



## *Guia rápido de uso do VIM*

*100 comandos para você se tornar um ninja nesse famoso editor*

O Vim é um editor muito prático para qualquer situação de edição de textos. A contrapartida, porém, é que o editor costuma ter uma curva de aprendizagem maior, por isso é fato bastante comum ser mais apreciado por programadores ou especialistas que desprendem muitas horas do seu dia editando textos técnicos. Pensando nisso a Linux Solutions traz para você esse guia com 100 comandos práticos que vai incrementar sua produtividade ao usar esse poderoso editor de textos.

## *O modo de comandos do vim*

*Quando você abre o Vim pela primeira vez, você está no modo de comando. O modo de comando permite navegar na tela, excluir e copiar texto entre muitos outros. O modo de inserção, por outro lado, como está implícito, permite que os usuários insiram texto no console do Vim. Você pode usar a tecla ESC para voltar ao modo de comando depois de terminar a digitação.*

### *I*

Pressionar “i” no console coloca você no modo de inserção. Nesse modo você pode começar a digitar seu texto de fato, o texto irá aparecer na frente do cursor.

### *a*

É um comando normalmente usado no Vim que você pode usar para inserir texto. Os textos serão anexados logo após o cursor.

### *i*

Quer colocar seus textos no início da linha atual? O comando “I” permite inserir texto diretamente no local desejado.

### *A*

Utilize o comando “A” para anexar texto no final da minha linha atual. Também se enquadra nos comandos do Vim usados para inserir texto.

### *O*

Para criar uma nova linha após a linha atual utilize o comando “o”.

### *O*

Da mesma forma que o comando anterior o “O” maiusculo cria uma nova linha antes da linha atual.

### *gf*

Você pode usar este comando para abrir seus arquivos sob o cursor.

### *gi*

O comando “gi” reinicia o modo de inserção no último ponto de inserção que você usou.

### *wq*

Digitar um texto não importa se você não pode salvá-lo para uso futuro. Digite o comando “wq” no modo de comando para salvar seu arquivo e sair do Vim.

### *q!*

Para sair sem salvar o texto digitado, basta usar o comando “q!”.

### *h*

Use o comando “h” para navegar para o lado esquerdo do console do Vim. Isso é análogo à seta esquerda do seu teclado.

### *I*

Os comandos “I” ajudam você a navegar para a direita e é um substituto para a tecla de seta para a direita.

## ***k***

Use o comando “k” minúsculo para se mover para cima. Semelhante à seta para cima.

## ***j***

O comando “j” leva você para baixo e é um substituto para a tecla de seta para baixo.

## ***H***

Usando “H” como comando leva o cursor para a região mais alta da sua tela.

## ***M***

Use “M” para colocar o cursor no meio da sua tela do Vim.

## ***L***

O comando “L” coloca o cursor no final da tela.

## ***H L M***

Os comandos H M L Vim são úteis sempre que você quiser chegar a uma parte aproximada do seu console Vim.

## ***O ou Home***

Digitar “O ou a tecla <HOME>” levará você ao início da linha.

## ***^***

Este é um dos comandos mais usados no Vim para chegar ao primeiro caractere não vazio em uma linha.

## ***\$***

Este comando levará o cursor no final da linha atual.

## ***b***

O comando “b” permite voltar pelos tokens.

## ***w***

Este comando ajuda você a avançar pelos tokens.

## ***B***

Este comando permite que você volte pelas palavras.

## ***W***

Este comando permitirá que você avance por palavras.

*Um token é qualquer seqüência de dígitos, letras ou sublinhados; enquanto uma palavra é qualquer seqüência não em branco que é separada por espaço em branco*

## ***ctrl+u***

Este comando Vim é basicamente um comando Page Up. No entanto, ele move a tela pela metade e mantém a posição atual do cursor.

## ***ctrl+d***

Funciona da mesma maneira que o comando Page Down, no entanto ele move a tela pela metade e mantém a posição atual do cursor.

## ***G***

Prefixe o comando “G ” com um número para pular diretamente para um número de linha específico. Suponha que, se você tiver a numeração de linha ativada e estiver escrevendo um script de shell, poderá ir para a décima linha simplesmente digitando 10G.

#

Dado que # é o número de uma linha específica, inserir este comando o levará diretamente para essa linha. É um dos comandos mais usados do Vim para pular de uma linha para outra.

“

Este comando (“) leva você aonde quer que você estivesse.

)

Você pode ir para o início da frase seguinte com este comando.

(

Ir para no início da sentença anterior.

}

Usado para ir ao início do próximo bloco de texto.

{

Leva o cursor no início do bloco de texto anterior.

## Comandos do Vim para edição de textos

*O Vim combina um número significativo de recursos de edição poderosos e flexíveis que permitem que os usuários editem seus documentos sem problemas enquanto abrem caminho para um fluxo de trabalho versátil. O poder do Vim vem do fato de que você pode combinar muitos desses comandos do Vim para resolver suas necessidades personalizadas. Esta seção do guia aborda alguns comandos elementares do Vim usados na edição diária de texto. Continue lendo para descobrir mais comandos úteis do Vim para edição.*

### ***yy***

O comando “yy” permite copiar uma linha inteira.

