho como um arquivo do **EXCEL**. No menu **Arquivo**, clique em **Salvar como** e em **Pasta de trabalho do EXCEL (\*.xls)** na caixa **Salvar como tipo**.

## EXERCÍCIOS PROPOSTOS

- 01) Abrir a pasta **TabelaDinamica.xls** que está no diretório **C:\Meus Documentos\01-ApoioAluno** e depois salvar na sua pasta de trabalho.
- 02) Criar uma Tabela Dinâmica que mostre o resumo dos lançamentos financeiros do mês de Setembro do Banco Clínica. Esta tabela dinâmica deverá ter os seguintes parâmetros: Em **Página** coloque as **Agências** / Em **Coluna** coloque os **Clientes** / Em **Linha** coloque os **Tipos de Conta** / Em **Dados** coloque a **Soma de Valores** / Salve a tabela dinâmica numa nova planilha e depois renomeie para **Tabela Dinâmica Setembro**. Coloque esta nova planilha após a planilha **Banco Clínica**.
- 03) Formate os valores da Tabela Dinâmica clicando no ícone **Separador de Milhares**.
- 04) Formate os títulos das colunas clicando no botão **Cliente** e em **Negrito**, **Itálico**, **Centralizado** e **Fonte Vermelha**.
- 05) Formate os títulos das linhas clicando no botão **Tipo de Conta** e em **Negrito**, **Itálico** e **Fonte Vermelha**.
- 06) Observando a Tabela Dinâmica recém-criada responda às seguintes perguntas:
  - a) Qual o total de aplicações em Caderneta de Poupança de todas as agências ? \_\_\_\_\_  
(Resp.: R\$ 638.956,00)
  - b) Qual o total aplicado em C/C por Clientes Novos da Agência Central ? \_\_\_\_\_  
(Resp.: R\$ 49.228,00)
  - c) Na Agência Zona Norte, qual o total aplicado por clientes antigos a Longo Prazo? \_\_\_\_\_  
(Resp.: R\$ 125.374,00)
- 07) Criar um nova planilha que mostre todos os lançamentos da **Agência Zona Oeste**, em aplicações a **Longo Prazo** de clientes **Antigos** e que somaram R\$ 10.000,00 (são 5 lançamentos). Renomear depois a planilha para **ListaLongoPrazoOeste**.

- 08) Vá até a planilha **Banco Clínica** e altere o primeiro lançamento do dia 01/09 de R\$ 340,00 para R\$ 86.027,00. Volte para a planilha **Tabela Dinamica Setembro** e faça com que o **Excel** atualize os valores (estando em qualquer célula da Tabela Dinamica, basta clicar no ícone **Atualizar Dados** ( ! )). Observe que o valor total de **C/C**, Clientes **Antigos** de todas as Agências passou de R\$ 314.313,00 para R\$ 400.000,00.
- 09) Fazer uma nova Tabela Dinâmica, baseada na tabela dinâmica já feita (para economizar memória), com **Agência** na **Linha** / **Cliente** na **Página** / **Tipo de Conta** na **Coluna** / **Contagem de Valores** em **Dados** / Observação: para mudar a **Soma de Valores** para **Contagem de Valores**, basta dar um clique duplo no botão **Soma de Valores** em **Dados** / Salve esta nova Tabela Dinâmica na célula **F1** da mesma **Tabela Dinamica Setembro** / Confirme se realmente são 5 lançamentos de Clientes Antigos na Agência Zona Oeste em Aplicações a Longo Prazo.
- 10) Usando a 2a. Tabela Dinâmica (feita na questão anterior), faça aparecer todos os 10 lançamentos de Clientes **Novos** da Agência **Central** das Aplicações a **Curto Prazo**. Salve a nova planilha como **Lista-CurtoPrazoCentral**.



# PROGRAMAÇÃO

- Um Programa é um conjunto de comandos com uma determinada sequência lógica e que permite o processamento de determinada ação, proporcionando a realização de uma tarefa. É através do Programa que se processa a interface entre o usuário e o computador.
- Sem o programa o computador é considerado uma máquina "burra", ou seja, o maquinário (hardware) existe mas precisa Ter um "cérebro" que o faça agir em determinadas situações.
- O Programa Word, por exemplo, é um conjunto de comandos que transforma o seu computador numa potente máquina de escrever; já o programa Excel, converte o mesmo computador numa espetacular planilha de cálculos; e assim por diante.
- Quem cria o Programa é o **Programador**.

## **SISTEMA**

- É um conjunto de programas que realizam uma ou várias tarefas pré-determinadas.
- O profissional que cuida dos Sistemas é o **Analista de Sistema**.

## **LINGUAGENS DE PROGRAMAÇÃO**

- Existem várias linguagens de programação: Visual Basic, Delphi, Turbo Pascal, Fox Pro e outras. São elas que interpretam o que falamos e convertem (traduzem) da nossa linguagem (humana) para a linguagem do computador (de máquina).

## **LÓGICA DE PROGRAMAÇÃO**

- É um conjunto de comandos que executam determinadas ações, permitindo ao computador que se transforme numa máquina eficiente.
- Quando damos um determinado comando no computador, em linguagem humana, é a linguagem de programação que traduz para o código de máquina, normalmente códigos numéricos, para que aquele comando seja executado.

## **SEQUÊNCIA**

- Um programa é uma sequência lógica de comandos que executam determinadas ações.
- Esta sequência tem que ser lógica, pois senão o programa não executará as ações esperadas. Exemplo: se fizermos um programa que irá calcular o número de dias já vividos por uma pessoa, **primeiro** temos que perguntar a data do seu nascimento, para **depois** calcular o número de dias.
- Quando você vai fazer uma viagem de carro, você toma atitudes como (a) fazer uma revisão no automóvel; (b) abastecer; (c) ver os recursos financeiros; (d) viajar. Isto é uma sequência lógica.
- Na informática, quando queremos elaborar um programa qualquer, precisamos determinar a sequência lógica das ações e, depois, passar estas informações para o computador. A lógica de programação é a união da sequência com a linguagem.
- As sequências não são, necessariamente, iguais. Cada pessoa pode agir de maneira diferente para fazer uma mesma ação. Por exemplo: para almoçar você poderia seguir a seguinte sequência:
  - a) Lavar as mãos;
  - b) Pegar o prato;
  - c) Pegar os talheres;
  - d) Servir a comida;
  - e) Comer;
  - f) Colocar o prato usado na pia;
  - g) Escovar os dentes
- Algumas pessoas podem ter uma sequência diferente para executar esta mesma ação. Por exemplo: alguém pode pegar os talheres (c) antes de pegar o prato (b). Mas o programa será executado da mesma forma. Assim é, também, com um programa.



# **VBA NO EXCEL**

## **PLATAFORMA DE TRABALHO**

- Usaremos como plataforma para programar em VBA o aplicativo Excel.
- Por se tratar de um software mais popular, todos os exemplos poderão ser aplicados na Planilha Eletrônica da Microsoft.
- Mas até o final do Curso também faremos alguns exemplos para uso no Access e no Word.

## **COMO CHEGAR NO VBA ATRAVÉS DO EXCEL**

- Abra o Excel;
- Clique em FERRAMENTAS → Macro → Editor do Visual Basic
- O atalho pelo teclado é ALT-F11.

