

Tutorial do Emacs. Veja no fim as condições para cópia.

Os comandos do Emacs geralmente envolvem a tecla CONTROL (algumas vezes nomeada CTRL ou CTL) ou a tecla META (algumas vezes nomeada EDIT ou ALT). Antes de escrever isso tudo toda vez, usaremos as seguintes abreviações:

- C-<chr> Significa pressionar a tecla CONTROL enquanto digitar o caractere <chr>. Assim, C-f seria: Pressionar a tecla CONTROL e digitar f.
- M-<chr> Significa pressionar a tecla META, EDIT ou ALT enquanto digitar <chr>. Se não tiver META, EDIT ou ALT, no lugar digite ESC e depois <chr>. Escrevemos <ESC> para tecla ESC.

Nota importante: para terminar a sessão do Emacs, digite C-x C-c. (Dois caracteres.) Os caracteres ">>" na margem esquerda indicam direções para você tentar usar um comando. Por exemplo:

<<linhas vazias inseridas ao redor da linha seguinte por ajuda-com-tutorial>>

[Metade da página em branco para propósitos didáticos. O texto continua abaixo]

>> Agora digite C-v (Ver próxima tela) para ir para próxima tela.

(vá em frente, faça isso pressionando a tecla CONTROL enquanto digitar v).

De agora em diante, você deve fazer isso de novo sempre que você terminar de ler a tela.

Note que tem uma sobreposição de duas linhas quando você move de tela em tela; isso fornece uma continuidade para que você continue lendo o texto.

A primeira coisa que você precisa saber é como se mover de um lugar a outro no texto. Você já sabe como mover para próxima tela, com C-v. Para mover para trás uma tela, digite M-v (pressione a tecla META enquanto digita v, ou digite <ESC>v se o seu teclado não tiver a tecla META, EDIT ou ALT).

>> Tente digitar M-v e depois C-v, algumas vezes.

## \* SUMÁRIO

-----

Os comandos a seguir são úteis para visualizar telas inteiras

- C-v Move para tela inteiraposterior
- M-x Move para tela inteira anterior
- C-l Limpa a tela e re-mostrá todo o texto, movendo o texto ao redor do cursor para o centro da tela. (Isso é control-L, não control-1.)

>> Encontre o cursor, e note o texto que está perto dele.

Digite C-l.

Encontre o cursor novamente e perceba que o mesmo texto continua perto do cursor.

## \* CONTROLE BÁSICO DO CURSOR

-----

Movendo de uma tela inteira para outra tela cheia é útil, mas como você move para um lugar específico com o mesmo texto na tela?

Existem diversas maneiras para você poder fazer isso. A maneira mais básica é utilizar os comandos C-p, C-b, C-f e C-n. Cada um desses comandos movem o cursor uma linha ou coluna na tela em uma direção particular. Aqui está uma tabela mostrando esses quatro comandos e as

direções que eles movem:

```

                Linha anterior, C-p
                :
                :
Traz, C-b .... Posição corrente do cursor .... Frente, C-f
                :
                :
                Linha posterior, C-n
```

>> Mova o cursor para a linha no meio do diagrama acima utilizando C-n ou C-p. Digite C-l para ver o diagrama inteiro centralizado na tela.

Você provavelmente achará isso fácil pensando neles pela letra: P para Previous (anterior), N para Next (Posterior), B para Backward (trás) e F para Forward (frente). Estes são os comandos básicos de posicionamento de cursor e você os estará utilizando a TODO tempo, então seria ótimo se você os aprendesse agora.

>> Faça alguns C-n's para trazer o cursor abaixo desta linha.

>> Mova para a linha com C-f's e depois para cima com C-p's.  
Veja o que C-p faz quando o cursor está no meio da linha.

Cada linha de texto termina com um caractere de nova linha, que serve para separar da próxima linha. No fim da última linha deve ter um caractere de nova linha (mas não é exigido pelo Emacs).

>> Tente digitar C-b no começo da linha. O cursor deve mover para o fim da linha anterior. Isso acontece por que ele se moveu para trás através do caractere de nova linha.

C-f pode mover o cursor através de uma Nova linha do mesmo jeito que C-b.

>> Faça mais alguns C-b's, assim você percebe onde o está o cursor.  
Depois alguns C-f's para retornar para o fim da linha.  
Depois mais um C-f para mover para linha seguinte.

Quando você move o cursor passando o topo ou o rodapé da tela, o próximo texto além da borda desloca-se na tela. Isso é chamado rolagem. Isso ativa o Emacs para mover o cursor para o lugar especificado no texto sem movê-lo fora da tela.

>> Tente mover o cursor fora do rodapé da tela com C-n, e veja o que acontece.

Se mover por caracteres é muito lento, você pode se mover por palavras. M-f (Meta-f) move uma palavra para frente e M-b uma palavra para trás.

>> Digite alguns M-f's e M-b's

Quando você está no meio de uma palavra, M-f move o cursor para o fim da mesma. Quando você está em um espaço em branco no meio de duas palavras, M-f move o cursor para o fim da palavra seguinte. M-b funciona do mesmo jeito na direção contrária.

>> Digite M-f e M-b algumas vezes, intercalado com C-f's e C-b's assim você pode observar a ação do M-f e M-b de varias lugares dentro e no meio de palavras.

Observe o paralelismo entre C-f e C-b por um lado, e M-f e M-b por outro lado. Muito freqüentemente os caracteres Meta são usados para operações relacionadas com as unidades definidas pela linguagem (palavras, sentenças, parágrafos), enquanto caracteres Control operam nas unidades básicas que são independente do que você está editando (caracteres, linhas, etc).

Esse paralelismo aplica-se entre linhas e sentenças: C-a e C-e move o cursor para o começo ou fim da linha, e M-a e M-e move o cursor para o começo ou fim da sentença.

>> Tente alguns C-a's, e depois alguns C-e's.  
Tente alguns M-a's, e depois alguns M-e's.

