

CURSO
DE MACROS DO
LOTUS 1-2-3
R5W
(Português)

ÍNDICE

1 – Introdução

2 - Estrutura do macro

2.1 - Digitando o Macro

2.2 - Nomeando o Macro

2.3 - Executando o Macro

3 – Teclado Equivalente para Macro

4 – Comandos de Idioma de Macro

5 – Controle da Tela

5.1 - {panelon}, {paneloff}

5.2 - {windowson}, {windowsoff}

5.3 - {buzine}

5.4 - {indicate}

6 – Interagindo Com o Mundo

6.1 - Comando {?}

6.2 - {acquire}

6.3 - {getlabel}. {getnumber}

6.4 - {menubranh}, {memucall}

6.5 - {breakon}, {breakoff}

7 – Movendo e fazendo Looping

7.1 - {branch}

7.2 - {if}

7.3 - {quit}

7.4 - {for}

7.5 - {forbreak}

7.6 - Subrotinas de Macro

8 – Manipulando Dados

8.1 - {blank}

8.2 - {let}

9 – Considerações de Cálculo

10 – Exemplos de Macros

10.1 - Digitando Macros

10.2 - Somando uma Coluna

10.3 - Dados

10.4 - Exemplos de Menus

10.5 - Largura de Colunas

10.6 - Imprimindo com o {for}

10.7 - Combinação de Arquivos e a Subrotina de Macro

MACROS DO LOTUS 1-2-3 V.5w

1 – Introdução

Macros são recursos avançados do Lotus 1-2-3, que permitem automatizar muitas funções da planilha eletrônica. Há duas partes no idioma de macro do 1-2-3:

* Macros que fazem armazenar toques de tecla em células. Estes toques de tecla são exatamente os mesmos toques de tecla que você digitaria no 1-2-3 ao executar alguma tarefa ou comando--talvez um comando de menu, movimentos de cursor, ou entrada de dados. Quando você roda o macro, estes toques de tecla são executados no 1-2-3, da mesma maneira que você os teria digitado.

* Os Macros do 1-2-3 também podem conter comandos de macro avançados que deixam que os macros tomem decisões, processos repetidos várias vezes, e menus presentes..

A condição mínima que você precisa para desenvolver macros é um conhecimento íntimo do 1-2-3. Isso é porque quando você escreve um macro, você está usando o 1-2-3 sem ver os resultados de seus comandos. Isso é no princípio enganador, mas você logo saberá para o que são usados os programas de macro.

Em sua forma mais básica, um macro é um conjunto de toques de tecla armazenados na planilha.

Você digita estes toques de tecla em uma séries de células na planilha. Então, você dá um nome de região à célula de início do macro de forma que 1-2-3 pode achar o macro quando você quiser encontrá-lo.

Por exemplo, suponha que você se ache freqüentemente formatando uma única célula com o formato de Moeda corrente, com duas casas decimais. Esta tarefa é uma boa tarefa para um macro--algo que se repete muitas vezes. Agora, pense nos toques de tecla usados no processo:

- 1) Pressione a tecla **slash** (/) para ativar o menu.
- 2) Pressione R para o comando de região.
- 3) Pressione F para o comando de formato
- 4) Pressione C para o formato de moeda corrente.
- 5) Digite 2 e pressiona Enter para o número de casas decimais.
- 6) Pressione Enter novamente para responder o pedido de formato de região

Na notação de macro do 1-2-3, estes comandos se mostrarão assim:

/rfc2~~

Na realidade, isso é um macro. Você poderá reconhecer muitos dos caracteres no macro como representando as teclas que você digitaria para formatar a célula. O **til** (~) representa um toque da tecla Enter em um macro.

2. Estrutura de macro

O tipo mais comum do macro do 1-2-3 é uma séries de toques de tecla. Você digita os toques de tecla como um texto em uma célula ou células na planilha e um comando do 1-2-3 executa os toques de tecla. O primeiro passo usando um macro, então, é digitar os toques de tecla em uma célula ou células na planilha.

2.1 Digitando um Macro

Quando digitando o macro, escolha um local para ele. Por exemplo, você poderia começar seus macros na célula A7000--certamente um local que a maioria das planilhas não usarão. Considerando que os macros começam com uma região nomeada, você pode voltar rapidamente a eles pressionando a tecla *Goto*(função **F5**) e apertando a tecla de Nome (função F3) para digitar um menu de região e nomes de macro.

Lembre-se, macros são textos, e você pode manipulá-los com quaisquer dos comandos do 1-2-3 que trabalham com textos—copia (copy), mova (move), apague (delete), tudo que você pode digitar com toques de tecla de macro, em qualquer forma que você quiser. Você poderia digitar os toques de tecla para o macro do exemplo discutido acima em uma única célula assim:

/rfc2 ~ ~

Ou, você poderia colocar cada toque de tecla em sua própria célula assim:

/
r
f
c
2
~
~

Ambos estes métodos funcionam igualmente bem, porque o processador do macro do 1-2-3 começa com a célula de topo do macro, executa os toques de tecla naquela célula, baixa para a próxima célula, executa os toques de tecla, e repetem até que ache um {quit}.

Quando você adquirir mais experiência com macros, você achará que faz sentido se agrupar toques de tecla, juntos, em uma única célula. A maioria dos usuários escreveria este macro simples em uma única célula.

Quando digitar a tecla de um macro, você tem que digitar freqüentemente um prefixo de texto para forçar o 1-2-3 a entrar **a sua** digitação a uma célula. Por exemplo, o primeiro toque de tecla neste macro é a **tecla /**. Se você mover o cursor para uma célula e digitar a **tecla /** (slash), o 1-2-3 produz os menus. Mais adiante a digitação executará o comando de menu ou produz um erro *buzina*. Para digitar este macro, você tem que começar com um prefixo de texto (geralmente a **apóstrofe (')** (para alinhamento à esquerda).

Assim, para digitar este macro, você digitaria de fato **'/rfc2 ~ ~**, e clica Enter. Em geral, quando uma célula de macro começa com **tecla /** ou um número, uma fórmula ou um texto que o macro deveria digitar, você tem que começar a célula de macro com um prefixo de texto.

2.2 Nomeando um Macro

Uma vez você digitou o macro na tela, você tem que dar um nome de região à célula topo do macro. Nomes de região de macro são a **tecla **, seguido por uma única letra do alfabeto. Por exemplo, você poderia dar para este macro o nome de região **\c**, o **"c"** representando o formato de moeda corrente.

Considerando que os nomes de macro podem ter apenas uma única letra do alfabeto, você pode ter apenas 26 macros numa planilha de 26 linhas. Mais tardes, entretanto, nós veremos que um único macro poderá apresentar um menu de outros macros que não precisam ter nomes com uma única letra, assim, usando menus, você pode ter tantos macros quantos você quiser numa página..

Existe um macro especial nomeado **\0** (é o dígito **zero**, não a letra **O**). Este macro é chamado o macro *auto-executável*, porque toda vez você executa o comando ***Recover***, o 1-2-3 percorre a planilha procurando por um macro chamado **\0**. Se o 1-2-3 achar um tal macro, executa-o automaticamente. Você não pode digitar **Alt-0** do teclado, pois isto auto-executaria o macro. Assim, se você quiser executar o macro **\0** do teclado, você terá que dar a ele um nome de macro adicional.

Para nomear um macro, você deve ir para a célula de topo de um macro e digitar o comando **Nome de Região Crie**. Outro modo para nomear um macro é digitar o nome do macro à esquerda na célula de topo do macro.

Então, use o comando **Região Nome Textos Direita** para nomear o macro. Esta técnica torna fácil nomear um macro, e também documenta o nome do mesmo. Nós veremos um exemplo disto, logo a seguir..

Alguns macros são de uso específico, e só têm valor na planilha para a qual eles foram criados. Outros, como o macro de dados, mais tarde usado como um exemplo são mais gerais e úteis em diversas planilhas. Ao desenvolver estes macros gerais, você os poderia criar em uma planilha de teste, e depois de testado e aperfeiçoado, salvos nas suas planilhas, contendo apenas um único macro.

Então, quando você decidir usar um destes macros em uma planilha nova, você pode usar o comando **/Arquivo Combina Fórmulas Inteiro-Arquivo** para trazer o macro para a planilha atual. Uma dificuldade são aqueles nomes de região que não estão na planilha atual, assim você terá que criar as regiões nomeadas para uso do macro. Isso é outra boa razão para se usar a técnica de nomeação mencionada acima, estabelecendo mais facilmente as regiões nomeadas.

2.3 Executando o Macro

Depois que você tenha nomeado o macro, você pode executá-lo. Uma vez que este macro é nomeado **\c**, você poderá executá-lo mantendo a tecla ***Alt*** pressionada e clicando a letra **C**. Neste momento, o processador **toma as rédeas** e executa os toques de tecla como se você as tivesse digitando diretamente do teclado.