## **MÓDULOS**

- Mande inserir um Módulo (Inserir → Módulo).
- Os módulos são como o Editor de Textos Word. São neles que digitamos a linguagem de programação (VBA) e que será depois compilada na hora da execução.
- Você pode fazer um programa em cada Módulo ou ter um Módulo só para as Macros, outro para as Funções e outro para as macros de teclado, por exemplo.

# **FUNÇÕES PERSONALIZADAS**

## **CRIANDO NOSSAS PRÓPRIAS FUNÇÕES**

- Você pode criar funções personalizadas no EXCEL usando a linguagem VISUAL BASIC.
- Para acessar o módulo de programação da linguagem VBA pressione Ferramentas → Macro → Editor do Visual Basic (ou o atalho pelo teclado ALT+F11).
- Para fechar o módulo VBA e voltar para a planilha, pressione ALT+Q.
- Para alternar entre o módulo VBA e a planilha, clique no rodapé dos arquivos abertos ou pressione ALT+TAB ou ainda ALT+F11.

## **CRIANDO A 1ª FUNÇÃO**

- Vamos criar uma função chamada RAIZC, que irá calcular a raiz cúbica do número digitado.
- Pressione ALT+F11 para abrir o VBA.
- Clique em INSERIR → MÓDULO.
- O VBA irá introduzir uma folha de módulo vazia na tela. Para uma melhor visualização, maximize a tela.
- Para iniciar uma função personalizada com parâmetros, digite FUNCTION, que é uma palavra interna do VBA, seguido do NOME DA FUNÇÃO e do nome das variáveis entre parênteses.
- O nome da função e os nomes das variáveis não poderão conter espaços, não poderão ser nome.
- Para este exemplo, digite os dados a seguir. Excel não faz distinção entre letras maiúsculas e minúsculas.

```
FUNCTION RAIZC (RADICANDO)
```

- Quando você pressionar <ENTER> Excel irá finalizar a função com o comando END FUNCTION, ficando

```
Function RAIZC (RADICANDO)  
I  
End Function
```

- Na parte interna da função, endente o parágrafo, pressionando <TAB> e digite a expressão que irá calcular a raiz cúbica do número digitado, ou seja,

```
RAIZC = RADICANDO ^ (1/3)
```

- A sua função vai ficar com o seguinte aspecto:

```
Function RAIZC (RADICANDO)  
RAIZC = RADICANDO ^ (1/3)  
End Function
```

- Observe o jogo de cores do VB. Azul é para nomes internos; preto é para nomes definidos por nós, usuários; verde é para comentários, valores que não serão criticados pelo VB; e, finalmente, vermelho é para erros encontrados.
- Se você quiser fazer um comentário à medida que digita, use o apóstrofe. VB não lê nada após este sinal. Experimente colocar os seguintes comentários na sua função:

```
Function RAIZC(RADICANDO) 'nome da função  
RAIZC = RADICANDO ^ (1 / 3) 'cria a fórmula  
End Function 'encerra a função
```

- Para testar a sua função e verificar se ela está funcionando perfeitamente, volte à planilha de trabalho clicando com o mouse na barra de status.
- Pegue uma célula vazia e digite

```
=RAIZC(64)
```

- Excel irá responder **4** como resposta após você pressionar <ENTER>.
- Você também pode usar o assistente de função. Vamos calcular a raiz cúbica de 27 usando o assistente de função.
- Vá até uma célula vazia e pressione f(x).

- Escolha a categoria DEFINIDA PELO USUÁRIO (a última) e a função RAIZC que acabamos de definir e depois OK.
- Digite 27 para a radicando. Observe que a resposta (3) já aparece no rodapé da função. Esta resposta será transferida para a planilha quando você pressionar <ENTER> ou clicar em OK.

## **CRIANDO UMA 2ª FUNÇÃO**

- Agora vamos criar uma função com 2 variáveis, chamada RAIZN, que irá calcular a raiz n-ésima de qualquer número.
- Volte para o módulo clicando no módulo (barra de status).
- Digite a seguinte função, após a função já digitada RAIZC. Observe que não precisamos inserir 1 módulo para cada função. Numa mesma folha de módulo podemos criar

```
Function RAIZN (RADICANDO , INDICE)  
    RAIZN = RADICANDO ^ (1/INDICE)  
End Function
```

- Experimente a função que você criou voltando para a planilha e calculando a raiz 10ª de 1024 (a resposta é 2).

## **FUNÇÕES CONDICIONADAS**

- Para criar uma função condicionada, podemos usar o comando IF ... END IF, equivalente à função SE do assistente de função do Excel.
- Vamos criar uma função chamada IIRR, que analisa um salário e devolve a alíquota que a pessoa irá pagar, segundo os seguintes critérios: quem ganha até R\$ 600 está isento; quem ganha acima disto paga 10% de imposto.
- Digite os seguintes dados:

```
Function IIRR (SALARIO)  
    If SALARIO <= 600 Then  
        IIRR = "ISENTO"  
    Else  
        IIRR = "10%"  
    End If  
End Function
```

- Faça um teste digitando, na planilha, o salário de R\$ 200 e R\$ 900. As respostas serão **ISENTO** e **10%** respectivamente.

## **FUNÇÕES COM OPERADORES MATEMÁTICOS**

- Vamos criar uma função que calcule os juros de uma prestação em atraso.
- Digite no módulo uma função com o nome de MULTA com as seguintes características:

```
Function MULTA (VALOR, ALIQUOTA)  
    MULTA = VALOR * ALIQUOTA  
End Function
```

- Faça um teste com o valor de R\$ 2.300 e a alíquota de 17,3%. Você deverá encontrar uma multa de R\$ 397,90.

## **EXERCÍCIOS PROPOSTOS**

- 01) Criar uma função no VB, denominada POT que pedirá 2 parâmetros (sugestões: BASE e EXPOENTE) e irá calcular qualquer potência.
- 02) Faça uma função que calcule e devolva a faixa de salário em que a pessoa se encontra. Para quem ganha até R\$ 1.000 enquadrar na Faixa 1; quem ganha acima, enquadrar na Faixa 2.
- 03) Crie uma função chamada IR que calcule o IR a ser pago por um contribuinte. Se a base de cálculo dele for menor ou igual a R\$ 900, ele estará ISENTO; se ganhar mais de R\$ 900 e menos de R\$ 1.800, pagará 15% e deduzirá R\$ 135; se ganhar acima de R\$ 1.800, pagará 27,5% e deduzirá R\$ 360. A função já deverá devolver o valor a ser pa-

go de imposto.

- 04) Um condomínio pediu a você que elaborasse uma função (sugestão: COND) para a planilha Excel que irá calcular o montante a ser pago por um condômino que está em atraso. As variáveis serão: VALOR, TAXA\_MÊS, DIAS\_ATRASO e MULTA). A multa incide apenas sobre o principal, assim como os juros (ou seja, você joga os juros sobre a multa).
- 05) Criar uma função que calcule os juros simples (sugestão: JS) de uma aplicação, fornecido o capital, a taxa mensal, o prazo em dias e um COD. Esta função devolverá os Juros Simples se o COD for 1 ou o Montante Simples se o COD for 2 ou uma mensagem de erro (COD ERRADO) se o valor de COD for diferente de 1 ou de 2.