Veja como repetidos C-a's fazem nada, mas repetidos M-a's mantêm movendo mais uma sentença. Embora estes não sejam completamente análogos, cada um parece natural.

A posição do cursor no texto é também chamado "ponto" (point). Para parafrasear, o cursor mostra na tela onde o ponto é encontrado no texto.

Aqui está um simples sumário das operações de movimento de cursor, incluindo comandos de movimento de palavra e sentença.

C-f	Move o cursor um caractere para frente
C-b	Move o cursor um caractere para trás
M-f	Move o cursor uma palavra para frente
M-b	Move o cursor uma palavra para trás
C-n	Move o cursor para linha posterior
C-p	Move o cursor para linha anterior
C-a	Move o cursor para o começo da linha
C-e	Move o cursor para o fim da linha
M-a	Move o cursor para trás no começo da sentença
M-e	Move o cursor para frente no fim da sentença

>> Tente agora todos estes comandos algumas vezes para praticar.  
Estes são os comandos mais utilizados.

Outros dois comandos de movimento de cursor importantes são M-< (Meta Menor-que), o qual move o cursor para o começo de todo o texto, e M-> (Meta Maior-que), o qual move para o fim de todo o texto.

Na maioria dos terminais, o "<" é acima da vírgula, então você precisa utilizar a tecla shift para digitá-la. Nestes terminais você precisa utilizar a tecla shift para digitar M-< também; sem a tecla shift, você estaria teclando M-virgula.

>> Tente agora M-<, para mover o cursor ao começo do tutorial.  
Use então C-v repetidamente para trazer o cursor novamente aqui.

>> Tente agora M->, para mover o cursor ao fim do tutorial.  
Use então M-v repetidamente para trazer o cursor novamente aqui.

Você também pode mover o cursor com as setas, se o seu terminal tem teclas com setas. Nós recomendamos aprender C-b, C-f, C-n e C-p por três razões. Primeira, elas funcionam em todos os terminais. Segunda, uma vez que você ganhar prática usando o Emacs, você descobrirá que digitar nestes caracteres Control é mais rápido do que digitar nas setas (porque você não tem que mover suas mãos tão longe da posição de digitação). Terceira, uma vez que você tiver o hábito de utilizar estes comandos de caracteres Control, você pode facilmente aprender como usar outros comandos avançados de movimentação do cursor.

A maioria dos comandos do Emacs aceitam argumentos numéricos; para maioria dos comandos, isso serve como um fator de repetição. A maneira que você dá a um comando uma repetição teclando C-u e depois os dígitos antes de digitar o comando. Se você tem o META (EDIT ou ALT), tem outra alternativa para digitar um argumento numérico: digite os dígitos enquanto pressionando a tecla META. Nós recomendamos aprender o método C-u porque funciona em todos os terminais. O argumento numérico também é chamado de "argumento prefixo", porque você digita o argumento antes do comando que ele se aplica.

Por exemplo, C-u 8 C-f move o cursor oito caracteres para frente

>> Tente usar C-n ou C-p com um argumento numérico, para mover o cursor para um linha perto dessa com um único comando.

A maioria dos comandos usam o argumento numérico para um fator de repetição, mas alguns comandos usam isso de outra maneira. Diversos comandos (mas nenhum daqueles comandos que você tenha aprendido até agora) usam isso como uma marca -- a presença de um argumento prefixo, independente do seu valor, faz o comando fazer alguma coisa diferente.

C-v e M-v são outro tipo de exceção. Quando você dá um argumento, eles rolam a tela para cima ou para baixo por aquele número de linhas, de preferência com a tela cheia. Por exemplo, C-u 8 C-v rola a tela 8 linhas.

>> Tente agora digitar C-u 8 C-v.

Isso deve ter rolado a tela para cima 8 linhas. Se você quiser rolar a tela para baixo novamente, você pode dar um argumento para o M-v.

Se você está usando um display com janelas, como o X11 ou MS-Windows, deve ter uma área retangular alta chamada barra de rolagem na mão esquerda da janela do Emacs. Você pode rolar o texto clicando o mouse na barra de rolagem.

>> Tente pressionar o botão do meio no topo da área destacada junto a barra de rolagem. Isso deve rolar o texto para uma determinada posição dependendo de quão alto ou baixo você clicar.

>> Tente mover o mouse para cima e para baixo, enquanto pressionar o botão do meio. Você verá que o texto rolará para cima e baixo assim que você mover o mouse.

#### \* QUANDO O EMACS PARECE MORTO

-----

Se o Emacs parar de responder os seus comandos, você pode seguramente pará-lo digitando C-g. Você pode utilizar C-g para parar um comando que está levando muito tempo para executar.

Você também pode utilizar C-g para interromper um argumento numérico ou o início de um comando que você não quer terminar.

>> Digite C-u 100 para fazer um argumento numérico de 100, e depois digite C-g. Agora digite C-f. O Cursor deve mover apenas um caractere, porque você cancelou o argumento com C-g.

Se você digitou um <ESC> por engano, você pode sair dele com um c-g.

#### \* COMANDOS DESABILITADOS

-----

Alguns comandos do Emacs são "desabilitados" para que os usuários iniciantes não utilizem por acidente.

Se você digita um dos comandos desabilitados, o Emacs mostrará uma mensagem dizendo qual comando foi, e pergunta se você quer seguir e executar o comando.

Se você realmente quiser tentar o comando, digite Espaço na resposta da pergunta. Normalmente, se você não quer executar o comando desabilitado, responda a pergunta com "n".

>> Digite C-x C-1 (que é um comando desabilitado), e então digite n para responder a pergunta.

## \* JANELAS

-----

O Emacs pode ter diversas janelas, cada uma mostrando seu próprio texto. Nós vamos explicar mais tarde como utilizar estas múltiplas janelas. Neste momento nos queremos explicar como se livrar das janelas extras e voltar a edição básica em uma janela. É simples:

C-x 1 Uma janela (mate todas as outras janelas).

Isto é Control-x seguido pelo dígito 1. C-x 1 expande a janela que está com o cursor, para ocupar a tela inteira. Ela remove todas as outras janelas.