Antes de executar um macro, você poderá pressionar a tecla ***Step***, que é a **Alt-F1**. Esta ação o coloca no modo **STEP**, como indicado embaixo da tela. Agora, se você executar o macro, o 1-2-3 o executa passo a passo.. Entre dois passos o 1-2-3 usa um indicador **SST** piscando no fundo da tela, chamando a atenção para pressionar qualquer tecla para continuar para o próximo passo. Para deixar o simples modo ***Step*** e retornar para o regular, obtendo a total velocidade de execução do macro, pressionando novamente a tecla **Alt-F1**.

No momento em que o 1-2-3 executa o macro, você verá o número **100** mudando para **\$100,00**, como evidência de que o macro funciona. Se você olhou com atenção, você teria visto o menu piscando no topo da tela. Se você quiser, execute o macro novamente e olha para o painel de controle operando, mas se você tiver um computador muito rápido, não verá nada.

Para interromper um macro, aperte a **Tecla Ctrl + f**. Na maioria das vezes, o 1-2-3 exibirá o indicador de **ERRO** à direita e no topo da tela, assim pressione **Fuga** ou **Enter** para corrigir o erro e voltar a modo **PRONTO**.

Segue-se uma descrição detalhada para criar e executar este macro. Comece este exercício com uma tela em branco.

Mova para a célula **B1** Nós colocaremos o macro na célula **B1**. Embora este não seja um lugar apropriado para a maioria dos macros, faremos para este exercício.

Digite **/rfc2 ~** e clique **Enter**. Este é o macro atual. Lembre-se de digitar o prefixo de texto **~**. Caso contrário, você não estará entrando um texto à célula, você estará executando, de fato, o comando de **Formato de Região**.

Mude para célula **A1** Nós digitaremos o nome do macro nesta célula.

Teclle **\c** e pressione **Enter**. Este é o nome do macro. O nome da região que nós queremos usar é **/c**. Use-o **/**, não o comando que chama o menu do 1-2-3. Você tem, também que digitar um prefixo de texto antes, ou o **/** se torna o prefixo de texto repetido e encherá a célula com a letra **c**.

Digite **/Começo de rnlr os Textos de Nome de Região Certo**.

Este comando levará os conteúdos de uma região e moverá para a direita e cada célula terá um nome de região, determinando o nome de região que é o conteúdo de cada célula da região.

Clica **Enter** e responde qual a região para nomear, enquanto nomeando só a célula **B1** com o nome **/c**.

Isto completa a *fabricação* do macro. Vamos executá-lo agora:

Mova para célula A15, digite 100, e pressione Enter em uma célula para testar o macro.
A tecla **Alt-C** executa o macro.

3. Teclado Equivalente para Macros

Cada tecla especial no teclado tem uma notação para uso em um macro. Aqui é uma lista de algumas delas:

Os nomes de função fundamentais para uso em um macro.

{esc} ou {saida} A tecla de Saida.

{bs} ou {retrocesso} A tecla de retrocesso.

{del} ou {apague} A tecla de apagar

.

{~}, {{}, e {} Use estes comandos um ~, {, ou } em um macro.

Teclas do Macro	Teclas do Teclado
~ (o til)	A tecla Enter (~) é muito importante em qualquer macro, porque quando digitada em um macro, na realidade ele o encerra. Omitir o Enter é um engano comum cometido ao se escrever ma
{up}, {down}, {left}, {right}, {pgup}, {pgdn}, {home}, {end}, {bigleft}, {bigright}	{ {p. cima}, {p. baixo}, {esquerda}, {direita}, {pgup}, {pgdn}, { {end}, bigleft}, {bigright} Estas são as teclas de navegação de Com estas teclas como com outros nomes de macro, você pode um fator de repetição para movimentos múltiplos. Por exemplo, em vez de escrever {up} {up} {up} para subir três l você pode escrever, simplesmente, {up3}.
{edit}, {name}, {abs}, {goto}, {window}, {query}, {table}, {calc}, {graph}	A função de nomes fundamentais para uso em um macro.
{escape} ou {esc}	A tecla de saída..
{backspace} ou {bs}	A tecla de backspace . (espaço)

{delete} ou {del}	A tecla de apagar ..
{~}, {{}, e }	Estas teclas são constantemente usadas em macros.

4. Comandos de Idioma de macro

Além do teclado, teclas de função, comandos de menu, o idioma de macro do 1-2-3 contém muitos outros comandos de macro. Esta referência não descreve todos os comandos do macro, pois alguns deles são bastante complexos e especializados.

5. Controle da Tela

5.1 {panelon}, {paneloff}

Quando o 1-2-3 executa um macro, digita a tecla do macro da mesma maneira que você estava os digitando. O painel de controle também operará, mostrando textos, lembretes, ou outra informação. Enquanto a operação do painel de controle for inofensiva, pode confundir alguém que está usando seus macros e não está familiarizado com 1-2-3. Também, a operação do painel de controle usa muito tempo do processador do computador. Se você coloca o comando {paneloff} no topo do macro, o 1-2-3 não exibirá mudanças no painel de controle. {panelon} restabelece operação do painel de controle. Ao término de um macro, o 1-2-3, automaticamente, volta o painel de controle ao modo off.. Este é um comando *cosmético*, não usado, necessariamente..

5.2 {windowson}, {windowsoff}

De uma maneira semelhante para {paneloff}, {windowsoff} congela a tela, de forma que uma mudança de movimentos do cursor não será exibida. Isto é útil porque torna mais rápida a execução do macro. Fazendo novo desenho na tela o 1-2-3 leva muito tempo, enquanto eliminando o desenho o macro se torna muito mais rápido. O comando {windowson} restabelece atualização de tela.

5.3 {buzine}

O comando {buzine} – o computador emite sons de comando durante a execução de um macro. Você pode controlar o comando *buzine* usando um número de 1 a 4 como um argumento, como em {buzine 3}. Use o comando {buzine} para chamar a atenção do usuário durante a execução de um macro.

5.4 {indicate}

O comando de macro {indicate} fá-lo controlar o indicador de modo à direita e no topo da tela. Por exemplo, se você colocasse o comando {indicate Wait} em um macro, o indicador de modo exibirá a palavra WAIT. O uso do comando {indicate}, sem argumentos, restabelece o modo normal de **PROMPT** do 1-2-3.

Se você mudar o indicador de modo durante a execução de um macro, o 1-2-3 interrompe o indicador de modo, mesmo quando o macro terminar. Normalmente, você deverá usar o comando *indicate* sem argumento (como em {indicate}) ao término de um macro, restabelecendo a operação normal do indicador de modo..

O uso do {indicate} durante a execução de um macro mostra para o usuário o progresso da execução do macro. Isto é especialmente importante se você usou os comandos {paneloff} e {windowsoff} porque, então, o computador parecerá estar "morto" durante a execução do macro. Usando um indicador de modo apropriado e mudando-o, talvez, durante a execução do macro, deixará o usuário saber que o macro está funcionando..

6. Interagindo “Com o Mundo”

6.1 {?}

O comando ponto de interrogação (?) causa a parada de execução do macro. Você pode digitar tudo que você quiser ou mudar a posição da planilha. Quando você pressiona a tecla Enter o 1-2-3 volta à execução do macro. Este comando permite macros interativos onde o usuário contribui durante a execução dos mesmos.

É importante notar que, pressionando a tecla Enter depois de uma mudança ou durante uma pausa {?} significa que o usuário está alerta. Se você quer que o macro pressione a tecla Enter você tem que incluir um ~ depois do {}. por exemplo, olhe para este macro:

```
/rfc2 ~ {?} ~
```

Neste macro, o 1-2-3 assume a Moeda Corrente de Formato da Região 2 casas decimais, então ele para e deixa o usuário mudar de região para formatar. Ao executar o macro, o usuário usará a tecla de cursor para destacar a região e então pressionar Enter. O último ~ no macro aciona a tecla Enter para completar o destaque da região. Isto se o macro só estivesse assim:

```
/rfc2 ~ {?} ~
```

Neste caso, depois de ampliar o destaque da região e clicar a tecla Enter, o 1-2-3 executa uma parada do macro e o painel de controle ainda estará pedindo a região para formatar. O usuário terá que pressionar a tecla Enter novamente para completar a o destaque da região.

6.2 {acquire}

O comando {acquire} pára o macro e espera o usuário digitar um único toque de tecla. Uma vez que o usuário pressiona uma tecla, o comando {acquire} pára no local e continua com o resto do macro. O {acquire} aparece assim:

```
{acquire local}
```

Local é um endereço de célula ou nome de região onde o 1-2-3 armazenará o toque de tecla. Se o local é uma região nomeada, o 1-2-3 coloca o toque de tecla na célula de topo da região. A tecla pressionada pode ser uma qualquer. O 1-2-3 coloca a representação do macro daquela tecla como um texto. Por exemplo: apertando a tecla *para baixo*- a tecla de seta mostrada pelo comando {acquire} produzirá o texto '{down}.