## GABARITO

01)

```
Function POT(BASE, EXPOENTE)
    POT = BASE ^ EXPOENTE
End Function
```

02)

```
Function FAIXA (SALARIO)
    If SALARIO <= 1000 Then
        FAIXA = "FAIXA 1"
    Else
        FAIXA = "FAIXA 2"
    End If
End Function
```

03)

```
Function IR(SALARIO)
    If SALARIO <= 900 Then
        IR = "ISENTO"
    Else
        If SALARIO <= 1800 Then
            IR = SALARIO * 0.15 - 135
        Else
            IR = SALARIO * 0.275 - 360
        End If
    End If
End Function
```

04)

```
Function COND(VALOR, TAXA_MES, DIAS_ATRASO, MULTA)
    COND = VALOR + VALOR * TAXA_MES * DIAS_ATRASO / 30 + VALOR * MULTA
End Function
```

05)

```
Function JS (CAPITAL, TAXA_MES, DIAS, COD)
    If COD = 1 Then
        JS = CAPITAL * TAXA_MES * DIAS / 30
    ElseIf COD = 2 Then
        JS = CAPITAL+CAPITAL * TAXA_MES * DIAS / 30
    Else
        JS = "COD ERRADO"
    End If
End Function
```

# **MACROS DE TECLADO**

## **LÓGICA DA MACRO DE TECLADO**

- O Excel permite que várias macros sejam criadas usando-se o sistema de gravação automática. A gravação é convertida para o código em VBA que pode ser depois criticado e alterado.
- A lógica da macro de teclado é a seguinte:
  - a) Defina o que você quer;
  - b) Execute as ações uma vez para treinar;
  - c) Coloque o Excel no modo de gravação;
  - d) Execute as ações mais uma vez, gravando;
  - e) Comande o Excel para parar de gravar;
  - f) Teste a macro para ver se ficou como desejado.
- As macros são convertidas em código VBA e podem ser analisadas, estudadas e modificadas.

## **EXEMPLO 1**

- Fazer uma macro de teclado que pegue o conteúdo de uma célula e faça as seguintes ações: negrite, centralize na célula, mude a fonte para ARIAL NARROW Tamanho 12 pts e com a cor vermelha. Salve-a como MAC\_FormataTexto.

# DECLARAÇÃO DE VARIÁVEIS

## VARIÁVEL

- Normalmente os programas que desenvolvemos faz uma interface com o usuário. Quando solicitamos alguns dados precisamos armazená-los em algum lugar da memória para uso posterior.
- Este lugar é chamado de **variável**.
- **Variável** é um local de armazenamento nomeado que pode conter dados, os quais podem ser modificados durante a execução do programa. Cada **variável** possui um nome que a identifica com exclusividade dentro de seu escopo. Um tipo de dados pode ser ou não especificado.
- Os nomes de variável devem começar com um caractere alfabético, devem ser exclusivos dentro do mesmo escopo, não podem ter mais de 255 caracteres e também não podem conter um ponto ou caractere de declaração de tipo incorporado.
- O VBA não exige que as variáveis sejam declaradas em seus programas, porém é uma segurança quanto ao uso errado da grafia de uma delas, quando optamos por tornar a declaração obrigatória (ao usarmos o comando Option Explicit).

## O COMANDO "OPTION EXPLICIT"

- É utilizada no nível de módulo (um conjunto de declarações seguidas de procedimentos) para forçar a declaração explícita de todas as variáveis desse módulo.
- Sua Sintaxe é: **Option Explicit** e ela deverá ser colocada na 1ª linha do módulo (antes de qualquer procedimento).
- Quando **Option Explicit** aparece em um módulo, você deve declarar explicitamente todas as variáveis utilizando as instruções **Dim**, **Private**, **Public**, **ReDim** ou **Static**. Se a declaração não for feita, VBA acusará um erro em tempo de compilação (período durante o qual o código-fonte é traduzido em código executável).
- Se você não utilizar a instrução **Option Explicit**, todas as variáveis não declaradas serão do tipo **Variant**, a menos que o tipo padrão seja especificado com uma instrução **Deftipo**.
- Recomendamos que você utilize **Option Explicit** para evitar a digitação incorreta do nome de uma variável existente ou para evitar confusão no código em que o escopo da variável não esteja claro.

## ESCOPO

- Define a visibilidade de uma variável, um procedimento ou objeto. Por exemplo, uma variável declarada como **Public** é visível a todos os procedimentos de todos os módulos em um projeto de referência direta, a menos que **Option Private Module** esteja ativo.
- Quando **Option Private Module** está ativo, o próprio módulo é privado e portanto não está visível a projetos de referência.
- As variáveis declaradas em um procedimento são visíveis somente dentro do procedimento e perdem seu valor entre chamadas, a menos que sejam declaradas **Static**.
- Exemplo de um programa com o comando Option Explicit:

Option explicit	'Força a declaração de variável explícita
Dim MinhaVar	'Declara a variável
MeuInt = 10	'Variável não declarada gera erro
MinhaVar = 10	'Variável declarada não gera erro

## DIM

- Declara variáveis e aloca espaço de armazenamento.



# **PRINCIPAIS COMANDOS**

## **ACTIVATE**

- Este método ativa a pasta de trabalho.
- Sintaxe:

```
Workbooks("MinhaPasta").Activate
```

## **CLEARCONTENTS**

- Limpa as fórmulas do intervalo. Limpa os dados de um gráfico mas deixa a formatação.
- Sintaxe:
- Onde expressão é (obrigatória) é uma expressão que retorne um objeto Chart ou Range.
- Exemplo:

```
Worksheets(1).Range("Regiao").ClearContents  
'Limpa o conteúdo do intervalo chamado "Regiao" da 1ª. planilha.
```

# **WORKSHEETS**

## **O COMANDO “WORKSHEETS”**

- É com este comando que você manipula as planilhas da sua pasta de trabalho.
- Pode-se abreviar escrevendo apenas Sheets.

## **EXERCÍCIOS**

- 01) Faça um programa que ative a planilha **Plan2** usando o comando **Activate** (que torna objetos ativos) e selecione as células **B5:C7** com o comando **Select**.
- 02) Fazer um programa que selecione as células **A1:C3** de **Plan1** e, em seguida, faça de **B2** a célula ativa.
- 03) Usando o comando **Add** faça com que seja inserida uma nova planilha na Pasta atual (Worksheets.Add).
- 04) Faça uma macro que se desloque para a 2ª Planilha.
- 05) Faça um programa que insira na célula atual a quantidade de planilhas da **Pasta de Planilhas Ativa**.
- 06) Faça um programa que insira uma planilha no final da pasta atual (Worksheets.Add.Move after:=Worksheets(Worksheets.Count)).

# ACTIVECELL

## O COMANDO "ACTIVECELL"

- Retorna um objeto Range representando a célula ativa da janela ativa (a janela visível) ou da janela especificada.
- Se a janela não estiver exibindo uma planilha, essa propriedade falhará.
- Somente para leitura.
- Quando você não especifica um qualificador de objeto, essa propriedade retorna a célula ativa da janela ativa.
- Tenha cuidado de distinguir entre célula ativa e seleção. A célula ativa é uma única célula dentro da seleção atual. A seleção pode conter mais de uma célula, mas somente uma é a célula ativa.
- Exemplo: Todas as expressões seguintes retornam a célula ativa, sendo todas equivalentes.

```
ActiveCell  
Application.ActiveCell  
ActiveWindow.ActiveCell  
Application.ActiveWindow.ActiveCell
```

## EXERCÍCIOS

- 01) Faça um programa que capture o conteúdo da célula atual e o devolva, na mesma célula, com a frase: "O conteúdo desta célula era ...".