>> Mova o cursor para esta linha e digite C-u 0 C-l.

>> Digite Control-h k Control-f.

Veja como esta janela diminui, enquanto uma nova aparece para mostrar a documentação do comando Control-f.

>> Digite C-x 1 e veja a listagem de documentação desaparecer.

Este comando é diferente dos outros que você aprendeu por isso ele consiste em dois caracteres. Ele inicia com o caractere Control-x. Tem uma grande série de comandos que iniciam com Control-x; a maioria deles tem a ver com janelas, arquivos, buffers, e coisas relacionadas. Estes comandos tem dois, três ou quatro caracteres de tamanho.

## \* INSERINDO E REMOVENDO

-----

Se você quiser inserir um texto, apenas digite o texto. Os caracteres que você verá, como A, 7, \*, etc. são vistos pelo Emacs como texto e são inseridos imediatamente. Digite <Return> para inserir um caractere de nova linha.

Você pode remover o último caractere que você inseriu digitando <Delete>. <Delete> é uma tecla no teclado, que está nomeado como "Del". Em algumas casos, a tecla "Backspace" serve como o <Delete>, mas nem sempre!

Mas geralmente, <Delete> remove o caractere imediatamente anterior a posição corrente do cursor.

>> Faça isso agora - Digite alguns caracteres, então remova-os digitando <Delete> algumas vezes. Não se preocupe em modificar este arquivo; você não alterará o arquivo principal. Esta é a sua cópia pessoal dele.

Quando uma linha de texto fica muito grande na tela, a linha do texto é continuada na linha seguinte. Uma barra invertida ("\") na margem direita indica a linha que foi continuada.

>> Insira texto até que você alcance a margem direita, e mantenha inserindo. Você verá a linha de continuação aparecer.

>> Utilize <Delete> para remover o texto até a linha caber em uma linha novamente. A linha de continuação irá embora.

Você pode remover o caractere de nova linha como qualquer outro caracter. Removendo o caractere de nova linha entre duas linhas você juntará as duas em uma única linha. Se o resultado for muito longo para acomodar na largura da tela, aparecerá um caractere de continuação.

>> Mova o cursor para o início da linha e digite <Delete>. Isso juntará esta linha com a anterior.

>> Digite <Return> para re-inserir a nova linha que você removeu.

Lembre-se que a maioria dos comandos do Emacs podem receber um número de repetição; isso inclui texto de caracteres. Repetir um texto de caractere o insere algumas vezes.

>> Tente isso agora -- digite C-u 8 \* para inserir \*\*\*\*\*.

Você acabou de aprender o jeito mais simples de digitar alguma coisa no Emacs e corrigir os erros. Você pode remover palavras ou linhas também. Aqui vai um sumário dos comandos de deleção:

<Delete>	remove o caractere imediatamente antes do cursor
M-d	remove o próximo caractere apos o cursor
M-<Delete>	Mata a palavra imediatamente antes do cursor
M-d	Mata a próxima palavra depois do cursor
C-k	Mata a posição do cursor até o fim da linha
M-k	Mata até o fim da sentença corrente

Note que o <Delete> e o C-d vs M-<Delete> e M-d estende o paralelo iniciado por C-f e M-f (bem, <Delete> não é realmente um controle de caractere, mas não vamos nos preocupar com isso). C-k e M-k são como C-e e M-e, uma parte, nessas linhas são sentenças opostas.

Você pode também matar parte de um buffer com um método uniforme. Mova para o fim dessa parte, e digite C-@ ou C-SPC (um ou outro). (SPC é a Barra de Espaço.) Mova para o outro fim dessa parte, e digite C-w. Isso matará todo o texto entre as duas posições.

>> Mova o cursor para o V no inicio do parágrafo anterior.  
>> Digite C-SPC. Emacs vai mostrar a mensagem "Mark set" no rodapé da tela.  
>> Mova o cursor para o . (ponto) no "fim", na segunda linha do parágrafo.  
>> Digite C-w. Isso vai matar o texto iniciando no V, e terminando bem antes do . (ponto).

A diferença entra "matar" e "remover" é que o texto "matado" pode ser re-inserido, onde os textos "removidos" não podem ser re-inseridos. Re-inserção de um texto matado chama-se colar (yanking). Geralmente, o comando que pode remover bastante texto mata o texto (eles fazem com que você possa colar (yank) o texto), enquanto os comandos que removem um caractere, ou apenas linhas brancas ou espaços, fazem deleção (para que você não possa colar esse texto).

>> Mova o cursor para o inicio da linha que não esteja vazia.  
Agora digite C-k para matar o texto nessa linha.  
>> Digite C-k pela segunda vez. Você verá que a nova linha que segue a linha será matada.

Note que um simples C-k mata o conteúdo de uma linha, é um segundo C-k mata a linha, e faz com que todas as outras linhas movam para cima. C-k trata um argumento numérico especialmente: ele mata tantas linhas E seus conteúdos. Isso não é uma simples repetição. C-u 2 C-k mata duas linhas e suas novas linhas; digitar C-k duas vezes não faria isso.

Para trazer o texto matado novamente é chamado colar "yaking". (Pense nisso como colar, ou trazer de volta, algum texto que foi levado embora.) Você pode colar (yank) o texto matado no mesmo lugar onde ele foi matado, ou em qualquer outro lugar do buffer, ou ate em um arquivo diferente. Você pode colar o texto diversas vezes, o que fará diversas cópias dele.

O comando para colar é C-y. Ele re-insere o último texto matado, na posição corrente do cursor.

>> Tente; digite C-y para colar o texto novamente.

Se você fizer diversos C-k's em uma linha, todo esse texto matado será salvo junto, então aquele C-y vai colar todas aquelas linhas de uma vez.

>> Faça isso agora, digite C-k diversas vezes.

Agora pegue o texto matado:

Digite C-y. Então mova o cursor para baixo algumas linhas e digite C-y novamente. Agora você sabe como copiar determinado texto.