6.3 {getlabel}, {getnumber}

Estes dois comandos provêm um método para pedir ao usuário que digite algo e aceita a contribuição, armazenando-a em uma célula. {Getlabel} coloca a contribuição como um texto, {getnúmero} coloca isto como um número, se o usuário digitou uma entrada numérica válida.

A sintaxe do comando fica assim:

```
{prompt, getlabel location }
{prompt, getnumber location }
{getnumber "Enter the amount of the check",a50}
{getnumber "Enter the amount of the check",check}
```

Prompt é um caractere que normalmente pede ao usuário que digite algo - -seu nome, a quantia de um cheque, ou outras. **Location** é uma célula ou nome de região. Depois que o usuário digita algo e clica a tecla Enter, o 1-2-3 coloca a digitação deles/delas naquela célula.

Por exemplo:

```
{getnumber "Entre a quantia do check",A50}

{getnumber "Entre a quantia do check", check}
```

O primeiro exemplo exhibe o lembrete *Entre a quantia do cheque* no painel de controle, e então coloca o número digitado pelo usuário na célula A50.

O segundo executa a mesma ação, mas coloca o resultado na célula nomeada **check**.

```
{getlabel "Digita seu nome: ",A50}
```

```
{getlabel "Digita seu nome:, nome}
```

Estes dois exemplos usam o comando {getlabel}, quando a informação pedida é um texto, não uma entrada numérica. Também note que estes dois exemplos usam uma vírgula e um espaço depois do nome. Isso faz o lembrete mais fácil de responder. Por exemplo, se o comando fosse

```
{getlabel "Digita seu nome",A50}
```

o painel de controle exibiria

Digite seu nome

Depois de digitar o nome, coloca-o no painel de controle como

Digite seu nome nameJohnDoe

Note que o primeiro caractere do nome John Doe aparece r melhor, mas aquela ação apresenta um espaço ao nome da célula A50, o que provavelmente não será bem-vindo. Mas o comando:

```
{getlabel "Digita seu nome: ",A50}
```

produz um resultado assim:

Digite seu nome: John Doe

Isto parece muito melhor e é menos confuso para o usuário.

6.4 {menubbranch}, {menucall}

Os comandos {menubbranch} e {menucall} fazem criar seus próprios menus. Estes menus de macro são iguais aos próprios menus do 1-2-3.

Um menu de macro aparece assim:

A B C D

10 Moeda corrente de fmenu Fixo Por cento

11 formato como Formato de Moeda Corrente Formato Fixo como Por cento

12 /rfc2 ~ ~ /rff2 ~ ~ /rfp2 ~ ~

O próprio menu consiste na região B10.. D11. Os comandos que você quer que apareçam no menu estão na linha de topo do mesmo, linha 11 neste exemplo. Este menu, então, exibirá os três comandos de Moedas corrente Fixo, e Por cento.

Os lembretes para cada um dos comandos de menu aparecem diretamente debaixo dos comandos do menu. Debaixo dos lembretes, coloque as instruções de macro que pertencem a cada comando. Para este exemplo simples, o comando de macro formata a célula atual com o formato selecionado.

Cada menu tem um nome de região de forma que o 1-2-3 possa achá-los com facilidade.

O nome de região pode ser qualquer nome de região válido, e você só precisa do nome na célula de topo do menu, célula B10 neste exemplo.

O 1-2-3 reconhece o fim de um menu de macro por uma célula em branco, assim neste exemplo, a célula E10 deve estar em branco. Caso contrário, os conteúdos de E10 seriam considerados parte do menu). Neste exemplo nós digitamos a palavra **fmenu** na célula A10 para fins de documentação, e assim nós pudéssemos usar o comando Gama Nome Textos Direita para nomear célula B10.

Para usar este cardápio, você precisará fazer um macro como este

:

A B

1 /f {fmenu de menubbranch}

O comando de macro {menubbranch} significa mudar para o menu cujo canto esquerdo superior tem o nome de região *fmenu* e mostra o menu. Depois que o usuário fizer uma seleção do

menu, o 1-2-3 executa os comandos de macro associados com aquela seleção de menu. Quando estes comandos de macro terminam, o macro **/f** termina.

O comando `{menucall}` trabalha como o `{menubranh}`, mas depois que o 1-2-3 termina o os comandos de macro para uma seleção de menu, ele retorna para a célula que continha o comando `{menucall}` e continua executando os comandos de macro.

6.5 `{breakon}`, `{breakoff}`

Normalmente, quando executando um macro você queira interrompê-lo pressionando a tecla Control-Break. Mas pode acontecer que quando você quiser desabilitar a ação da tecla Control-Break, para prevenir que alguém termina prematuramente o macro. O comando `{breakoff}` faz isto. Uma vez que um macro encontra o comando `{breakoff}` pressionando a tecla Control-Brak nada faz.. Colocando o comando `{breakon}` em um macro restabelece a operação do Control-Break. Ao término de um macro, o 1-2-3 restabelece, automaticamente, a operação de tecla Control-Break..

Não coloque o `{breakoff}` em um macro até que você esteja seguro de que o macro está operando corretamente. Você poderá querer tirar a habilidade para parar seus próprios macros se eles ficarem fora do controle. Uma vez que eles estejam perfeitos, adicione o comando `{breakoff}`.

7. Movendo e fazendo Looping

7.1 `{branch}`

O comando `{branch}` em um macro transfere a execução do macro para outro local e continua lá o comando do macro de leitura. Exemplos:

```
{branch B45}
{branch step2}
```

O primeiro exemplo salta para a célula B45 e continua o macro. O segundo exemplo usa uma região nomeada para saltar para ela, que é a prática preferida.

Não confunda o comando `{branch}` com o comando `{goto}` que é a representação de macro para função F5, tecla Goto. O comando `{branch}` só afeta onde o 1-2-3 mostra o próximo comando de macro, enquanto o comando `{goto}` move o ponteiro de uma célula para outra célula.

Existe uma grande diferença entre eles.

7.2 `{if}`

O comando de macro `{if}` deixa um macro fazer uma de duas opções, baseado em alguma condição. Este comando trabalha testando uma condição, e então ou lê um comando de macro adicional da mesma célula, ou os lê da célula seguinte. Aqui é um exemplo:

```
A
1 {if ok de b5=0}{branch}
2 {branch notok}
If b5=0 then
branch to ok
else
branch to notok
```

Você pode ver que a sintaxe para o comando `{if}` é `{se condição}`, onde condição é algo que avalia para verdadeiro ou falso. (O Lotus 1-2-3 considera que o valor numérico zero é falso; qualquer outro valor é verdadeiro.) Se a condição for verdadeira, o processador do macro continua a leitura da tecla da mesma célula em que o comando `{if}` está, neste caso o comando `{branch ok}`. Se a condição for falsa, o executor do macro lê os comandos de macro da célula seguinte. Neste caso, ele se move para célula A2 e executa o comando `{branch notok}`.

SINTAXE do comando IF EXIST:

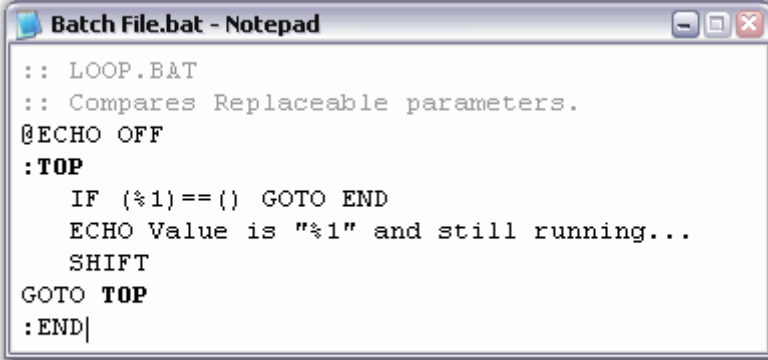
- IF EXIST {filename} {If filename exist, perform this action}

Se a declaração que é testada for VERDADEIRA executa a ação naquela linha. Se a declaração for FALSA continua na próxima linha. A VERDADEIRA declaração de ação é então, essencialmente, ignorada..

Uma declaração de GOTO poderia estar, talvez, presente nesta linha enviando o processamento do grupo de arquivos. O comando IF é de grande valia para testar três tipos de condições que incluem parâmetros substituíveis, ERRORLEVEL de sistema, e isto pode comparar duas linhas..

Lembre-se que vários comandos IF's podem ser usados para resolver um problema mais complexo e criar um algoritmo mais detalhado. Um algoritmo, simplesmente, pode encontrar a solução do problema.