Supondo que na célula atual havia o número 30, a macro, ao ser executada, deverá substituir o conteúdo da célula atual (30) por (O conteúdo desta célula era 30).

Teste o programa, depois de pronto, digitando numa célula o seu nome e vendo o resultado.

- 02) Faça aparecer na célula atual o seu nome.

- 03) Usando o comando **Date** faça com que apareça na célula atual a Data de Hoje (antes, atualize a data do seu computador).

- 04) usando o comando **Now** faça com que apareça na célula atual a Data e a hora atuais.

# RANGE

## O OBJETO RANGE

- Range (faixa) é um dos objetos que fazem o trabalho pesado no Excel. Normalmente antes de qualquer na Planilha, temos que identificar a área.
- Um objeto Range representa uma célula, uma fileira, uma coluna, uma seleção de células, contendo um ou mais blocos contíguos de células ou uma faixa em 3D.

## PROPRIEDADE RANGE

- Use **Range** (*argumento*), onde *argumento* nomeia o intervalo, para retornar um objeto **Range** representando uma única célula ou um intervalo de células.

## EXEMPLO 1

- O exemplo seguinte coloca o valor da célula A1 da planilha Plan1 na célula A5 da mesma planilha.

```
Worksheets ("Plan1").Range("A5").Value = _  
Worksheets("Plan1").Range("A1").Value
```

## EXEMPLO 2

- O exemplo seguinte preenche o intervalo A1:H8 com números randômicos definindo a fórmula para cada célula do intervalo.
- Quando usada sem um qualificador de objeto (um objeto à esquerda do ponto), a propriedade **Range** retorna um intervalo da planilha ativa. Se a planilha ativa não for uma planilha de trabalho, o método falhará.
- Use o método **Activate** para ativar uma planilha antes de usar a propriedade **Range** sem um qualificador de objeto explícito.

```
Worksheets("Plan1").Activate  
Range("A1:H8").Formula = "=rand()"
```

## EXEMPLO 3

- O exemplo seguinte limpa o conteúdo do intervalo chamado "Regiao" da 1ª. planilha.

```
Worksheets(1).Range("Regiao").ClearContents
```

- Se você usar um argumento de texto para o endereço do intervalo, você terá que especificar o endereço em notação de estilo A1 (você não poderá usar a notação de estilo L1C1).

## PROPRIEDADE CELLS

- Use **Cells**(*linha*, *coluna*) onde *linha* é o índice da linha e *coluna* é o índice da coluna, para retornar uma única célula.

## EXEMPLO 4

- O exemplo seguinte define o valor da célula A1 da primeira planilha como 24.

```
Worksheets(1).Cells(1, 1).Value = 24
```

## EXEMPLO 5

- O exemplo seguinte define a fórmula para a célula A2 da planilha ativa.

```
ActiveSheet.Cells(2, 1).Formula = "=sum(B1:B5)"
```

## EXEMPLO 6

- Embora você também possa usar `Range("A1")` para retornar a célula A1, pode haver ocasiões em que a propriedade **Cells** seja mais conveniente porque você pode usar uma variável para a linha ou coluna.
- O exemplo seguinte cria cabeçalhos de coluna e linha na Plan1. Observe que após a planilha ser ativada, a propriedade **Cells** pode ser usada sem uma declaração explícita de planilha (ela retorna uma célula da planilha ativa).

```
Sub Range_Q06()  
    Worksheets("Plan1").Activate      'Torna Plan1 Ativa  
  
    For Ano = 1 To 5  
        Cells(1, Ano + 1).Value = 1990 + Ano  
    Next Ano  
  
    For Linha = 1 To 4  
        Cells(Linha + 1, 1).Value = "Q" & Linha  
    Next Linha  
  
End Sub
```

## EXEMPLO 7

- Apesar de você poder usar funções de cadeia de caracteres do VBA para alterar as referências de estilo A1, é muito mais fácil (e é uma prática de programação muito melhor) usar a notação `CELLS(1,-1)`.
- Use *expressão.Cells(linha, coluna)*, onde *expressão* é uma expressão que retorne um objeto **Range**, e *linha* e *coluna* são relativas ao canto superior esquerdo do intervalo, para retornar parte de um intervalo.
- O exemplo seguinte define a fórmula para a célula B6.

```
Worksheets(1).Range("A5:D10").Cells(2, 2).Formula = "=rand()"
```

## RANGE E CELLS

- Use `Range(célula1, célula2)`, onde *célula1* e *célula2* são objetos **Range** que especificam as células inicial e final, para retornar um objeto **Range**.

## EXEMPLO 8

- O exemplo seguinte define o estilo da linha da borda das células A1:J10.

```
With Worksheets(1)  
    .Range(.Cells(1, 1), _  
        .Cells(10, 10)).Borders.LineStyle = xlThick  
End With
```

- Observe o ponto na frente de cada ocorrência da propriedade **Cells**. O ponto será obrigatório se o resultado da instrução **With** anterior for aplicado à propriedade **Cells** — nesse caso, para indicar que as células estão na planilha um (sem o ponto, a propriedade **Cells** retornaria as células da planilha ativa).

## PROPRIEDADE OFFSET

- Use `Offset(linha, coluna)`, onde *linha* e *coluna* são os deslocamentos de linha e coluna, para retornar um intervalo em um deslocamento especificado de um outro intervalo.

## EXEMPLO 9

- O exemplo seguinte seleciona a célula que se encontra 3 linhas abaixo e 1 coluna à direita da célula do canto superior esquerdo da seleção atual. Você não pode selecionar uma célula que não esteja na planilha ativa, portanto, você precisa ativar primeiro a planilha.

```
Worksheets("Plan1").Activate      'Torna Plan1 Ativada  
Selection.Offset(3, 1).Range("A1").Select
```

## MÉTODO "UNION"

- Use `Union(intervalo1, intervalo2, ...)` para retornar intervalos de várias áreas, isto é, intervalos compostos de dois ou mais blocos contíguos de células.

**EXEMPLO 10**

- O exemplo seguinte cria um objeto definido como a união de intervalos A1:B2 e C3:D4 e, em seguida, seleciona o intervalo definido.

```
Dim r1 As Range, r2 As Range
Dim myMultiAreaRange As Range
Worksheets("Plan1").Activate
Set r1 = Range("A1:B2")
Set r2 = Range("C3:D4")
Set myMultiAreaRange = Union(r1, r2)
myMultiAreaRange.Select
```

**EXERCÍCIOS**

- 01) Faça uma macro que grave "Bahia" na célula **B10**.
- 02) Faça uma macro que grave o conteúdo da **célula atual** na célula **C7**.
- 03) Faça uma macro que grave o conteúdo da célula **A1** na **célula atual**.
- 04) Faça uma macro que pegue o número da célula atual e grave as seguintes expressões: na célula **D5** deverá aparecer "**O número é ...**"; em **D6** deverá aparecer "**o seu dobro é ...**"; e, finalmente, em **D7** deverá aparecer "**e o seu triplo é ...**".
- 05) Faça uma macro que preencha todas as células do intervalo **A5:F10** da planilha **Plan2** com palpites para uma Loteria que sorteia dezenas de 0 a 60. Para testar a macro, execute-a a partir de **Plan1**.