O que você faria se você tivesse determinado texto que você gostaria de colar, e então você mata alguma outra coisa? C-y colará o kill mais recente. Mas o antigo texto ainda não foi perdido. Você pode tê-lo novamente usando o comando M-y. Depois que você der um C-y para pegar o kill mais recente, digitar M-y troca o que você colou pelo anterior a ele. Digitar M-y novamente e novamente traz kills anteriores e anteriores. Quando você tiver alcançado o texto que você que esta procurando, você não precisa fazer nada para mantê-lo. Apenas continue com a sua edição, deixando o texto onde ele esta.

Se você digitar M-y o suficiente, você começará do início novamente (o kill mais recente).

>> Mate uma linha, mova por ai, mate mais uma linha.  
Agora digite C-y para colar a segunda linha matada.  
Agora digite M-y e ele vai ser trocado pelo primeiro kill.  
Faça mais alguns M-y's e veja o que você tem. Faça isso ate que o segundo kill volte, e depois mais alguns.  
Se você preferir, você pode tentar dar argumentos positivos e negativos.

#### \* DESFAZENDO (UNDO)

-----

Se você fizer uma mudança no texto, e então decidir que isso foi um erro, você pode desfazer com o comando de desfazer (undo), C-x u.

Normalmente, C-x u desfaz as mudanças feitas por um comando; se você repetir C-x u diversas vezes em uma linha, cada repetição desfazá um comando adicional.

Mas tem duas exceções: comandos que não trocam o texto não contam (isso inclui o movimento do cursor e a barra de rolagem), e caracteres auto-inseridos são usualmente cuidados em grupos de mais de 20 (Isso e para reduzir o numero de C-x u's que você deve digitar para desfazer a inserção de um texto.).

>> Mate essa linha com C-k, então digite C-x u e isso vai re-aparecer.

C-\_ é um comando alternativo para desfazer; ele funciona exatamente como o C-x u, mas é mais fácil de digitar diversas vezes em uma linha. A desvantagem de C-\_ é que alguns teclados ele não é tão obvio para digitar. É exatamente por isso que o C-x u é fornecido. Em alguns terminais, você pode digitar C-\_ digitando / enquanto segurando o CONTROL.

Um argumento numérico para C-\_ ou C-x u age como um contador de repetição.

Você pode desfazer a deleção de texto da mesma maneira que você pode desfazer o kill de um texto. A distinção entre matar e remover alguma coisa afeta se você pode colar com C-y; isso não faz diferença para o desfazer (undo).

#### \* ARQUIVOS

-----

Para poder fazer o texto que você edita permanente, você precisa colocá-lo em um arquivo. Ou então, isso será perdido quando o seu Emacs for fechado. Para colocar seu texto em um arquivo, você precisa "encontrar" o arquivo antes de digitar o texto (Isso também é chamado "visitar" um arquivo.).

Encontrando um arquivo significa que você verá o conteúdo de um arquivo com o Emacs. De diversas maneiras, isso é como se você estivesse editando o arquivo. De qualquer maneira, as mudanças que você fizer utilizando o Emacs não se tornarão permanente até que você "salve" o arquivo. Isso é para que você evite deixar um arquivo meio-modificado no sistema enquanto você não quer isso. Mesmo quando você salva, o Emacs deixa o arquivo original sobre um arquivo de nome alterado no caso que você mais tarde decida que aquelas mudanças foram um erro.

Se você olhar no rodapé da tela você verá a linha que começa e termina com traços e inicia com "--:-- TUTORIAL.pt\_BR" ou algo parecido com isso. Essa parte da tela normalmente mostra o nome do arquivo que você está visitando. Agora mesmo, você está visitando um arquivo chamado "TUTORIAL.pt\_BR" que é sua própria cópia do tutorial do Emacs. Quando você encontra um arquivo com o Emacs, o nome do arquivo aparecerá nesse lugar específico.

Algo especial para se dizer sobre o comando de encontrar arquivo e que você tem que dizer qual nome do arquivo você quer. Dizemos que o comando "lê um argumento do terminal" (nesse caso, o argumento é o nome do arquivo). Depois que você digitar o comando

C-x C-f Encontrar um arquivo

O Emacs pede para você digitar o nome do arquivo. O nome do arquivo aparece no rodapé da tela. A linha de rodapé é chamada de minibuffer quando é utilizada para esse tipo de introdução de dados. Você pode utilizar os comandos padrões do Emacs para editar o nome do arquivo.

Enquanto você estiver digitando o nome do arquivo (ou qualquer outro tipo de dado para o minibuffer), você pode cancelar o comando com a tecla C-g.

>> Digite C-x C-f, então digite C-g. Isso cancelará o minibuffer, e também cancelará o comando C-x C-f que estava sendo utilizado no minibuffer. Para que você não procure nenhum arquivo.

Quando você terminar de digitar o nome do arquivo, digite <Return> para terminar. Então o comando C-x C-f executará, e encontrará o arquivo que você escolheu. O minibuffer desaparece quando o comando C-x C-f é finalizado.

Em pouco tempo o conteúdo do arquivo aparece na tela, e você poderá editar o conteúdo dele. Quando você terminar de fazer suas alterações permanentes digite o comando

C-x C-s Salve o arquivo

Isso copia o texto do Emacs para o arquivo. A primeira vez que você fizer isso, o Emacs vai renomear o arquivo original para um novo nome para que você não o perca. O nome novo é feito adicionando "~" no fim do nome original do arquivo.

Quando o salvamento for concluído, Emacs imprimirá o nome do arquivo escrito. Você deve salvar o conteúdo do seu arquivo regularmente, para que você não perca muito trabalho caso o sistema caia por um algum motivo.

>> Digite C-x C-s, salvará a cópia do seu tutorial.  
Isso imprimirá "Wrote ...TUTORIAL.pt\_BR" no fim da tela.