Abaixo, um exemplo que testa parâmetros substituíveis (Note que isso em algumas versões de Windows ou uma configuração diferente de seu lembrete de comando poderia mudar como este exemplo é mostrado. Veja como isto funciona, usando as notações do DOS.):



```

Batch File.bat - Notepad
:: LOOP.BAT
:: Compares Replaceable parameters.
@ECHO OFF
:TOP
  IF (%1)==() GOTO END
  ECHO Value is "%1" and still running...
  SHIFT
GOTO TOP
:END|
  
```

Type it Copy

As primeiras duas linhas são declarações de observação que indicam o nome do programa como também uma descrição breve do que o programa faz. A terceira linha traz comando que ecoa a uma paralisação. Revista quatro marcas fora o programa de substituto para o que um comando de GOTO poderia pular. Revista 5 testes para um parâmetro substituível em branco. Quando %1 contém qualquer informação então informação é colocada no primeiro fixado de parênteses (parâmetro) == () e avalia para

FALSO. Isto é o que nós queremos. Um engano comum é omitir os parênteses, porém nós precisamos deles assim nós não criamos uma volta infinita.

Quando o parâmetro %1 substituível estiver vazio, o teste se torna () == () e adequadamente avalia para RETIFICAR. Esta linha é o código núcleo deste programa. Quando a expressão avalia para RETIFICAR o GOTO END que comando é implementado. Isto termina o programa. Quando a expressão (condição) é FALSO, a próxima linha de código é executada. As sextas exibições de linha que o valor de parâmetro combinou em mensagem,

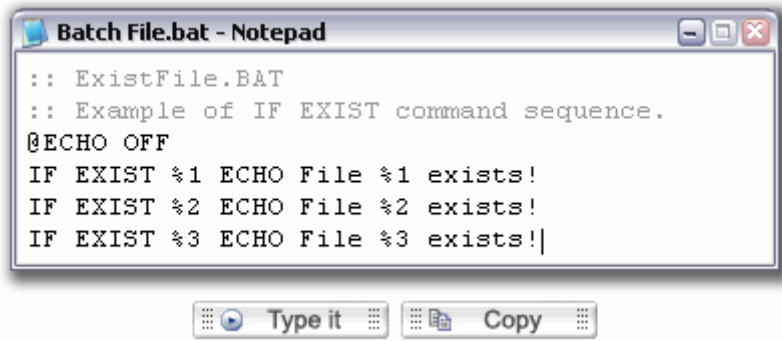
Valor é "algo" e ainda executando... usando o comando de ECO. Aquela próxima linha de código contém o comando CHANGE. Nós sabemos que o comando de CHANGE muda os valores de parâmetros substituíveis acima. A última linha é o substituto-programa de FIM que não tem nada contido dentro de significar o fim do programa.

Nota que o recua não mude seu código e não seja necessário. É mais fácil de ler isto se nós recuarmos tudo dentro do: substituto-programa de TOPO como mostrado.

O comando IF é por si só porém um grande comando que rende grandes possibilidades quando combinado com outros comandos. Como o comando EXIST. O comando EXIST é usado para testar a existência de um arquivo. Isto que usa o esquema de sintaxe seguinte pode ser feito: IF EXIST sintaxe de comando:

{IF EXIST {filename} {Se filename existir, execute esta ação}}

Para este comando trabalhar você deve colocar o nome de arquivo onde o filename em seu local respectivo como mostrado acima seguiu pela ação para levar. Um exemplo que mostra um possível uso:



```

Batch File.bat - Notepad
:: ExistFile.BAT
:: Example of IF EXIST command sequence.
@ECHO OFF
IF EXIST %1 ECHO File %1 exists!
IF EXIST %2 ECHO File %2 exists!
IF EXIST %3 ECHO File %3 exists!|
  
```

As linhas 1 e 2 são declarações de observação que indicam o nome do arquivo de grupo e uma descrição breve do que o arquivo de grupo faz. Revista 3 voltas fora comando ecoar. Linhas 4 por linha 6 testes a existência de qualquer valores está respectivamente nos parâmetros 1 substituíveis, 2, e 3. Este programa é bastante trivial nisso em seu estado atual não há nenhum conteúdo atual que enche

Em muitas linguagens de programação, nós poderíamos escrever este fragmento de macro assim:

A
 Se b5=0 então
 Mude para ok
 outro valor
 mude para a notok

Nós poderíamos chamar o comando {branch ok} a cláusula de "então", e o comando {branch notok} a cláusula de "outro valor".

Tenha cuidado com o comando {if}. Lembre-se que depois do 1-2-3 executar a cláusula "outro valor" do macro, ele passará para a próxima célula e executará a cláusula "outro valor", a menos que a cláusula "de então" contenha um comando *branching* ou *quiting* (muda ou sai).

7.3 {quit}

O comando **{quit}** indica o fim de um macro. Não é necessário usar este comando para terminar um macro, porque uma célula em branco também o termina. Mas é uma boa idéia usar o comando **{quit}** como marca do fim de um macro, formalmente. Também, se por alguma razão a célula em branco que termina um macro ganhar conteúdos de algum tipo, esses conteúdos são considerados parte do macro.

7.4 {for}

O comando de macro **{for}** faz executar um macro repetidamente. Este comando, então, provê uma capacidade de *looping*. A sintaxe do {for} é assim:

{for counter-location,start-número,stop-número,step-número,starting-location}

Um exemplo do comando {for} pode parecer:

```
{for A50,1,10,1,printmac}
```

Starting location é um nome de região que marca o começo do macro que você gostaria de executar repetidamente. Este macro não precisa ser nomeado com a convenção *backward* o / - única letra. O *counter location* é um endereço de célula ou célula nomeada que o comando {for} usará para colocar um valor do contador.

O comando `{for}` trabalha assim, usando o exemplo anterior:

:

1) Na primeira vez que o 1-2-3 encontra o comando `{for}` coloca o **start número** no **counter location**.

2) O 1-2-3 compara o valor de **counter location** com o **stop value**. Se o **counter location** for menor ou igual ao **stop value** o 1-2-3 executa o macro nomeado pela **start location**. Se o **counter location** for maior do que o **stop value** o 1-2-3 finda o comando `{for}` e procede para executar os comandos de macro da célula abaixo do comando `{for}`.

3) Depois que o macro nomeou pelo **starting-location** a conclusão, o 1-2-3 soma o **step-number** com o **counter-location**.

4) Repete o passo 2.

No nosso exemplo, durante passo 1, o 1-2-3 coloca o valor 1 na célula A50, o **counter location**. No passo 2, ele compara 1 a 10. Desde que 1 é menor do que 10, o 1-2-3 executa o macro nomeado por **printmac**. Então, o 1-2-3 soma o **step-número** de 1 à célula A50, assim se torna 2 e o 1-2-3 que roda o **printmac** novamente, porque 2 ainda é menor ou igual a 10. Eventualmente célula A50 cresce para 11, e desde que 11 não é menor ou igual a 10, o comando `{for}` é encerrado e o 1-2-3 se move para a próxima célula procurando por mais comandos de macro.

É possível que o comando de macro nomeado pelo comando `{for}` nunca seja executado. Um comando como: `{for A50,11,10,1,printmac}`, onde o **start-número** é maior que o **stop-número** nunca executa **printmac**. Um comando `{for a50,11,10,0,printmac}` roda até que seu computador se **explode**. Porque o **step-value** é **zero**, o **couter-location** nunca excede o **stop-número**, assim o comando `{for}` nunca pára.

Durante o comando `{for}`, você pode fazer referência ao **number-location** em uma fórmula ou outro comando. Isto provê um mecanismo **à mão** para contar ou numerar tarefas.

7.5 `{forbreak}`

O comando `{forbreak}`, quando usado em um macro que é executado sob o controle do comando `{for}`, o processador do macro executa os comandos na célula que segue o comando `{for}`, da mesma maneira que o comando `{for}` terminaria normalmente.

7.6 Subrotinas de Macro

Se você quer que um macro execute um conjunto de toques de tecla (ou até mesmo um conjunto semelhante de batidas de tecla), você poderá escrever uma subrotina de dados de macro para executar estes toques de tecla. A subrotina de dados de macro tem um nome, e em vez de escrever os toques de tecla no macro, você pode fazer uma ligação à subrotina de dados.

Suponha você precisa digitar suas várias vezes o seu nome durante um macro. Você poderia escrever um macro que se parece:

```
A B
10 myname Bob Weeks ~
11 {return}
```

Note que a seqüência de dados começa com uma região nomeada, neste caso o myname de nome de região para célula B10 como indicado pela convenção que nomeia as regiões. Ao término da seqüência de dados, use o comando de macro `{return}` para indicar o fim da subrotina. Para usar esta subrotina, coloque este comando em um macro:

```
{myname}
```

Quando o 1-2-3 encontra o comando de macro `{myname}`, transferirá o controle à célula myname de nome de região e executa o macro da tecla lá. Quando o 1-2-3 encontra o comando `{return}` na subrotina, transfere o controle de volta para o comando de macro `{myname}` de chamada e continua executando os comandos adicionais.

Você também pode passar informação à subrotina de macro pelo uso de argumentos.