**GABARITO**

01)

```
Sub Range_Q01 ()
    Range("B10") = "Bahia"
End Sub
```

02)

```
Sub Range_Q02 ()
    Range("C7") = ActiveCell
End Sub
```

03)

```
Sub Range_Q03 ()
    ActiveCell = Range("A1")
End Sub
```

04)

```
Sub Range_Q04 ()
    Range("D5") = "O número é " & ActiveCell
    Range("D6") = "O seu dobro é " & ActiveCell * 2
    Range("D7") = "E o seu triplo é " & ActiveCell * 3
End Sub
```

05)

```
Sub Range_Q05 ()  
    Sheets("Plan2").Activate  
    Range("A5:F10").Formula = "=int(rand()*60)"  
End Sub
```

# INPUTBOX

## O COMANDO INPUTBOX

- Exibe um aviso em uma caixa de diálogo, aguarda até que o usuário insira texto ou clique em um botão e retorna um String com o conteúdo da caixa de texto.
- Sintaxe: InputBox(prompt[, title] [, default] [, xpos] [, ypos] [, helpfile, context])
- Prompt = Obrigatório. Expressão de sequência de caracteres exibida como mensagem na caixa de diálogo. O comprimento máximo de prompt é de aproximadamente 1024 caracteres, dependendo da largura dos caracteres utilizados. Se prompt consistir em mais de uma linha, você poderá separar as linhas utilizando um caractere de retorno de carro (Chr(13)), um caractere de alimentação de linha (Chr(10)) ou uma combinação de caracteres de retorno de carro e alimentação de linha (Chr(13) & Chr(10)) entre cada linha.
- Title = Opcional. Expressão de sequência de caracteres exibida na barra de título da caixa de diálogo. Se você omitir title, o nome do aplicativo será inserido na barra de título.
- Default = Opcional. Expressão de sequência de caracteres exibida na caixa de texto como resposta padrão se nenhuma entrada for fornecida. Se você omitir default, a caixa de texto será exibida vazia.
- Xpos = Opcional. Expressão numérica que especifica, em twips, a distância horizontal da extremidade esquerda da caixa de diálogo em relação à extremidade esquerda da tela. Se xpos for omitido, a caixa de diálogo será centralizada horizontalmente.
- Ypos = Opcional. Expressão numérica que especifica, em twips, a distância vertical da extremidade superior da caixa de diálogo em relação ao topo da tela. Se ypos for omitido, a caixa de diálogo será posicionada na vertical a aproximadamente um terço da extremidade inferior da tela.
- Helpfile = Opcional. Expressão de sequência de caracteres que identifica o arquivo de Ajuda a ser utilizado para fornecer ajuda sensível ao contexto relativa à caixa de diálogo. Se helpfile for fornecido, context também deverá ser fornecido.
- Context = Opcional. Expressão numérica que é o número de contexto da Ajuda atribuído ao tópico da Ajuda apropriado pelo autor da Ajuda. Se context for fornecido, helpfile também deverá ser fornecido.

## EXERCÍCIOS

- 01) Faça uma macro que solicite o nome da pessoa, através da Caixa **InputBox** e armazene a frase **O nome é ...** na célula atual.
- 02) Faça uma macro que solicite um valor, através da Caixa **InputBox** e armazene em A10 = **O número dado é ...**; e em A11 = **A sua raiz quadrada é ...**; e em A12 = **E a sua raiz cúbica é ...**.
- 1) Faça uma macro que solicite ao usuário que insira um valor entre 1 e 3. A variável MeuValor deverá conter o valor inserido pelo usuário se o usuário clicar em OK ou pressionar a tecla ENTER. O Valor padrão deverá ser **2**. A tela deverá vir na posição horizontal = 100 e vertical = 500. Coloque como título a **Clínica de Matemática**;



# COMANDOS IF

## O COMANDO "IF"

- Executa condicionalmente um grupo de instruções, dependendo do valor de uma expressão.
- Sintaxe: If condição Then [instruções] [Else instruçõeselse]
- Observe que na sintaxe de 1 linha apenas, não há necessidade de **End If**.
- Você também pode utilizar a sintaxe de formato de bloco:

```
If condição Then      [instruções]
[ElseIf condição-n Then  [instruçõeselseif] ...
[Else      [instruçõeselse]]
End If
```

- A instrução If...Then...Else possui as partes a seguir:
- **Condição** = Obrigatória. Uma expressão numérica ou expressão de sequência de caracteres que é avaliada como True ou False. Se condição for Null, será tratada como False; Uma expressão do formato TypeOf nomedoobjeto Is tipodoobjeto. O nomedoobjeto é qualquer referência a objeto e tipodoobjeto é qualquer tipo de objeto válido. A expressão será True se nomedoobjeto for o tipo de objeto especificado por tipodoobjeto; caso contrário, será False.
- Você pode utilizar o formato de uma só linha (primeira sintaxe) para testes simples e curtos. Entretanto, o formato de bloco (segunda sintaxe) proporciona mais estrutura e flexibilidade do que o formato de uma só linha e normalmente é mais fácil de ler, manter e depurar.
- Observação Com o formato de uma só linha é possível ter múltiplas instruções executadas como resultado de uma decisão If...Then. Todas as instruções devem estar na mesma linha e separadas por dois-pontos, como na seguinte instrução:

```
If A > 10 Then A = A + 1 : B = B + A : C = C + B
```

- Uma instrução If em formato de bloco deve ser a primeira em uma linha. As partes Else, Elseif, e End If da instrução podem ter somente um número de linha ou rótulo de linha precedendo-as. O If em bloco deve terminar com uma instrução End If.
- Para determinar se uma instrução é ou não If em bloco, examine o que vem em seguida à palavra-chave Then. Se qualquer coisa exceto um comentário aparecer depois de Then na mesma linha, a instrução será tratada como instrução If de uma só linha.
- As cláusulas Else e Elseif são opcionais. Você pode ter tantas cláusulas Elseif em um bloco If quantas desejar, mas nenhuma pode aparecer depois de uma cláusula Else. Instruções If em bloco podem ser embutidas, isto é, contidas uma dentro da outra.
- Executando um If em bloco (segunda sintaxe), **condição** é testada. Se **condição** for True, as instruções seguintes a Then são executadas. Se condição for False, cada condição Elseif (se houver) será por sua vez avaliada. Quando uma condição True for encontrada, as instruções imediatamente subseqüentes à Then associada são executadas. Se nenhuma das condições Elseif forem True (ou se não houver cláusulas Elseif), as instruções subseqüentes a Else são executadas. Depois da execução das instruções subseqüentes a Then ou Else, a execução continua com a instrução subseqüente a End If.
- Dica: Select Case pode ser mais útil na avaliação de uma expressão única com diversas ações possíveis. Entretanto, a cláusula TypeOf nomedoobjeto Is tipodoobjeto não pode ser utilizada com a instrução Select Case.