Você pode encontrar um arquivo já existente, para vê-lo ou editá-lo. Você



também pode encontrar um arquivo que ainda não existe. Essa é a maneira de criar um arquivo com o Emacs: encontre o arquivo, que iniciará vazio, então comece a inserir o texto para o arquivo. Quando você pedir para "salvar" o arquivo, o Emacs perguntará se você realmente quer criar o arquivo com o texto que você acabou de inserir. De agora em diante, você pode considerar que esta editando um arquivo já existente.

#### \* BUFFERS

-----

Se você encontrar um segundo arquivo com C-x C-f, o primeiro continuara dentro do Emacs. Você pode voltar para ele encontrando-o novamente com C-x C-f. Dessa maneira você pode ter um grande numero de arquivos dentro do Emacs.

>> Crie um arquivo chamado "foo" digitando C-x C-f foo <Return>. Depois digite algum texto, edite-o, e salve "foo" digitando C-x C-s. Finalmente, digite C-x C-f TUTORIAL.pt\_BR <Return> para voltar ao tutorial.

O Emacs guarda cada texto de um arquivo em um objeto chamado "buffer". Encontrar um arquivo cria um novo buffer dentro do Emacs. Para ver uma lista dos buffers que existem no serviço do seu Emacs, digite

C-x C-b Lista os buffers

>> Tente C-x C-b agora.

Veja como cada buffer tem um nome, e ele também pode ter um nome de arquivo para o conteúdo de arquivo que ele armazena. Alguns buffers não correspondem a arquivos. Por exemplo, o buffer chamado "\*Buffer List\*" não tem nenhum arquivo. Esse é o buffer que contem a lista de buffers que foi feito com C-x C-b. QUALQUER texto que você ver numa janela do Emacs e sempre parte de algum buffer.

>> Digite C-x 1 para sair da lista dos buffers.

Se você fizer mudanças no texto de um arquivo, então encontrar um novo arquivo, isso não salva o conteúdo do primeiro arquivo. As mudanças continuam dentro do Emacs, nos buffers de arquivo. A criação ou edição do buffer do segundo arquivo não tem nenhum efeito no buffer do primeiro arquivo. Isso é muito útil, mas também significa que você precisa de uma maneira conveniente de salvar o buffer do primeiro arquivo. Seria inconveniente ter que voltar para ele com C-x C-f para salvá-lo com C-x C-s. Por isso temos

C-x s Salve alguns buffers

C-x s pergunta a você sobre cada buffer que contem modificação e você não salvou. Ele te pergunta, para cada buffer, se você quer salvar.

>> Insira uma linha de texto, então digite C-x s. Isso deve te perguntar se você quer salvar o buffer chamado TUTORIAL.pt\_BR. Responda sim para a pergunta digitando "y".

#### \* ESTENDENDO O COMANDO SET

-----

Existem mais, muito mais comandos que o Emacs poderia possivelmente ser colocado em todos os control e caracteres meta. O Emacs faz isso com o comando X (estender). Que vem em duas versões:

C-x Extensor de caractere. Seguido por um caractere.  
M-x Extensor de nome de comando. Seguido por um longo nome.

Esses comandos que são geralmente úteis mas pouco usados que os comandos

que você já aprendeu. Você já viu dois deles: os comandos de arquivos C-x C-f para Encontrar e C-x C-s para Salvar. Outro exemplo de um comando para finalizar a sessão do Emacs - esse é o comando C-x C-c (Não se preocupe em perder as mudanças que você fez; C-x C-c oferece salvar cada arquivo modificado antes de matar o Emacs.).

C-z é o comando para sair do Emacs \*temporariamente\* - então você pode voltar para a mesma seção do Emacs depois.

Em sistemas que permitem isso, C-z "suspende" o Emacs; isso e, retorna ao shell mas não destrói o Emacs. Nos shells mais comum, você pode resumir o Emacs com o comando 'fg' ou com '%emacs'.

Nos sistemas que não implementam suspensão, C-z cria um subshell rodando sobre o Emacs para dar a você a chance de rodar outros programas e retornar ao Emacs depois; isso não "sai" verdadeiramente do Emacs. Nesse caso, o comando shell 'exit' é o comando usual para voltar ao Emacs do subshell.

A hora de usar o C-x C-c é quando você esta prestes a sair do sistema operacional. É também a maneira correta de sair do Emacs quando chamado através de um cliente de e-mail ou outros utilitários, porque eles não controlam a suspensão do Emacs. Em outras circunstâncias, se você não estiver saindo do sistema operacional, é melhor suspender o Emacs com C-z ao invés de sair do Emacs.

Existem diversos comando C-x. Aqui esta uma lista dos que você aprendeu:

C-x C-f	Encontrar arquivo.
C-x C-s	Salvar arquivo.
C-x C-b	Listar buffers.
C-x C-c	Sair do Emacs.
C-x 1	Remover todas menos a janela atual.
C-x u	Desfazer.

O extensor de nome de comando é usado menos ainda, ou comandos que são usados apenas em alguns modos. Um exemplo é o comando replace-string, que globalmente troca uma string por outra. Quando você digita M-x, o Emacs te pergunta no rodapé da tela M-x e você deve digitar o nome do comando; nesse caso, "replace-string". Apenas digite "repl s<TAB>" e o Emacs completará o nome. (<TAB> é a tecla Tab, usualmente encontrando abaixo do CapsLock ou Shift bem na extremidade do teclado.). Termine o comando com <Return>.

O comando replace-string requer dois argumentos - a string para ser trocada, e a string que a substituirá. Você precisa terminar cada argumento com <Return>.

```
>> Mova o cursor para a linha branca duas linhas abaixo dessa.  
  Digite então M-x repl s<Return>mudou<Return>alterado<Return>.
```

Note como essa linha mudou: você substituiu a palavra m-u-d-o-u por "alterado" quando ocorreu, apos a posição inicial do cursor.