Por exemplo, suponha que você precise computar a área de um círculo várias vezes durante um macro. Esta é uma tarefa boa para uma seqüência de dados com um argumento, assim escrevamos:

```
A B
10 área {defina radius:value}
11 +@pi*radius^2
12 {retorno}
13 rádio
```

A primeira linha da subrotina de área estabelece a existência de um argumento para o macro. Neste caso, o argumento é um valor (em vez de um texto) e, ao chamar a subrotina, o 1-2-3 colocará o argumento na célula com o *raio* como nome de região. A célula B11 na subrotina digita a fórmula para calcular a área do círculo, enquanto usando o argumento de raio. A célula B12. O comando {return} indica o fim da subrotina. A célula A13, quando usada como comando Região Nome Textos Direito, move para cima da célula B13 com o *raio* como de nome de região.

Para usar esta subrotina, coloque este comando em um macro:

```
{area 10}
```

Quando o 1-2-3 executa esta declaração, o argumento do comando de chamada (o número 10) é colocado na célula com o *raio* como nome de região, por causa da ação do comando {define} ..

Então, a subrotina de macro executa:

O comando {define} pode especificar dois tipos de argumentos--um valor ou um texto, um argumento de valor, um número, endereço de cela, ou fórmula (que pode ser um valor numérico ou fórmula de texto). Omitindo o valor ou: sufixo de texto significa o argumento é um texto.

8. Manipulando Dados

8.1 {blank}

O comando de macro **{blank}** apaga os conteúdos de uma célula ou região. Típico {blank} se mostra assim:

```
:
{blank A50 }
{blank A50 .. B60}
{blank outrange}
```

O primeiro comando apaga uma única célula, o segundo apaga uma região de células, e o terceiro apaga uma região nomeada que poderia ser uma única célula ou um grupo de células.

8.2 {let}

O comando **{let}** faz um macro colocar um valor em uma célula. A sintaxe do comando é

```
{let location,número}
{let location,string}
```

Location é um endereço de célula ou região nomeada (se a região nomeada cobre mais que uma única célula, o 1-2-3 usa a célula de topo). Número e texto são expressões que avaliam um número ou texto. O comando {let} coloca o comando de número ou nomeia um local.

```
{let A50,100}
{let title,Amount Devido}
{let título, "Quantia Devido, Líquido"}
A B
10 area {define radius:value} *{defina o valor do raio}*
11 +@pi*radius^2
```

12 {return}
13 raio

A primeira linha da subrotina de área estabelece a existência de um argumento para o macro. Neste caso, o argumento é um valor (em vez de um texto) e ao chamar a subrotina, o 1-2-3 colocará o argumento na célula com o *raio* como nome de região. A célula B11 na subrotina digita a fórmula para calcular a área do círculo, usando o argumento de raio. A célula B12, o comando {return}, indica o fim da subrotinas. A célula A13, quando usada com o comando Nome de Região Textos Direita, escreve na célula B13 o rádio como nome da região.

Para usar esta rotina, coloque este comando em um macro:

{area 10}

Quando o 1-2-3 executa esta subrotina, o argumento no comando de chamada (o número 10) é colocado na célula com o *raio* como de nome de região, por causa da ação do comando {define}. Então, a subrotina de macro o executa.

O comando {define} pode especificar dois tipos de argumentos--um valor ou um texto. Um argumento de valor é um número, endereço de célula, ou fórmula (que pode ser um valor numérico ou fórmula de texto). Omitindo o: valor ou: sufixo significa o argumento é um texto.

9. Considerações de Cálculo

Você pode ter dificuldades com comandos como {getlabel} e {getnúmero} devido ao modo que o 1-2-3 atualiza a tela da planilha e calcula a planilha. Por exemplo, você poderia usar {getlabel} para colocar dados em uma célula, mas quando o macro terminar, os dados não aparecem na célula onde deveriam estar. Eventualmente poderão estar lá, mas poderia não estar, até que você precise deles.

A dificuldade surge porque há dois fatores que controlam a planilha do 1-2-3 que atualizam e cálculo. Planilha *atualizando* está mostrando o resultado de um comando de macro como {getlabel}. *Calculando* está atualizando os valores de fórmulas. Estas duas ações são diferentes.

Se o 1-2-3 atualiza a planilha depois de um comando como {getlabel} depende em se você usa um comando ~ depois do comando. Se você o faz, o 1-2-3 atualizar a planilha. Se você não inclui o ~, o 1-2-3 não atualiza a planilha. Um ~ é suficiente para atualizar a planilha inteira, assim você poderia usar vários comandos {getlabel} com um só ~ depois do último comando.

Se o modo de cálculo é fixado para automático, um ~ em um macro também faz com que o 1-2-3 recalcule a planilha inteira--possivelmente uma ação muito demorada. Com cálculo fixado em manual, o ~ causa só uma atualização da planilha, sem cálculo de fórmula. Com o cálculo fixado como manual, use {calc} em um macro para recalcular fórmulas..

Estes comandos de macro são, deste modo, afetados pela inclusão ou omissão do ~ {blank}, {getlabel}, {getnúmero}, e {let}. Outros comandos de macro do 1-2-3 requerem um {calc} para atualizar a planilha.

10. Exemplos de Macros

10.1 Digitando Macros.

Uma tarefa comum para um macro é executar a tarefa de digitação, como seu nome e endereço. Aqui é um macro de exemplo que digita um nome e endereço:

A B
1 \ n Bob Weeks ~
2 {abaixo} {down}
3 4455 N. Rockwell
4 {abaixo}
5 Chicago, IL 60625 ~
6

Explicação:

A1 Esta célula contém o nome de região para a célula B1. Depois de digitar isto (lembre-se de digitar o prefixo de texto ' antes do \ n), use a Região Nome Texto-Direito comando com a região A1.. A1.

B1 A célula inicial do macro. Esta célula digita o nome Bob Weeks na célula . atual e completa a entrada de célula devido ao caractere ~ que representa o . macro que digita a tecla Enter Se nós não usássemos o ~, nunca completariamos a entrada de texto, e o resto do macro seria interpretado como digitação adicional na célula atual, em vez de ser reconhecido como comando. , de macro.

B2 Move o ponteiro de célula a uma linha até prepare para digitar a próxima linha do endereço.

B3 Digita o endereço da rua. Isto é um pouco enganador Se você vai . para a célula B3 e digita 4455 N. Rockwell e clica Enter, você cometerá um . erro. Isso é porque o 1-2-3 interpreta uma célula que começa com um dígito . como uma entrada numérica, e você não pode ter espaços, N's, e assim . sucessivamente em um número. A maioria dos usuários do 1-2-3 conhecem . isto e começam uma entrada de célula assim com um prefixo de texto, digo, . apóstrofe (') ('). Se você começa esta célula de comando de macro com uma única apóstrofe (') esta célula como um texto em lugar de um número. Quando o processador de macro processa esta célula, porém, ele elimina a apóstrofe (') inicial e digita 4455 N. Rockwell sem um prefixo de texto na célula atual que produz um erro. Assim digite duas apóstrofe (')s nesta célula--um para o prefixo de texto para a célula B3, e outra para a célula em que o macro digita.

B4 Digita a tecla de seta para baixo move à próxima linha para baixo. Note .que a célula B3 não terminou com o ~ representando digitar a tecla Enter.. . isto está certo neste caso, como pressionando a tecla de seta para baixo . completa a entrada de texto prévia como também movendo-a. Você pode ver . que a digitação do macro segue as mesmas regras como na datilografia comum.

B5 Digita a cidade, estado, e CEP, e então pressiona a tecla Enter pelo . caractere ~ para completar a entrada.

B6 desde que esta célula está em branco, significa o fim do macro. É uma boa idéia usar o comando {let} para identificar o fim de um macro explicitamente.

10.2 Somando uma Coluna

Este macro somará uma coluna de números. Faz uso amplo da navegação da tecla de cursor, inclusive a tecla de End (Fim). Para este macro trabalhar corretamente, o ponteiro de célula deve estar em qualquer lugar na coluna de números a somar, com exceção da figura de fundo.

Aqui é uma visão da tela de um macro para realizar esta tarefa. Este macro segue uma convenção comum para a digitação de macros. Na coluna A (ou qualquer coluna à esquerda dos comandos de macro), digita textos que provêem nomes de região para as células à sua direita. Na coluna B, digite os comandos de macro. Em uma coluna à direita, escreva um comentário ou nota para cada célula do macro. Esta documentação, (ambos o nome de regiões e os comentários) é importante. O idioma de macro do 1-2-3 pode estar confundindo quando você está tentando entender o que o macro que você escreveu no mês passado faz, e qualquer nota ou comentários que você escreva ajudarão a esclarecer os assuntos para você ou outra pessoa.

A B C D E F

1 \c {end}{down} Mova para terminar a coluna

2 {down} move uma mais célula para baixo.