### ***yw***

Este é um dos comandos Vim mais utilizados para copiar uma palavra no editor.

### ***y\$***

Dá aos usuários a capacidade de copiar textos da posição atual do cursor até o final de uma linha.

### ***v***

O comando “v” pode ser usado para destacar um único caractere de cada vez em combinação com as teclas de movimento (setas / h j k l).

### ***V***

Quase o mesmo que o anterior, mas em vez de uma palavra, este comando irá destacar uma linha inteira.

### ***p***

Você pode colar o conteúdo de sua área de transferência para o registro não nomeado com este pequeno comando prático.

### ***d***

Este é um dos comandos mais usados do Vim para excluir textos destacados.

### ***dd***

Quer apagar uma linha inteira com um único comando Vim? O comando dd é especialmente criado para este propósito.

### ***dw***

Você pode excluir uma única palavra rapidamente com o comando dw.

### ***D***

Um dos comandos Vim mais poderosos de todos os tempos, o comando D apaga tudo, desde a posição atual do cursor até o final da linha.

### ***d0***

Este comando é usado para excluir tudo, desde a posição atual do seu cursor até o começo da linha.

### ***dgg***

Você pode usar esse comando para excluir tudo, desde a posição atual do seu cursor até o começo do arquivo.

## ***dG***

Este comando eliminará tudo, desde a posição atual do seu cursor até o final do arquivo.

## ***x***

Use o comando “x” sempre que precisar excluir um único caractere.

## ***u***

Utilize o comando “u” para desfazer a última operação. A combinação com um postfix permite que os usuários desfaçam várias ações. Então, u # desfaz o último # número de ações.

## ***ctrl+r***

Use esse comando para refazer a última operação de desfazer.

•

O comando ponto “.” É um daqueles comandos úteis do Vim que diminui significativamente a sua carga de trabalho, repetindo a última ação sempre que você precisar de tal funcionalidade.

## ***cc***

Use o comando “cc” para alterar a (s) linha (s) limpando e entrando no modo de inserção ao mesmo tempo. O cursor é colocado no nível atual de recuo.

## Comandos úteis do Vim para substituir textos

o Vim possui outro modo chamado modo visual. Você pode destacar e alterar uma parte específica do seu documento de texto nesse modo. Usando um conjunto selecionado de comandos do Vim você pode obter o mesmo resultado, apenas um pouco mais rápido. Abaixo estão alguns dos meus comandos favoritos do Vim para mudar os textos na hora.

### ***r***

O comando ***r*** é uma ferramenta muito útil quando se trata de mudar um único caractere. Siga-o com `<char>` e ele mudará o caractere atual sob o cursor com `<char>`.

### ***R***

A capital ***R*** abre o modo de inserção, mas em vez de inserir textos, você pode substituí-los por esse comando.

### ***~***

O comando ***~*** é bastante útil quando você precisa alterar o caso de um caractere em seu documento. Siga-o com um número para inverter esses caracteres.

### ***t<char>***

Digite ***t <char>*** para selecionar até, mas não incluindo o próximo `<char>` em uma linha específica.

### ***f<char>***

Pressione ***f <char>*** para selecionar até e incluindo o próximo `<char>` em uma linha.

### ***i<char>***

Digite ***i <char>*** para selecionar tudo o que fica entre dois `<char>` consecutivos.

### ***a<char>***

Este comando é idêntico ao anterior, mas inclui o `<char>` em ambas as extremidades do texto.

## Comandos Vim mais utilizados para pesquisar em um documento.

A funcionalidade de pesquisa é de maior importância para qualquer editor de texto versátil do Linux. O Vim oferece comandos excepcionalmente poderosos para pesquisar em um documento. É uma das áreas mais poderosas deste inovador editor de texto. Dê uma olhada nos comandos abaixo do Vim para pesquisar o documento com eficiência.

**/**

Basta digitar “/” e segui-lo com os textos que você deseja que o Vim procure por você e observe o canto inferior do console.

**^c**

A opção \c, quando canalizada para o comando search “/”, permite que os usuários pesquisem textos de maneira sensível a maiúsculas e minúsculas.

**?<pattern>**

Este é um dos comandos mais úteis do Vim para procurar textos anteriores por um dado <padrão>.

**n**

O comando “n” procura na direção da sua última pesquisa. Use este comando se você souber em qual direção seu item de pesquisa está.

**N**

Quase idêntico ao comando acima, mas procura na direção oposta à sua última pesquisa.

**:%s/[pattern]/[replacement]/g**

O comando acima utiliza a expressão regular para procurar todas as ocorrências de [padrão] e substituí-lo por [substituição] sem solicitar confirmação.