#### \* SALVAMENTO AUTOMÁTICO

-----

Quando você fizer mudanças em um arquivo, mas você ainda não tiver salvo, elas podem ser perdidas se o seu computador der algum problema. Para proteger você disso, o Emacs periodicamente escreve um arquivo "auto salvo" para cada arquivo que você estiver editando. O nome do arquivo auto salvo tem um # no inicio e no fim; por exemplo, se o seu arquivo tiver o nome "ola.c", o nome do arquivo auto salvo dele será "#ola.c#". Quando você salvar o arquivo da maneira normal, o Emacs removerá os arquivos auto salvos.

Se o computador der problema, você pode recuperar suas modificações auto salvas abrindo o arquivo normalmente (o arquivo que você estava

editando, não o arquivo auto salvo) e então digitar M-x recover file<Return>. Quando for pedida a confirmação, digite yes<Return> para ir em frente e recuperar os dados auto salvos.

#### \* ÁREA DE SAÍDA (ECHO AREA)

-----

Se o Emacs ver que você está digitando comandos multi-caractere lentamente, ele os mostrará para você no fim da tela numa área chamada Área de Saída ou Echo Area. A área de saída contém a linha de rodapé da tela.

#### \* MODE LINE

-----

A linha imediatamente acima da área de saída é chamada de "mode line". O mode line diz algo algo mais ou menos assim:

```
--:** TUTORIAL.pt_BR      (Fundamental)--L670--58%-----
```

Essa linha dá informações úteis sobre o status do Emacs e o texto que você está editando.

Você já sabe o que o nome do arquivo significa -- é o arquivo que você encontrou. -NN%- indica sua posição no arquivo texto; significa que NN por cento do texto está acima do topo da tela. Se o topo do arquivo estiver na tela, ele mostrará --Top-- ao invés de --00%-. Se o rodapé do texto estiver na tela, ele mostrará --Bot--. Se você estiver olhando um texto muito pequeno que cabe todo na tela, o mode line mostrará --All--.

O L são dígitos e indicam a posição de uma outra maneira: eles mostram a linha corrente do ponto.

Os asteriscos próximo ao início significam que você fez mudanças no texto. Logo após você visitar um arquivo ou salvar um arquivo, essa parte do mode line não mostrará asteriscos, apenas traços.

A parte do mode line dentro do parênteses lhe diz quais modos você está utilizando. O modo padrão é o Fundamental que é o que você está utilizando agora. É um exemplo de um "modo majoritário".

O Emacs tem diversos diferentes modos majoritários. Alguns deles são para edição de diferentes linguagens de programação e/ou tipos de texto, como modo Lisp, modo Texto, etc. Em qualquer momento um é apenas um modo majoritário e ativo, e seu nome pode ser encontrado no mode line exatamente onde o "Fundamental" está agora.

Cada modo majoritário faz com que alguns comandos operem de modo diferente. Por exemplo, existem comandos para criar comentários nos programas, e desde que cada linguagem de programação tem uma idéia diferente de como um comentário deve ser, cada modo majoritário tem que inserir seus comentários individualmente. Cada modo majoritário é o nome de um comando estendido, e é assim que você pode mudar para determinado modo. Por exemplo, M-x fundamental-mode é o comando para trocar para o modo Fundamental.

Se você vai editar texto em Português, como neste arquivo, você pode utilizar o modo Texto.

>> Digite M-x text mode<Return>.

Não se preocupe, nenhum dos comandos do Emacs que você aprendeu mudarão. Mas você pode observar que o M-f e o M-b tratam os apóstrofes como parte das palavras. Anteriormente, no modo Fundamental, M-f e M-b trataram os apóstrofes como separadores de palavras.

Modos majoritários fazem usualmente modificações como esta: a maioria dos comandos fazem "a mesma coisa" em cada modo majoritário, mas eles trabalham um pouquinho diferente.

Para ver a documentação do seu modo majoritário, digite C-h m.

```
>> Utilize C-u C-v uma ou mais para trazer essa linha próxima ao topo da
tela.
>> Digite C-h m, para ver como o modo texto difere do modo Fundamental.
>> Digite C-x 1 para remover a documentação da tela.
```

Modos majoritários são chamados de majoritários porque também existem modos minoritários. Modos minoritários não são alternativos aos modos majoritários, apenas uma modificação minoritária deles. Cada modo minoritário pode ser ligado ou desligado por ele mesmo, independente de todos os outros modos minoritários, é independente do modo majoritário. Então você pode não utilizar modos minoritários, ou um modo minoritário, ou qualquer combinação de modos minoritários.

Um modo minoritário que é bastante útil, especialmente para editar texto em Português, é o modo Auto Fill. Quando ele está ligado, o Emacs quebra a linha entre as palavras automaticamente quando você inserir um texto e criar uma linha muito comprida.

Você pode ligar o modo Auto Fill fazendo M-x auto fill mode<Return>. Quando o modo estiver ligado, você pode desligá-lo novamente fazendo M-x auto fill mode<Return>. Se o modo estiver desligado, esse comando o liga, e se ele estiver ligado, este comando o desligará. Nos dizemos que o comando inverte seu status.

```
>> Digite M-x auto fill mode<Return> agora. Então insira uma linha de
"asdf " diversas vezes ate que você veja dividir em duas linhas. Você
precisa colocar espaços entre as palavras porque o modo Auto Fill
quebra apenas nos espaços.
```

A margem é usualmente definida com 70 caracteres, mas você pode modificá-la com o comando C-x f. Você deve dar a margem que você quer como um argumento numérico.

```
>> Digite C-x f com um argumento de 20. (C-u 20 C-x f).
Então digite algum texto e veja o Emacs encher linhas com ate 20
caracteres. Então, volte a definir a margem para 70 utilizando C-x f.
```

Se você fizer uma mudança no meio de um parágrafo, o modo Auto Fill não re-preencherá para você. Para re-preencher o parágrafo, digite M-q (Meta-q) com o cursor dentro do parágrafo.

```
>> Mova o cursor dentro do parágrafo anterior e digite M-q.
```

#### \* PESQUISANDO

-----

O Emacs pode fazer pesquisas por strings (esses são grupos de caracteres ligados ou palavras) tanto para frente do texto ou para trás. Pesquisar por uma string é um comando de movimento do cursor; ele move o cursor para o próximo lugar onde a string aparece.