3 \ - ~ faz uma célula cheio de sinais de menos

4 {down} Movem para baixo uma mais célula

5 @ soma (datilografia de Começo a função de soma
 6 {up}{end}{up} Mova para terminar a coluna
 7. Ancore ponteiro de cela com período
 8 {end}{down} Mova para terminar de coluna
 9 {para cima} Nós movemos uma célula muito distante.
 10) ~ função de soma de fim e pressiona a tecla Enter
 11 {let} Deixa o macro

Visão do macro do Cambridge Planilha eletrônica Analista:

B1 '{end}{down}
 B2 '{abaixo}
 B3 '\ - ~
 B4 '{down}
 B5 '@ soma (
 B6 '{up}{end}{up}
 B7 '
 B8 '{end}{down}
 B9 '{ up}
 B10 ')' ~
 B11 '{let}

Explicação:

A1 O nome do macro, para uso com os comandos Textos de Nome de Região comando Direito.

B1 O começo do macro. Usa as teclas de seta *End* e *Down* para mover ao número do fundo na coluna.

B2 Move para baixo uma linha adicional, para onde a tecla *End* moveu.

B3 Digita o / e um sinal de menos e completa a entrada com o ~ representando a tecla Enter. Ao digitar este macro, escreva um prefixo de texto (geralmente a apóstrofe (')) primeiro. Se você não o fizer, o 1-2-3 encherá a célula B3 de sinais de menos. Ao contrário, esta célula produz uma linha de sinais de menos no fundo da coluna de números.

B4 Move para baixo mais uma linha em preparação para digitar a @ função de soma.

B5 Começa a @ função de soma pelo parêntese de abertura. Ao digitar esta célula no macro, use um prefixo de texto, ou você estará digitando uma fórmula em vez de um texto que o processador de macro do 1-2-3 necessita..

B6 Use o a tecla de seta *para cima* {up}. A tecla de seta promove uma linha, então o *End* a tecla de seta para cima, movendo para o topo da coluna..Nós precisamos usar um único {up}, de modo que a tecla de End encontra o topo da coluna.

B7 Esta célula, um único período, ancora o ponteiro de célula ao topo da coluna.

B8 Usa as teclas End e Down para mover para o fundo da coluna

B9 As teclas de End e Down da célula B8 nos move para uma linha distante devido à célula de sinais de menos (-) que nós entramos antes. Assim, use o comando {up} para mover para cima uma única linha. .

B10 Digite o parêntese da direita para completar a função @ de soma, e digite a tecla Enter para completar a digitação da fórmula.

B11 O fim do macro.

Como você pode ver, os macros do 1-2-3 podem ser bastante gerais se você os planejar corretamente. Para escrever este macro, porém, você precisa estar muito familiarizado com o modo com que a tecla de Fim opera- e quando você estiver digitando o macro, você não consegue ver os efeitos de seus comandos de macro. Muitas pessoas traduzem o macro escrevem embaixo as teclas que eles pressionaram, desenvolvendo uma receita para criar o macro.

10.3 Dados

Um uso muito útil para um macro é tornar a entrada de dados mais fácil. Os dados no 1-2-3 devem ser entrados usando a @ função de dados assim:

@ date(year,month,day)

Para entrar em 1 de setembro de 1987, você digitaria @ date(87,9,1). Isto representa muita digitação e força você entrar os elementos do calendário uma ordem pouco conhecida. Também, depois de entrar a data muitos usuários formatarão a data usando um dos formatos de data. Nós desenvolveremos um macro que o alivia de digitar o @ função de data, e permite que você entre os elementos de data em uma ordem mais familiar, e formata a data..

Aqui são os passos nós precisamos implementar.

1) peça o mês, dia e ano e os armazene em três células. Nós podemos lhes pedir em qualquer ordem, assim nós pederemos pelo mês atual, dia e ano, nesta ordem. O macro usa o comando {getnúmero} para esta tarefa.

2) digite a @ função de datas que usa estes introduz valores.

3) para economizar memória e aumentar a velocidade do cálculo da planilha coverta a a @função data em um simples número

4) formata a célula ela com um formato de data.

Aqui é uma visão de tela de um macro realizar estas tarefas:

```
A B C D E F G H
1 \ d {paneloff}{windowsoff} Paine! & janelas fora
2 {getnúmero "Entram em Mês: " Adquira o mês para *zmonth*
3 {getnúmero "Entram em Dia: ", O dia para *zday*
4 {getnúmero "Entram em Ano: ", O ano para *zyear*
5 @ date(zyear,zmonth,zday) ~ Digite a função de data
6 {edit}{calc} ~ Use Edite e Calc para "númeroize"
7 / rfd4 ~ Formato como Data 4
8 / rezmonth ~ Apague zmonth
9 / rezday ~ Apague zday
10 / rezyear ~ Apague zyear
11 {panelon}{windowson} Paine! & janelas em
12 {let} Deixe o macro
13 Célula de zmonth durante mês
14 Célula de zday durante dia
15 Célula de zyear durante ano
Visão do macro do Cambridge Planilha eletrônica Analista:
B1 '{paneloff}{windowsoff}
B2 '{getnúmero "Entram em Mês: ", zmonth}
B3 '{getnúmero "Entram em Dia: ", zday}
B4 '{getnúmero "Entram em Ano: ", zyear}
B5 '@ date(zyear,zmonth,zday) ~
B6 '{edit}{calc} ~
B7 / rfd4 ~
B8 / rezmonth ~
B9 / rezday ~
B10 / rezyear ~
```

B11 '{panelon}{windowson}

B12 '{deixe}

Explicação:

B1 Esta célula tem o nome de região \d. Para nomear esta célula, digite \d na célula A1 e usa o comando Nome da regiãoLabels Right. Os comandos de macro nesta célula, {paneloff e {windowsoff} suprimem a visão do movimento painel de controle e da janela durante a execução do macro..

B2 Usa o comando {getnúmero} para pedir o número para o mes, O 1-2-3 o número digitado pelo usuário na célula nomeada *zmonth*. Num macro como este, que deve ser usado em diferentes planilhas, é bom usar um nome de região pouco comum como *zmonth*. Usando o nome de região month, um número comum para uma região, pode nos levar a conflitos com outros nomes de região na planilha.

B3 pede pelo dia e o coloca em *zday*.

B4 pede pelo ano e o coloca em *zyear*.

B5 a função @data que referencia as tres células *zyear, zmonth e zdahe. Esta célula de macro digita a formula na célula atual. Quando digita esta célula, , comece com um prefixo de texto. De outro modo você estará digitando a fórmula na célula B5, não o texto. Também, esteja certo de completar a entrada de dados com o comando ~ para completar a entrada da fórmula na célula atual., porque o o macro nunca completará a fórmula..

B6 Esta célula de macro converte a formula @date (zyear, zmonth, zda) para o número que a fórmula representa. Se não fizermos isto, o 1-2-3 sempre conservará a formula na célula e, na próxima vez que você executar o macro, as células nomeadas tomará novos valores, e as datas mudarão. Existem duas maneiras de fazer isto. Você pode editar uma célula, pressionar tecla *calculate* e clicar Enter, ou ou você pode usar o comando de Valor de Região.

B7 Coloca o comando RegiãoFormat Date 4 e clicar a tecla Enter para formatar a célula atual. Você pode mudar o 4 para outros números de formatos de data.

B8 Use o comando *Regiãoerase* para apagar a célula *zmonth*. Este passo não é absolutamente necessário, mas é mais pra completar do que por outro motivo..

B9 Apaga a célula *zday*.

B10 Retorna o controle dos movimentos do painel de controle e da janela para para o modo *on* Este passo não é absolutamente necessário, porque quando o macro termina na próxima célula o 1-2-3 retorna estes movimentos para *on*, automaticamente.

B12 O comando {quit} significa o fim do macro. Isato não é sempre necessário pois uma célula em branco também termina um macro. Mas é uma boa idéia usar o comando {quit}, pois ele, formalmente, indica o fim do macro. a good idea to use the {quit} command, as it formally indicates the end of a macro.

A13 Esta célula contém o texto *zday*. Usando-se o comando Nome da regiãoLabels Right, nós podemos usá-lo para nomear a célula para a direita (B13) *zday*.Você pode ver o valor da documentação desta técnica—o nome de uma célula está justamente à direita de uma célula nomeada..

A14 Nomeia a célula B14 *zmonth*..

A15 Nomeia a célula B15 *zyear*..

Explicação::

B1 Esta célula tem o nome de região \d, Para nomear esta célula, digita \d na célula A1 e usa o comando de * Textos de Nome de Célula Certo..O macro comanda esta célula, {paneloff} e {windowsoff}, s exibem o painel de controle e movimento de janela durante o macro.

B2 Use o comando {getnumber} para pedir o número do mês. O 1-2-3 colocará o número digitado pelo usuário na célula que é nomeada *zmonth*. Um macro assim poderia ser usado em várias planilhas diferentes e é bom para usar nomes de região incomuns como *zmonth*. Usando o nome de região mês, um nome comum para uma região poderia conduzir a conflitos com outros nomes de região nomeadas na planilha.

B3 Pede o dia e o coloca em*zday*.