**:%s/[pattern]/[replacement]/gc**

Igual ao comando anterior, mas pede confirmação antes de substituir cada instância de [padrão] por [substituição].

**:s/[pattern]/[replacement]/g**

Em vez de substituir cada instância de [padrão] no seu arquivo, este comando Vim substituirá apenas os [padrão] que estão na linha atual com [substituição].

**:bufdo /[pattern]**

Este é um dos poderosos comandos do Vim que permitem que os usuários procurem por [padrão] em cada buffer atualmente aberto. Isso aumentará sua produtividade e diminuirá significativamente o tempo de pesquisa.

**:g/string/d**

Esse é um dos comandos úteis do Vim que serão úteis sempre que você quiser excluir todas as linhas que contenham string do seu documento.

## Trabalhando com vários arquivos no Vim

A capacidade de dividir a tela ao trabalhar com vários arquivos ao mesmo tempo é algo que o Vim vem oferecendo antes mesmo de o conceito da tela dividida se tornar mainstream.

### ***:sp [nomearquivo]***

Use este comando para criar um novo arquivo e dividir a tela do console horizontalmente para mostrar os dois buffers diferentes.

### ***:vsp [filename]***

Divide verticalmente a tela do console para mostrar os dois buffers diferentes.

### ***:bn***

Este comando Vim irá mudar seu editor para o próximo buffer.

### ***:bp***

Idêntico ao comando anterior, mas alterna para o buffer anterior em vez de ir para o próximo.

### ***:bd***

Use este comando Vim ao fechar um buffer específico. Certifique-se de salvar seus dados usando os comandos apropriados do Vim.

### ***:ls***

Este comando irá apresentar aos usuários uma lista de todos os buffers abertos.

### ***ctrl+ws***

Divide as janelas do Vim horizontalmente.

### ***ctrl+wv***

Divide as janelas do Vim verticalmente.

### ***ctrl+ww***

Utilize este comando para alternar entre várias janelas diretamente do modo de comando.

### ***ctrl+wq***

Use este comando Vim para sair de uma determinada janela.

### ***ctrl+wh***

Este comando irá mover a localização do seu cursor para a janela da esquerda.

### ***ctrl+wL***

Este comando apontará para a janela da direita.

### ***ctrl+wj***

Use este comando sempre que quiser mover uma janela abaixo da sua existente.

### ***ctrl+wk***

Leva o cursor para a janela acima da atual.

# Comandos úteis do Vim ao trabalhar com várias guias

*O Vim oferece uma variedade de comandos do Vim para abrir e navegar em uma infinidade de guias diretamente do terminal.*

## ***:tabnew***

Você pode usar o comando: `tabnew` para criar uma nova guia e trabalhar com algum outro documento sem deixar o arquivo atual.

## ***gt***

O comando “`gt`” mostrará a próxima guia aberta por você.

## ***:tabfirst***

Mostra a primeira guia que você abriu em uma sessão específica.

## ***:tablast***

Mostra a última guia em uma seção específica.

## ***tabm n(posição)***

Reorganiza as guias existentes

## ***tabdo %s/foo/bar/g***

Utilize esse comando para executar um comando em todas as guias abertas ao mesmo tempo.

## ***:tab ball***

Coloca cada arquivo aberto em abas.

## ***:new abc.txt***

Permite abrir um novo arquivo chamado `abc.txt` em uma nova janela sem sair do documento atual.

## Comandos Diversos do Vim

Como o Vim oferece muitos comandos do Vim, não podemos cobrir cada um deles categoricamente. É por isso que nesta seção teremos alguns comandos úteis do Vim que são usados por usuários veteranos para uma ampla variedade de propósitos.

***:W*** Pressionar este comando Vim no modo de comando salva seu documento atual.

***:q*** Este comando sairá da sessão atual sem salvar suas alterações.

### ***:help [comando]***

O comando de ajuda realiza uma operação de pesquisa no comando que você digitou e mostra informações relevantes diretamente no console.

### ***:e [file]***

Esse comando abrirá um arquivo chamado [file] e criará um novo caso ele não exista em seu sistema de arquivos.

### ***:w [nomedoarquivo]***

Use este comando para salvar seu documento existente diretamente em um novo arquivo chamado [nomedoarquivo].

### ***:stop***

Escrever este comando no modo de comando suspenderá sua sessão atual do Vim. Você também pode fazer isso pressionando ctrl + z ao mesmo tempo.

### ***:browse e***

Use este comando sempre que você quiser invocar o explorador de arquivos gráficos a partir do seu console Vim

***:%!fmt*** Escrevendo este comando irá alinhar cada linha do seu arquivo atual.

### ***!}fmt***

Use isso sempre que precisar alinhar todas as linhas na posição atual do cursor.

### ***:set autoindent***

Este é um dos comandos mais usados do Vim que você usará durante seu tempo com o Vim. Ele define o autoindent para sua sessão atual.