O comando de pesquisa do Emacs é diferente dos comandos de pesquisa da maioria dos editores, porque é uma pesquisa "incremental". Isso significa que a pesquisa acontece enquanto você digita uma string para pesquisar.

O comando para iniciar a pesquisa é C-s para pesquisa para frente ou C-r para pesquisa inversa. MAS ESPERE! Não tente fazer agora.

Quando você digitar C-s você perceberá que uma string "I-search" aparecerá como prompt na área de saída. Isso diz para você que o Emacs está no que chamamos de pesquisa incremental esperando que você digite o

que você quer procurar. <Return> termina a pesquisa.

>> Agora digite C-s para iniciar a pesquisa. DEVAGAR, uma letra por vez, digite o caractere 'cursor', pausando cada vez que você digitar um caractere, perceba o que acontece com o cursor. Agora que você pesquisou por "cursor", uma vez.  
>> Digite C-s novamente, para pesquisa a próxima ocorrência de "cursor".  
>> Agora digite <Delete> quatro vezes e veja como o cursor move.  
>> Digite <Return> para terminar a pesquisa.

Você viu o que aconteceu? O Emacs, em uma pesquisa incremental, tenta ir para a ocorrência que você digitou até o momento. Para ir para próxima ocorrência de 'cursor' apenas digite C-s novamente. Se nenhuma ocorrência existir, o Emacs bipa e te diz que a pesquisa atual falhou. C-g também termina a pesquisa.

NOTA: Em alguns sistemas, digitar C-s trava a tela e você não verá a saída do Emacs. Isso indica que um "recurso" do sistema operacional chamado "controle de fluxo" (flow control) está interceptando o C-s e não deixando que passe pelo Emacs. Para destravar a tela, digite C-q. Então, leia na seção "Spontaneous Entry to Incremental Search" no manual do Emacs para uma dica de como lidar com esse "recurso".

Se você estiver no meio de uma pesquisa incremental e digitar <Delete>, você perceberá que o último caractere da pesquisa será apagado e a pesquisa voltará para o último lugar da pesquisa. Por exemplo, imagine que você tenha digitado "c", para pesquisar a primeira ocorrência de "c". Agora se você digitar "u", o cursor moverá para a primeira ocorrência de "cu". Agora digite <Delete>. Isso apagará o "u" da string de pesquisa, e o cursor moverá de volta para a primeira ocorrência de "c".

Se você estiver no meio de uma pesquisa e digitar um caractere control ou meta (com algumas exceções--caracteres que são especiais em uma pesquisa, como C-s e C-r), a pesquisa é terminada.

O C-s inicia uma pesquisa que procura por qualquer ocorrência da string de pesquisa APÓS a posição corrente do cursor. Se você quiser pesquisar por alguma coisa anterior no texto, digite C-r ao invés. Tudo que nos dissemos sobre C-s também implica para C-r, tirando a direção da pesquisa que é reversa.

## \* MÚLTIPLAS JANELAS

-----

Um dos recursos mais interessantes do Emacs é que você pode mostrar mais de uma janela na tela ao mesmo tempo.

>> Mova o cursor para essa linha e digite C-u 0 C-l (isso é control-L, não control-1).

>> Agora digite C-x 2 que vai dividir a tela em duas janelas. As duas janelas vão mostrar esse tutorial. O cursor fica na janela do topo.

>> Digite C-M-v para rolar para o fim da janela.  
(Se você não tiver uma tecla Meta de verdade, digite ESC C-v.).

>> Digite C-x o ("o" para "outra") para mover o cursor para o rodapé da janela.

>> Use C-v e M-v no rodapé da janela para rodá-la.  
Continue lendo essas direções na janela do topo.

>> Digite C-x o novamente para mover o cursor de volta para a janela do topo.  
O cursor na janela do topo está exatamente como estava antes.

Você pode continuar usando C-x o para alternar entre as janelas. Cada

janela tem sua própria posição de cursor, mas apenas uma janela mostra o cursor. Todos os comandos mais comuns de edição aplicam para a janela que o cursor esta. Chamamos isso de "janela selecionada".

O comando C-M-v é muito útil para quando você estiver editando o texto em uma janela e utilizando a outra janela para referencia. Você pode manter o cursor sempre na janela que você estiver editando, e avançar pela outra janela sequencialmente utilizando C-M-v.

C-M-v é um exemplo de um caractere CONTROL-META. Se você tiver uma tecla META real, você pode digitar C-M-v segurando ambos CONTROL e META enquanto digitar v. Não importa se CONTROL ou META "vem primeiro," porque ambas as teclas agem modificando o caractere que você digita.

Se você não tiver um tecla META real, você pode utilizar ESC ao invés, a ordem não importa: você precisa digitar ESC seguido por Control-v, porque Control-ESC v não funcionará. Isso porque ESC é um caractere com seu próprio direito, não é uma tecla de modificação.

>> Digite C-x 1 (na janela do topo) para fugir da janela do rodapé.

(Se você tivesse digitado C-x 1 na janela do rodapé, você fugiria da janela do topo. Pense neste comando como "Mantenha apenas uma janela--a janela que estou dentro.")

Você não precisa mostrar o mesmo buffer nas duas janelas. Se você utilizar C-x C-f para encontrar um arquivo em outra janela, a outra janela não mudara. Você pode encontrar um arquivo em cada janela independentemente.

Aqui esta uma maneira de utilizar duas janelas para mostrar coisas diferentes:

>> Digite C-x 4 C-f seguido pelo nome de um dos seus arquivos. Termine com <Return>. Veja o arquivo especificado aparece na janela do rodapé. O cursor irá para lá, também.

>> Digite C-x o para voltar a janela do topo, e C-x1 para remover a janela de rodapé.