## EXERCÍCIOS

- 01) Faça uma macro que avalie o número contido na célula A10 e colocar a sua classificação em B10. Se for até 5, classificar como **Pequeno**. Caso contrário, classificar como **Grande**.
- 02) Faça uma macro que solicite um número entre 50 e 100, usando o comando inputbox. Se o número digitado estiver entre 50 e 75, colocar na célula atual **Faixa 1**; se o número for maior que 75, até 100, colocar na célula atual **Faixa 2**; Se a pessoa digitar qualquer outro número fora desta faixa, colocar na célula atual **Erro**.

- 03) Usando a função **IsNumeric** (que retorna um valor Boolean que indica se uma expressão pode ser avaliada como um número), faça uma macro que solicite do usuário a idade dele e calcule quantos dias ele já viveu; mas a macro deverá analisar o número digitado e dar uma mensagem de erro na célula atual se o usuário não digitar um número.

# LAÇOS FOR

## LAÇOS

- São amarrações feitas nos programas para que possamos executar instruções um determinado número de vezes.
- No exemplo a seguir, vamos fazer um programa que irá gerar os números da MEGA SENA.

```
Option Explicit

Sub Megasena()

'Declaração de Variáveis
Dim Contador As Integer
Dim Palpite As Integer
Dim Teste As Integer
Dim Numero
Dim Erro

Worksheets("Loteria").Activate
Range("A1:H100").Value = ""

PedeDezena:
Numero = InputBox("Quantas dezenas você deseja?", "Megasena")

If Numero = "" Then Exit Sub

If IsNumeric(Numero) = False Then
    MsgBox "Número de dezenas inválido! Redigite!", vbCritical
    GoTo PedeDezena
End If

Numero = Val(Numero)

If Numero < 6 Or Numero > 20 Then
    MsgBox "ERRO: A quantidade de dezenas deverá estar entre 6 e 20", vbRetryCancel
    GoTo PedeDezena
End If

Range("A4").Value = "Palpites para a Megasena"
Range("A4").Select

For Contador = 1 To Numero
    ActiveCell.Offset(0, 0).Select
GeraDezena:
    Randomize
    Palpite = Int(Rnd() * 60)
    If Palpite = 0 Then GoTo GeraDezena

'Testa se está repetido
    For Teste = 1 To Contador
        If Palpite = Range("A4").Offset(Teste, 1) Then GoTo GeraDezena
    Next

    Palpite = Range("A4").Offset(Teste, 1)

'Grava na tela
    ActiveCell.Offset(Contador, 0).Value = "Dezena " & Contador
    ActiveCell.Offset(Contador, 1).Value = Palpite
Next

'Ordena registros
Range("B5:B25").Select
Selection.Sort Key1:=Range("B5"), Order1:=xlDescending
Range("A1").Select

End Sub
```

# OFFSET

## O COMANDO "OFFSET"

- Faz com que você estabeleça uma relação com as células do Excel, sem defini-las com exatidão.
- Pode ser usada combinada com o comando ACTIVECELL.

## EXERCÍCIOS

01) Criar um tipo de FORMULÁRIO de entrada de dados com o comando INPUTBOX. Fazer as críticas dos valores digitados.

```
Sub BD_1()  
    'Declaração de variáveis  
    Dim Titulo As String  
    Dim Nome As String  
    Dim Formacao As String  
    Dim Idade As Integer  
    Dim Continua As Integer  
    Dim Informa As String  
  
    'Entrada de dados  
    Titulo = "Banco de Dados"  
    Informa = "Dados cadastrados com sucesso!"  
    Informa = Informa & Chr(10) & Chr(10) & "Deseja incluir novo registro?"  
  
    Inicio:  
  
    PedeNome:  
    Nome = InputBox("Digite o nome", Titulo)  
  
    If Nome = "" Then  
        MsgBox "Digite o nome!"  
        GoTo PedeNome  
    End If  
  
    PedeFormacao:  
    Formacao = InputBox("Digite a formação", Titulo)  
  
    If Formacao = "" Then  
        MsgBox "Digite a formação!"  
        GoTo PedeFormacao  
    End If  
  
    PedeIdade:  
    Idade = InputBox("Digite a idade", Titulo)  
  
    If IsNumeric(Idade) = False Then  
        MsgBox "Digite uma idade válida!"  
        GoTo PedeIdade  
    End If  
  
    'Identificar primeira célula vazia  
    Worksheets("BDados").Activate  
    Range("A5").Select  
  
    Do  
        ActiveCell.Offset(1, 0).Select  
  
        If ActiveCell.Value = "" Then  
            Exit Do  
        End If  
    Loop
```

```
Loop
'Armazena dados
ActiveCell.Value = Nome
ActiveCell.Offset(0, 1).Value = Formacao
ActiveCell.Offset(0, 2).Value = Idade
Continua = MsgBox(Informa, vbYesNo + vbExclamation, Titulo)
If Continua = vbYes Then GoTo Inicio Else Exit Sub
End Sub
```

# FORMULÁRIOS

## CRIAÇÃO DE FORMULÁRIOS

- É possível criar Formulários de entrada de dados no VBA para uso no EXCEL.
- Os comandos são similares aos das macros.
- Observe o exemplo a seguir, que cria um formulário para entrar dados no Banco de Dados do Excel.

```
Option Explicit    'Força a declaração das variáveis nesta módulo

Sub Formulario()

    '1) Declaração de variáveis -----
    Dim Dialogo, cxBoxOK, vNome, vSexo, vFormacao

    '2) Diminui o nome da caixa de diálogo -----
    Set Dialogo = DialogSheets("Caixa1")

    '3) Estabelece o início do programa -----
    Inicio:

    '4) Limpa a caixa de diálogo (Formulário) -----
    Dialogo.EditBoxes("cxNome").Text = ""

    '5) Exibe a caixa de diálogo -----
    cxBoxOK = Dialogo.Show

    '6) Fecha a caixa de diálogo se pressionar ESC -----
    If Not cxBoxOK Then Exit Sub

    '7) Recebe os dados da caixa de diálogo -----
    vNome = Dialogo.EditBoxes("cxNome").Text    'Recebe o nome
    If Dialogo.OptionButtons("OpcaoM").Value = xlOn Then    'Recebe o sexo
        vSexo = "Masculino"
    Else
        vSexo = "Feminino"
    End If

    '7c) Recebe a formação
    vFormacao = Worksheets("Formacao").Range("D3").Value

    '8) Armazena as variáveis
    '8a) Vai para a primeira linha vazia do banco de dados
    Worksheets("BDados2").Activate
    Range("A5").Select

    Do
        ActiveCell.Offset(1, 0).Select
        If ActiveCell.Value = "" Then Exit Do
    Loop

    ActiveCell.Value = vNome
    ActiveCell.Offset(0, 1).Value = vSexo
    ActiveCell.Offset(0, 2).Value = vFormacao

    '9) Repete até cancelar
    GoTo Inicio

End Sub
```