B4 Pede o ano e o coloca em *zyear*..

B5 A @ função de dados que referência os três *zyear*, *zmonth*, e *zday*. Esta célula de macro digita a fórmula na célula atual. Ao digitar esta célula, comece com um prefixo de texto. Caso contrário, você estará digitando a fórmula na célula B5, não o texto. Também, complete a entrada com o ~ . Caso contrário, o processador do macro digitará os outros toques de tecla do macro na célula atual, porque o macro nunca completa a fórmula.

B6 Esta célula do macro converte a fórmula @ date(zyear,zmonth,zday) para o número que a fórmula representa. Se nós não fizermos isto, o 1-2-3 sempre reterá a fórmula na célula, e da próxima vez você executa este macro, as células nomeadas assumirão valores novos, e as datas mudarão. Há dois modos para fazer isto. Você pode editar uma cela, aperte o calcule chave, e clica Enter, ou você pode usar o comando de Valor de Região.

B7 Digita o comando de Formato de Região Data 4 e pressiona a tecla Enter para formatar a célula atual. Você pode mudar o 4 a outros números de formato de dados para apagar o *zmonth* da célula.. Este passo não é absolutamente necessário, mas é mais para perfeição que qualquer outra coisa.

B9 Apaga o zday de cela.

B10 Apaga o zyear de cela.

B11 Retorna o movimento do painel de controle e da janela. Este passo não é absolutamente necessário, porque quando o macro termina na próxima célula, o 1-2-3 volta o painel de controle e janelas automaticamente.

B12 O comando {let} significa o fim de um macro. Este não é sempre um comando necessário, pois uma célula em branco também termina um macro. Mas é uma idéia boa para usar o comando {let}, como indicando formalmente o fim de um macro..

A13 Esta célula contém o texto *zday*. Usando o comando *Região Gama Nome Textos Direito*, nós podemos usar isto para nomear a célula à direita (B13) *zday*. Você pode ver a documentação avaliar esta técnica--o nome de uma célula de topo e à esquerda de cada célula nomeada.

A14 Nome da célula *zmonth* de B14.

A15 Nome da célula *zyear* de B15.

Ao criar este macro, você pode usar um comando *Região Nome Textos Direito* com a região A1.. A15 para dar todos os nomes de região necessários para este macro. Esta é uma vantagem especial deste macro, desde que a sua natureza geral e aplicabilidade em muitas planilhas, você usará o comando de *Associação de Arquivo* provavelmente para trazer estas planilhas de *intomany* de macro. Desde que o 1-2-3 não traz os nomes de região da planilha entrante, enquanto documentando nomes de região é até mesmo mais importante.

Uma coisa que este macro não faz é checar a validade da data. Se o usuário digita 13 para o mês, a @ função de data devolverá o valor de erro ERR.

Uma variação deste macro entra a data de hoje pressionando uma tecla..

A B C D E F G H

1 \ t {paneloff}{windowsoff} Painel & janelas fora

2 @ int(@now) ~ Tipo o @ int e @ agora funções

3 {edit}{calc} ~ Use Edite e Calc para "númeroize"

4 / rfd4 ~ Formato como Data 4

5 {panelon}{windowson} Painel & janelas em

6 {deixe} Deixe o macro

B1 '{paneloff}{windowsoff}

B2 '@ int(now) ~

B3 '{edit}{calc} ~

B4 '/ rfd4 ~

B5 '{panelon}{windowson}

B6 '{deixe}

Este macro digita a fórmula @ int(now) produzir a data de hoje, então "numeroizes" e formata isto.

10.4 - Exemplos de Menus

Este macro que é bastante simples e ilustra o uso de um menu dentro de um macro. O macro apresenta um menu de possíveis formatos (Moeda corrente, Vírgula, Fixo, etc.) para o usuário escolher os locais) para célula atual.

Vista da tela de um macro:

A B C D E F G H

1 \f {paneloff}{windowsoff} Turn panels & windows off

2 {menucall fmenu} Call the menu fmenu

3 {panelon}{windowson} Panel & windows on

4 {quit} Quit the macro

5

6 fmenu Currency , Fixed Percent Scientific

7 Format asFormat asFormat asFormat asFormat as Scientific,

8 /rfc2~~ /rf,2~~ /rff2~~ /rfp2~~ /rfs2~~

9 {return} {return} {return} {return} {return}

A vista the do macro produzido pela Cambridge Spreadsheet Analyst.

MACRO: <\F> LISTING

B1 '{paneloff}{windowsoff}

B2 '{menucall fmenu}

B3 '{panelon}{windowson}

B4 '{quit}

MACRO: <FMENU> LISTING

MENU LINE: 'Currency ', 'Fixed 'Percent 'Scientific

MENU OPTION: 'Currency

MENU PROMPT: 'Format as Currency, 2 decimal places

B8 '/rfc2~~

B9 '{return}

MENU OPTION: ',

MENU PROMPT: 'Format as Comma, 2 decimal places

C8 '/rf,2~~

C9 '{return}

MENU OPTION: 'Fixed

MENU PROMPT: 'Format as Fixed, 2 decimal places

D8 '/rff2~~

D9 '{return}

MENU OPTION: 'Percent

MENU PROMPT: 'Format as Percent, 2 decimal places

E8 '/rfp2~~

E9 '{return}

MENU OPTION: 'Scientific

MENU PROMPT: 'Format as Scientific, 2 decimal places

F8 '/rfs2~~

F9 '{return}

Explicação

A1 Nome da Célula do macro quando usada com o comando *Nome da regiãoLabels Right*.

B1 A célula inicial do macro. Muda o painel de controle e atualiza e desativa a atualização da tela.

B2 O comando {menucall fmenu} chama o menu cujo nome da célula de topo à esquerda é *fmenu*. Depois deste comando o controle passa para o menu e o macro que o usuário selecionou. Uma vez que o comando é {menucall}, completando o menu de macro, o comando passa para a célula seguinte (célula B3). Se nós usamos o comando {menubracnh, este macro pára depois de completar o menu de macro..

B3 Depois de completado o menu de macro, o controle retorna aqui. Esta célula volta o painel de controle e a janela para *on*, o que não é, realmente necessário..

B4 Não é absolutamente necessário, mas é uma boa idéia usar o comando `{quit}` para indicar o fim de um macro.

A6 Esta célula quando usada com o comando `*Nome da regiãoName Labels right*`, nomeia a célula B6 de `*fmenu*`. Esta célula é a célula de topo esquerdo do menu. .

B6 A célula de de topo esquerdo do menu. Esta célula é a primeira escolha do menu, a palavra `Currency`..

C6 Atravez de F6 Outros comandos para o menu..

B7 A descrição (Formate como moeda, com duas casas decimais) para o comando de menu na célula B6..

C7 Atravez de F7 Outras descrições de comandos na célula C6 até F6..

B8 O macro para o comando de menu para a moeda. Este macro particular ocupa apenas uma célula, mas o macro poderia continuar o quanto for necessário.

C8 Atravez de F8 Macros para os outros comandos de menu..

B9 O comando de macro `{return}` signifoca o fim do macro para a coluna B e este processador de comando deve retornar para a célula seguinte o comando `{menucall fmenu}` e continua executando os comandos do macro.

C9 a F9 Outros comandos de macro `{return}` para os outros menus de macro. macros.

10.5 Largura de Colunas

Este macro pede ao usuário para digitar um número para a largura da coluna atual, e então passa para o comando, que coloca a largura.

O truque com este macro é que nós necessitamos fazer um macro digitar um número para cada vez que você executa o macro. Nós fazemos isto nomeando uma uma das células do macro para usar o comando `{getlabel}`. O número (que é realmente um texto—uma célula de macro tem que ser um texto), que, então, passa a ser largura parte do macro, e o processador do macro digita este número.

A tela deste macro:

A B C

```
1 \w {paneloff}{windowsoff} Panel and windows off
2 {getlabel "Enter width for column: ",width} Get the width
3 /wcs Start /wcs
4 width Número for width
5 ~ Enter key to complete
6 {panelon}{windowson} Panel and windows on
```

A tela do macro produzido pela Cambridge Spreadsheet Analyst.:

```
MACRO: <\W> LISTING
B1 '{paneloff}{windowsoff}
B2 '{getlabel "Enter width for column: ",width}
B3 '/wcs
B4
B5 '~
B6 '{panelon}{windowson}
```

Explicação:

A1 O nome do macro, `\w`.

B1 fecha o painel de controle e a janela

B2 Pede ao usuário a largura da coluna e coloca o resultado na célula nomeada (Célula B4). Note que o uso do `{getlabel}` para produzir um texto numa célula, embora nós estivéssemos pedindo por um número. Isto porque as células que contêm os comandos do macro têm que ser textos, e a largura da célula nomeada é uma célula de comando de macro.

B3 Introduce o comando `/Worksheet Column Set-Width command`.

A4 Introduce o comando Nomeia a célula B4 co o comando `Nome da regiãoLabels Right`..