#### \* NÍVEIS RECURSIVOS DE EDIÇÃO

-----

Algumas vezes você entrará no que chamamos de "nível de edição recursiva". Isso é indicado por colchetes no mode line, envolvendo os parênteses em volta do nome do modo majoritário. Por exemplo, você pode ver [(Fundamental)] ao invés de (Fundamental).

Para sair desses níveis recorríveis de edição, digite ESC ESC ESC. Isso é um comando de todos os propósitos "saia". Você pode também utilizar para eliminar janelas extras, e sair de um minibuffer.

>> Digite M-x para entrar no minibuffer; então digite ESC ESC ESC para sair.

Você não pode utilizar C-g para sair de um nível de edição recursiva. Isso porque o C-g é utilizado para cancelar comandos e argumentos COM o nível de edição recursiva.

#### \* CONSEGUINDO MAIS AJUDA

-----

Neste tutorial nos tentamos suprimir informações suficientes para que você comece a utilizar o Emacs. Existe tanto disponível sobre Emacs que seria impossível explicar tudo aqui. De qualquer maneira, você pode querer aprender mais sobre Emacs mesmo porque ele tem muitos outros recursos úteis. O Emacs proporciona comandos para leitura de

documentação sobre comandos do Emacs. Esses comandos de "ajuda" iniciam com o caractere Control-h, que é chamado de "o caractere de ajuda".

Para utilizar os recursos de ajuda, digite o caractere C-h, e então um caractere dizendo que tipo de ajuda você quer. Se você REALMENTE estiver perdido, digite C-h ? e o Emacs lhe dirá que tipo de ajuda ele pode lhe dar. Se você digitar C-h e decidir que você não quer nenhuma ajuda, apenas digite C-g para cancelar.

(Alguns sistemas mudam o significado do caractere C-h. Eles não deveriam fazer isso para todos os usuários, então você deve reclamar com o administrador do sistema. Enquanto isso, se C-h não mostrar a mensagem sobre ajuda no rodapé da tela, tente digitar a tecla F1 ou M-x help <Return>.)

O recurso de ajuda mais básico é o C-h c. Digite C-h, e o caractere c, é um comando de caractere ou sequência; então o Emacs mostrará uma breve descrição do comando.

```
>> Digite C-h c C-p
      A mensagem deve ser algo assim
```

```
      C-p roda o comando linha-anterior
```

Isso diz a você "o nome da função". Nome de funções são utilizados basicamente para personalizar o estender o Emacs. Mas desde que nomes de funções são utilizados para indicar que o que o comando faz, eles podem servir também como uma breve documentação--suficiente para lembrar você os comandos que você já aprendeu.

Comandos multi-caractere como C-x C-s e (se você não tiver a tecla META ou EDIT ou ALT) <ESC>v é também permitido depois de C-h c.

Para ter mais informação sobre um comando, utilize C-h k ao invés de C-h c.

```
>> Digite C-h k C-p.
```

Isso mostra a documentação da função, assim como seu nome, em uma janela do Emacs. Quando você terminar de ler a saída, digite C-x 1 para sair da saída do texto de ajuda. Você não precisa fazer isso imediatamente apos. Você pode fazer alguma edição enquanto utilizando como referencia o texto de ajuda, e então digite C-x 1.

Aqui estão algumas outras opções úteis do comando C-h:

```
C-h f          Descreve a função. Digite o nome da função.
```

```
>> Tente digitar C-h f previous-line<Return>
      Isso imprimirá toda informação que o Emacs tem sobre a função que
      implementa o comando C-p.
```

```
C-h a          Comando Apropos. Digite uma palavra chave e o Emacs
                listará todos os comandos que o nome contem essa
                palavra. Esses comandos podem ser chamados com
                Meta-x. Para alguns comandos, o comando Apropos
                também listará um ou dois caracteres de sequência que
                roda o mesmo comando.
```

```
>> Digite C-h a file<Return>.
```

Isso mostra em uma outra janela a lista de todos os comandos M-x com "arquivo" em seus nomes. Você verá comandos-caractere como C-x C-f listados ao lado do comando correspondente assim como o find-file.

```
>> Digite C-M-v para rolar a janela de ajuda. Faça isso algumas vezes.
```

```
>> Digite C-x 1 para remover a janela de ajuda.
```

## \* CONCLUSÃO

-----

Lembre, para sair permanentemente do Emacs utilize C-x C-c. Para sair para um shell temporariamente, para que você possa voltar ao Emacs mais tarde, utilize C-z.

Esse tutorial foi feito para ser entendível para todos os novos usuários, então se você encontrou algo que parece não estar claro, não sente e se culpe - reclame!

## \* CÓPIA

-----

Esse tutorial descende de uma longa linha de tutoriais do Emacs iniciando com um escrito por Stuart Cracraft para o Emacs original.

Essa versão do tutorial foi originalmente traduzida por Marcelo Toledo <marcelo@gnu.org> e como o GNU Emacs, tem um copyright, e vem com uma permissão de distribuição de cópias nas seguintes condições:

Copyright (C) 2004, 2005, 2006, 2007, 2008, 2009 Free Software Foundation, Inc.

Permissão é garantida a qualquer um para fazer ou distribuir cópias integrais deste documento como recebido, em qualquer meio, deixando disponível e preservados o aviso de copyright e o aviso de permissão, e o distribuidor garante que a permissão para distribuições posteriores permitam o mesmo que esta nota.

Permissão garantida para distribuir versões modificadas deste documento, ou partes dele, sobre as condições acima, sempre que elas tenham uma nota visível de quem foi o último a alterá-lo.

As condições para copiar o Emacs especificamente são mais complexas, mas no mesmo espírito. Por favor, leia o arquivo COPYING e dê cópias do GNU Emacs para seus amigos. Ajude a erradicar o obstrucionismo de software ("proprietário") usando, escrevendo, e compartilhando software livre!

```
;;; Local Variables:
;;;   coding: latin-1
;;;   sentence-end-double-space: nil
;;; End:
```

```
;;; arch-tag: 8ea256d6-2c4a-49ee-ac08-0ea6ef25bf8d
```