**EXERCÍCIOS**

```
Sub BD_2()  
  
    'Declara variáveis  
  
    Dim Dialogo, cxBoxOK, vNome, vSexo, vFormacao  
  
    'Diminui o nome da caixa de diálogo  
  
    Set Dialogo = DialogSheets("Caixa1")  
  
    'Estabelece o início do programa  
  
Inicio:  
  
    'Limpar caixa de diálogo  
  
    Dialogo.EditBoxes("cxNome").Text = ""  
  
    'Exibir a caixa de diálogo  
  
    cxBoxOK = Dialogo.Show  
  
    'Fecha a caixa de diálogo se pressionar ESC  
  
    If Not cxBoxOK Then Exit Sub  
  
    'Recebe os dados da caixa de diálogo  
    '1) Recebe o nome  
  
    vNome = Dialogo.EditBoxes("cxNome").Text  
  
    '2) Recebe o sexo  
  
    If Dialogo.OptionButtons("OpcaoM").Value = xlOn Then  
        vSexo = "Masculino"  
    Else  
        vSexo = "Feminino"  
    End If  
  
    '3) Recebe a formação  
  
    vFormacao = Worksheets("Formacao").Range("D3").Value  
  
    'Armazenar as variáveis  
    'Vai para a primeira linha vazia do banco de dados  
  
    Worksheets("BDados2").Activate  
    Range("A5").Select  
    Do  
        ActiveCell.Offset(1, 0).Select  
        If ActiveCell.Value = "" Then Exit Do  
    Loop  
  
    ActiveCell.Value = vNome  
    ActiveCell.Offset(0, 1).Value = vSexo  
    ActiveCell.Offset(0, 2).Value = vFormacao  
  
    'Repete até cancelar  
  
    GoTo Inicio  
  
End Sub
```

# **SENHAS DE SEGURANÇA**

## **SENHA**

- Uma palavra, frase ou seqüência de caracteres que deve ser novamente inserida para se ter acesso a uma célula, gráfico, objeto, planilha, pasta de trabalho, pasta ou arquivo protegido.
- No EXCEL, uma senha pode conter até 255 caracteres com qualquer combinação de letras, números, espaços e símbolos.
- À medida que você digita a senha, o EXCEL exibe um asterisco (\*) para cada caractere que você digitar. Como as senhas fazem distinção entre maiúsculas e minúsculas, você deve digitar as letras maiúsculas e minúsculas corretamente ao definir e inserir senhas.

## **SENHA DE PROTEÇÃO**

- A senha de proteção é aquela que protege a pasta de trabalho inteira, ou seja, o arquivo só será aberto se a senha for informada corretamente.
- Você pode colocar uma senha de proteção na abertura de um arquivo no Excel, ou seja, só poderemos abrir o arquivo se soubermos a senha.
- Quando você cria uma senha, deve copiá-la e guardá-la em um local seguro. Se você perder a senha, não poderá abrir a pasta de trabalho protegida por senha nem ter acesso aos dados nela contidos.
- Vamos fazer isso com a nossa pasta "CursoExcelAvancado", que estamos usando desde o início do curso (se você não tem ainda esta pasta, crie uma).
- Estando com a pasta aberta, proceda os seguintes comandos:
  - Arquivo
  - Salvar Como (ou F12)
  - Opções
  - Senhas de Proteção

## **OBSERVAÇÕES**

- Se a versão do seu Office é "2000", então proceda os seguintes comandos:
  - Arquivo
  - Salvar Como (ou F12)
  - Ferramentas (Barra de Ferramentas)
  - Opções Gerais
  - Senhas de Proteção
- Na caixa **Senha de proteção**, digite uma senha e clique em **OK**.
- As senhas coincidem maiúsculas e minúsculas. Digite a senha exatamente como deseja que os usuários a insiram, incluindo letras maiúsculas e minúsculas.
- Na caixa **Reinsira a senha para prosseguir**, digite a senha novamente e clique em **OK**.
- Clique em **Salvar**.
- Se for solicitado, clique em **Sim** para substituir a pasta de trabalho existente pela pasta de trabalho aberta.

## **SENHA DE GRAVAÇÃO**

- Você pode colocar uma senha apenas para proteger a planilha original de gravações, ou seja, você permite que qualquer pessoa abra a Pasta, pode até fazer alterações, mas não conseguirá salvar com o mesmo nome, a não ser que a pessoa digite a senha.
- O usuário poderá, entretanto, salvar as alterações feitas com outro nome.



- Estando com a pasta aberta, proceda os seguintes comandos:
- Arquivo
- Salvar Como (ou F12)
- Opções
- Senhas de Proteção
- Se a versão do seu Office é "2000", então proceda os seguintes comandos:
- Arquivo
- Salvar Como (ou F12)
- Ferramentas (Barra de Ferramentas)
- Opções Gerais
- Senhas de Proteção
- Na caixa **Senha de gravação**, digite uma senha e clique em **OK**.
- Na caixa **Reinsira senha para modificar**, digite a senha novamente e, em seguida, clique em **OK**.
- Clique em **Salvar**.
- Se for solicitado, clique em **Sim** para substituir a pasta de trabalho existente pela pasta de trabalho aberta.
- **Observações:** Você pode salvar as alterações em uma pasta de trabalho protegida por senha sem usar a senha, abrindo primeiro a pasta de trabalho como **somente leitura**. Faça as alterações desejadas na pasta de trabalho e, em seguida, salve a pasta com um nome diferente. A pasta de trabalho salva com um novo nome não necessita de uma senha e estará disponível para edição.

## **SENHAS NAS CÉLULAS**

- Você pode permitir que os usuários abram a sua Pasta de Trabalho e possam alterar dados e salvá-los. Entretanto, não poderão alterar determinadas células, como por exemplo, células que contenham fórmulas.
- Vamos supor que você queira proteger apenas as células "A1:B10". Para proteger, destrave quaisquer células que você deseja poder alterar depois que proteger a planilha. Como esta parte é um pouco complicada e como, por padrão, todas as células de qualquer planilha vêm protegidas, a sugestão é que primeiro você destrave todas elas e depois trave apenas as que você quer proteger.
- Faça assim: Marque toda a planilha (CTRL-T). Depois clique em Formatar / Células (ou CTRL-1) / Proteção / Travada.
- Destrave todas as células marcadas
- Agora marque as células que você deseja proteger e digite novamente os comandos: Formatar / Células (ou CTRL-1) Proteção / Travada.
- Marque travada para as células.
- Quando você protege a planilha, as células e os objetos gráficos que você não destravou permanecem protegidos e não poderão ser alterados.
- Oculte quaisquer fórmulas que você não deseja que fiquem visíveis.
- No menu **Ferramentas**, aponte para **Proteger** e clique em **Proteger planilha**.
- Para impedir alterações nas células das planilhas ou nos dados e outros itens em gráficos, e para impedir a exibição de linhas, colunas e fórmulas ocultas, marque a caixa de seleção **Conteúdo**.
- Para impedir alterações nos objetos gráficos de planilhas ou gráficos, marque a caixa de seleção **Objetos**.
- Para impedir alterações nas definições de cenários em uma planilha, marque a caixa de seleção **Cenários**.
- Para impedir que outros usuários removam a proteção da planilha, digite uma senha, clique em **OK** e, em seguida, redigite a senha na caixa de diálogo **Confirmar senha**. As senhas coincidem maiúsculas e minúsculas. Digite a senha exatamente como deseja inseri-la, incluindo letras maiúsculas e minúsculas.
- Se você atribuir uma senha, deve copiá-la e guardá-la em local seguro. Se perder a senha, não terá acesso aos elementos protegidos na planilha.