B4 Esta célula texto contém a reeposta que o comando `{getlable}` obtem a para pede feita pela célula B2. O processador do macro digita qualquer número aqui como a resposta à pede feita sobre a largura da coluna.

B5 Pressiona a tecla `Enter` para completar a resposta para a pede feita sobre a largura da coluna.

Este macro mostra que o truque que você fez para o seu macro funcionar. O Lótus 1-2-3 não mostra para você como realizar um truque como este--você tem que usar a sua imaginação para descobrir com resolver problemas

10.6 Imprimindo com o {for}

Este macro imprime múltiplas cópias e de uma planilha, usando o comando {for} para controlar a impressão. Este macro consiste de dois macros. O primeiro pede pelo número de cópias, a região a ser impressa, e processa um comando {for}. O segundo macro imprime uma cópia da folha, e é processado tantas vezes ditado pelo comando {for}. /este macro rquer que a região da folha a ser impressa seja uma região nomeada.

A tela deste macro fic assim::

```
A B C D E F G H
1 \p {paneloff}{windowsoff} Panel & windows off
2 {getlabel "Enter regiãooto pGet print regiãooto prntrange
3 {getnúmero "Número of copieGet número of copies to copies
4 {for counter,1,copies,1,dopThe for statement to run doprint
5 {quit} Quit the macro
6
7 doprint /ppr The print printer regiãoocommand
8 prntrregiãoAr região para imprimir
9 ~ The enter key
10 agpq Align, Go, Page advance, quit
11 {return} The end of the doprint macro
12
13 copies Cell of number of copies
14 counter Cell for counter value.
```

A vista do macro produzido pela Cambridge Spreadsheet Analyst:

MACRO: <DOPRINT> LISTING

B7 '/ppr

B8

B9 '~

B10 'agpq

B11 '{return}

MACRO: <\P> LISTING

B1 '{paneloff}{windowsoff}

B2 '{getlabel "Enter região para imprimir: ",prntrange}

B3 '{getnúmero "Número of copies: ",copies}

B4 '{for counter,1,copies,1,doprint}

B5 '{quit}

Explicação:

A1 O nome da célula B1, para iniciar o macro \p..

B1 O início do macro \p. Muda para *off* a atualização do painel de controle e da janela.

B2 pede pela região a ser impressa, e coloca a resposta na célula de região nomeada *prntrange* (célula B8).

B3 Pede pelo número de cópias a serem impressas, e coloca o número na célula nomeada *copies* (célula B13). Esta célula fica sendo o *stop número* para o comando {for}.

B4 O comando {for} que faz o macro trabalhar. Este comando usa a célula da região nominada *counter* (célula B14) como um contador, 1 como o *start-número*, a célula nomeada *copies* como o *stop-número*, 1 como o *step-número*, e célula nomeada *doprint* como o *staring-liocation*.

B5 O fim do macro \p.

O macro \p executa o macro *doprint* contudo, quantas vezes desejadas pelo usuário.

Vamos examinar o macro *doprint* para ver como ele trabalha:

B7 Ativa o comando /Print Printer Range.

B8 Esta célula, a qual tem a região nomeada *prntrange*, contém o resultado do comando {getlabel} na célula B2. Assim, o executor do macro digita o nome da região neste ponto.

B9 Pressiona a tecla Enter para completar a entrada da *printrange* da célula B8..

B10 Ativa o comando *Align*, o comando *Go* para imprimir a região, o comando *Page* para avançar para o topo da próxima página, e o comando */quit* para parar os menus de e retornar para o modo *READY*.

B11 O comando {return} significa o fim do macro *doprint* {return} e volta o controle para o comando {for} na célula B4. Neste momento o comando {for} decide se executará o *doprint* novamente.

Como você pode a função do macro *doprint* é imprimir uma única cópia de uma região nomeada . O comando {for} do macro \p decide quantas vezes deve executar o macro *doprint*.

10.7 A combinação de arquivos e a Subrotina de Macro.

Este macro usa o comando *File Combine* para incorporar os dados de 3 planilhas departamentais (depta, depth, and deptc) na planilha principal da companhia . Em cada uma das planilhas departamentais , a linha dos dados de *net income* é uma região nomeada *netincome*. Nós quremos colocar os dados da planilha da *depta*, começando na célula B5, *debtb* na B6 , e deptc* na B7. A linha 9 desta planilha contém as formulas *@sum* para acionar os dados de vendas. Existe uma região nomeada *dados* que cobrem as linhas 5, 6 e 7, começando na coluna B.

Este macro usa a subrotina chamada *combine* para executar o comando File Combine para cada um dos 3 departamentos..

A vista desta planilha e macros.:

```
A B C D E F G
1 All Departments
2
3 Jan Feb Mar Apr May Jun
4 -----
5 Department A
6 Department B
7 Department C
8 -----
9 Total $0.00 $0.00 $0.00 $0.00 $0.00 $0.00
10
11 \c {indicate WAIT} Mode indicator to WAIT
12 /wgrm Set calc. to manual for speed
13 /redata~ Use /RegiãoErase to erase
14 {goto}data~ Position cell pointer
15 {combine depta} File Combine from DeptA
16 {down} Position cell pointer
17 {combine deptb} File Combine from DeptB
18 {down} Position cell pointer
19 {combine deptc} File Combine from DeptC
20 {goto}data~ Position cell pointer
21 {down} for formatting
22 /rf,2~{end}{right}~ Format row as comma, 2 decimals
23 {down} Position cell pointer
24 /rf,2~{end}{right}~ Format row as comma, 2 decimals
25 /wgra Set calc. back to automatic
26 {calc} Press the F9 Recalculation key
27 {indicate} Restore the mode indicator
28 {quit} Quit the macro
29
30 combine {define infile,string} Location for incoming file name
31 /fcan File Combine Add Named-range
32 netincome~ The nome da região to combine
33 infile The worksheet to combine
34 ~ A tecla *Enter*.
35 {return} O fim da subrotina
```

A vista do mesmo macro produzido pela Cambridge Spreadsheet Analyst

```
MACRO: <COMBINE> LISTING
B30 '{define infile,string}
B31 '/fcan
B32 'netincome~
B33
B34 '~
B35 '{return}
MACRO: <\C> LISTING
B11 '{indicate WAIT}
B12 '/wgrm
B13 '/redata~
B14 '{goto}data~
B15 '{combine depta}
B16 '{down}
B17 '{combine deptb}
B18 '{down}
B19 '{combine deptc}
B20 '{goto}data~
B21 '{down}
```

B22 '/rf,2~{end}{right}~
 B23 '{down}
 B24 '/rf,2~{end}{right}~
 B25 '/wgra
 B26 '{calc}
 B27 '{indicate}
 B28 '{quit}

Explicação:

B30 O início da subrotina do macro **combine**. Este macro pega um argumento, o valor texto que é estocado na célula chamada **infile**. O argumento necessita ser um texto em vez de um valor, porque é um texto usado mais tarde num macro..

B31 Começa Starts a Combinação de Arquivos **Add Named/Specific Regiãocommand**.

B32 Digita o nome da região para combinar e pressiona Enter para completar o nome..

B33 Esta célula, que tem o nome de região **infile**, contém o nome do arquivo para combinar quando o macro é executado. Esta célula é referenciada na célula B30, a célula do comando {define}..

B34 Pressiona a tecla Enter para completar a entrada do arquivo **incoming** da célula B33..

B35 O comando **{return}** para indicar o fim da subrotina e retornar o controle para o **calling location**..

B11 O início do macro \c, que combina os arquivos . Coloca o indicador de modo no topo direito da tela para **WAIT**. É uma boa idéia, porque este macro pode demorar um pouco a executar, e o indicador WAIT mostra isto para o usuário....

B12 Inclue o comando **Worksheet Global Recalculation Manual** que aumenta a velocidade da execução do macro..

B13 Usa o comando **RegiãoErase** para apagar os dados existentes na região de nome dos dados. Isto é necessário porque o comando *File Combine** não adicionou os valores das vendas aos números que ainda estão na Planilha..

B14 Usa o comando **{goto}** (o que equivale a pressionar a tecla de função F5) para mover a topo esquerdo da região de dados (célula B5) preparando para o **file combine**..

B15 Chama a subrotina do macro que combina com o argumento **depta** Isto combina os dados da planilha **depta**..

B16 Move o ponteiro da célula uma linha para baixo para preparar para o próximo **File Combine**.

B17 Combina os dados da planilha **debtb**.

B18 Move para baixo..

B19 Combina os dados da planilha **depte**.

B20 Atravez da B24 Formata algumas células com a Virgula, formato de duas casas decimais.

B25 Chama o comando ** Worksheet Global Recalculation Automatic** para restaurar cálculo automático.

B26 Pressiona a tecla **F9 Recalc** para atualizar as formulas.

B27 O comando **{indicate}**, com nenhum argumento, restaura a operação normal do indicador de modo..

B28 O fim do macro \c..