## **SENHAS NOS MÓDULOS DO VBA**

- Para proteger os seus módulos do VBA, ...

**EXERCÍCIOS PROPOSTOS**

- 1) Crie uma planilha e salve-a com o nome de "Senhas". Digite em "A1" "Valores" e em "B1" "Dobro". De "A2" até "A7" digite 10, 20, 30, 40, 50 e 60. De "B2" até "B7" digite uma fórmula que irá calcular o dobro dos valores de "A2" até "A7" ( $=A2*2$ ). A planilha deverá ficar assim:

VALORES	DOBRO
10	20
20	40
30	60
40	80
50	100
60	120

- 02) Coloque uma senha de proteção (1234) na planilha. Feche-a e tente abrir digitando a senha.
- 03) Retire a senha de proteção da planilha.
- 04) Coloque uma senha de gravação (4321) na planilha. Feche-a e abra a pasta sem colocar a senha. Faça alguma alteração e depois tente salvar.
- 05) Retire a senha de gravação.
- 06) Coloque uma senha de proteção para as células que têm fórmulas (de B2 até B7). Depois tente digitar algo naquelas células.
- 07) Retire a senha de proteção de células.

# **RESUMO DOS PRINCIPAIS COMANDOS DO EXCEL NO VBA**

## **ABRIR UM ARQUIVO**

- Sintaxe: `Workbooks.Open Filename:= "endereço completo"`
- Exemplo: o comando a seguir abre o arquivo `Calculos.xls` que se encontra na pasta `Meus Documentos`  
`Workbooks.Open Filename:= "C:\Meus Documentos\Calculos.xls"`

## **APAGAR O CONTEÚDO DE UM CONJUNTO DE CÉLULAS**

`Selection.ClearContents`    `Equivalente a pressionar DEL

## **COPIAR UMA CÉLULA**

- Sintaxe: `Referência.Copy [Destino]`.
- Exemplos:
  - a) `ActiveCell.Copy Range("A1")`    `Copia a célula ativa para a célula A1.
  - b) `Selection.Copy`    `copia a região atualmente selecionada para a área de transferência
  - c) `Range("D20").Copy`    `copia a célula D20 para a área de transferência
  - d) `Range("D20").Copy Range("D21")`    `copia a célula D20 para a célula D21
  - e) `Range("D20").Copy Range("D21:F21")`    `copia a célula D20 para todas as células da região D21:F21
  - f) `ActiveCell.CurrentRegion.Copy Range("A1")`    `copia a região atual para a célula A1. Não há necessidade que a região esteja selecionada.

## **FORMATAR PARA ITÁLICO**

`Selection.Font.Italic = True`

## **FORMATAR PARA NEGRITO**

`Selection.Font.Bold = True`

## **FORMATAR PARA SUBLINHADO**

- a) `Selection.Font.Underline = True`    `sublinha sem especificar o tipo
- b) `Selection.Font.Underline = xlUnderlineStyleSingle`    `Sublinha com linha simples
- c) `Selection.Font.Underline = xlUnderlineStyleDouble`    `Sublinha com linha dupla

## **FUNÇÕES DO EXCEL NO VBA (TRADUZIDAS)**

Abs → Abs  
Agora → Now  
Aleatório → Rand  
Arred → Round  
Caract → Chr  
Código → Code  
Col → Column  
Cols → Columns  
Concatenar → Concatenate  
Cont.Núm → Count

Cont.Se → CountIf  
E → And  
Int → Int  
Média → Average  
ProcV → Vlookup  
Soma → Sum  
SomaSe → Sumif  
VP → PV

## **INSERIR COLUNAS**

```
Selection.EntireColumn.Insert
```

## **INSERIR LINHAS**

```
Selection.EntireRow.Insert
```

## **MOSTRAR A COLUNA ATUAL**

- Sintaxe: ActiveCell.Column
- Exemplo:

```
y = ActiveCell.Column  
MsgBox "A coluna onde você está é " & y
```

## **MOSTRAR A LINHA ATUAL**

- Sintaxe: ActiveCell.Row
- Exemplo:

```
x = ActiveCell.Row  
MsgBox "A linha onde você está é " & x
```

## **OCULTAR A LINHA ATUAL**

```
ActiveCell.EntireRow.Hidden = True
```

# **PRINCIPAIS FUNÇÕES E SUAS SINTAXES**

## **ABS (X)**

- Mostra o valor absoluto de x (positivo).

- Exemplos:

```
1 - se    y = Abs (-4)    então    y = 4
2 - se    y = Abs (10)    então    y = 10
3 - Se    y = Abs (10-11)    então    y = 1
```

## **AND**

- Função lógica (AND = E) usada para operações de avaliação.

- Exemplo:

```
If x > 2    And    x < 10    Then ...
```

## **BEEP**

- Emite sinal sonoro.

- Exemplo:

```
If y = "Erro" Then Beep
```

## **CHR(CÓDIGO NUMÉRICO)**

- Controla os códigos "ASCII" (conjunto de caracteres de 7 bits do American Standard Code for Information Interchange, utilizado para representar letras e símbolos encontrados em um teclado americano padrão) do teclado.
- O conjunto de caracteres ASCII é o mesmo dos 128 primeiros caracteres (0 a 127) do conjunto de caracteres ANSI.
- A seguir listamos alguns códigos importantes.

```
Chr(10) = Mudança de linha (Simula o pressionamento da tecla ENTER).
Chr(13) = Tem a mesma função do Chr(10), ou seja, Mudança de linha.
Chr(32) = Espaço (Simula o pressionamento da tecla BARRA DE ESPAÇO)
Chr(33) = !
Chr(35) = #
Chr(36) = $
Chr(37) = %
Chr(38) = &
Chr(65) = A
Chr(97) = a
Chr(62) = >
Chr(37) = %
Chr(128) = €
```

## **COLUMN (CÉLULA)**

- Mostra a coluna (em número) em que você está.

- Exemplos:

```
X = Range("D2").Column    'neste caso x = 4 que corresponde à coluna D
x = Range("d:d").Column    'x = 4, que corresponde à coluna D
x = ActiveCell.Column    'Mostra a coluna em que você está no momento
```

## **INT (X)**

- Mostra a parte inteira de x.

- Exemplos:

```
1 - se   y = Int (2,32)   então   y = 2
2 - se   y = Int (10)     então   y = 10
3 - se   y = Int (10/3)   então   y = 3
```

## **NOW ()**

- Equivalente à função “=Agora()”, mostra a data e a hora do momento.

## **RND ()**

- Equivalente à função “=Aleatório()”, gera um número aleatório (randômico), compreendido entre 0 e 1.

- Exemplos:

```
X = Rnd()      'Gera um número aleatório entre 0 e 1.
X = Rnd()*10    'Gera um número aleatório entre 0 e 10.
X = Int(Rnd()*10) 'Gera um número inteiro aleatório entre 0 e 10.
```

## **ROUND (NUM, NUM DE DÍGITOS)**

- Equivalente à função “=Arred()”, arredonda o número para o número de dígitos fornecido.

- Exemplos:

```
X = Round(2.39 , 1)    'equivale a arredondar 2,39 para 1 casa (x = 2,40)
X = Round (234.4 , 0)  'equivale a arredondar 234,4 para nenhuma casa (x = 234)
```