

Contents

Comandos e Funcionalidades Básicas do Vim e NeoVim	2
Índice	2
1. A Filosofia Modal do Vim/NeoVim	3
Os Quatro Modos Principais	3
Atalhos para Entrar no Modo de Inserção	4
2. Gerenciamento de Sessão e Arquivos	4
Comandos Essenciais de Sessão	4
Explicações Detalhadas	4
3. Edição Básica e Navegação	5
Operações de Cópia, Recorte e Cola	5
.	5
Notas Importantes	6
4. Busca e Substituição	6
Comandos de Busca	6
Comandos de Substituição	7
Explicações Detalhadas	7
5. Navegação por Saltos (Jump List)	7
Comandos de Jump List	7
O que Conta como um “Salto”?	8
Exemplo Prático	8
6. Gerenciamento de Buffers	8
O que é um Buffer?	8
Comandos de Gerenciamento de Buffers	8
Explicações Detalhadas	9
7. Janelas (Splits)	9
O que é uma Janela (Split)?	9
Criando Splits	10
Navegando Entre Janelas	10
Gerenciando Janelas	10
Exemplos Práticos	10
8. Redimensionamento de Janelas	11
Ajustando Largura e Altura	11
Usando Números para Ajustes Maiores	11
Abrindo Netrw com Tamanho Específico	11
Exemplo Prático	11
9. Abas (Tabs)	12
O que é uma Aba (Tab)?	12
Criando e Abrindo Abas	12
Navegando Entre Abas	12

Gerenciando Abas	12
Exemplos Práticos	13
10. Netrw: Gerenciamento de Arquivos	13
Abrindo o Netrw	13
Operações no Netrw	13
Exemplos Práticos	14
11. Netrw: Marcação e Cópia de Arquivos	14
Marcar Arquivos (Pré-requisito)	14
Copiar Arquivos Marcados	15
Processo Completo de Cópia	15
Exemplo Prático	15
12. Netrw: Movimentação de Arquivos	16
Mover Arquivos Marcados	16
Processo Completo de Movimentação	16
Exemplo Prático	17
Avisos Importantes	17
13. Configurações Essenciais (Bônus)	18
Configurações Básicas	18
Exemplo de Arquivo .vimrc	18
Onde Salvar o Arquivo .vimrc	19
Como Recarregar as Configurações	19
Explicações Detalhadas	19
Resumo Rápido de Atalhos Essenciais	20
Modos	20
Edição	20
Busca e Navegação	21
Buffers	21
Janelas (Splits)	21
Abas	21
Netrw	22
Sessão	22
Dicas Finais	22
Referências	23

Comandos e Funcionalidades Básicas do Vim e NeoVim

Índice

1. A Filosofia Modal do Vim/NeoVim
2. Gerenciamento de Sessão e Arquivos

3. Edição Básica e Navegação
 4. Busca e Substituição
 5. Navegação por Saltos (Jump List)
 6. Gerenciamento de Buffers
 7. Janelas (Splits)
 8. Redimensionamento de Janelas
 9. Abas (Tabs)
 10. Netrw: Gerenciamento de Arquivos
 11. Netrw: Marcação e Cópia de Arquivos
 12. Netrw: Movimentação de Arquivos
 13. Configurações Essenciais (Bônus)
-

1. A Filosofia Modal do Vim/NeoVim

O Vim é um **editor modal**. Dominar a transição entre os modos é o primeiro passo para a produtividade extrema.

Os Quatro Modos Principais

Modo	Propósito	Como Entrar	Como Sair (Voltar ao Normal)
Normal	Navegação e Comandos	Esc (Padrão)	N/A
Inserção	Digitar Texto	i, a, o, O	Esc ou Ctrl+[
Visual	Seleção de Texto	v, V, Ctrl+V	Esc
Comando	Executar Comandos (:)	: ou /	Enter (Executa) ou Esc (Cancela)

Ação	Atalho (Modo Normal)	Explicação
------	----------------------	------------

Atalhos para Entrar no Modo de Inserção

Ação	Atalho (Modo Normal)	Explicação
Inserir Antes do Cursor	i	Insert.
Inserir Após o Cursor	a	Append.
Inserir em Nova Linha Abaixo	o	Open new line below.
Inserir em Nova Linha Acima	O	Open new line above.

2. Gerenciamento de Sessão e Arquivos

Comandos básicos para iniciar e finalizar o trabalho, garantindo a segurança dos seus arquivos.

Comandos Essenciais de Sessão

Ação	Comando por Extenso	Abreviação	Atalho (Modo Normal)
Entrar no Vim	vim <arquivo> (no terminal)	vi <arquivo>	N/A
Salvar	:write	:w	N/A
Sair	:quit	:q	N/A
Salvar e Sair	:writequit	:wq OU :x	ZZ
Sair sem Salvar	:quit!	:q!	ZQ

Explicações Detalhadas

- **:w** - Salva as alterações no arquivo atual, mas não fecha o editor.
- **:q** - Fecha o Vim. Só funciona se não houver alterações não salvas.
- **:wq** - Salva as alterações e fecha o editor. É o comando mais comum para finalizar o trabalho.
- **:x** - Similar a **:wq**, mas um pouco mais inteligente: só salva se houver alterações.
- **:q!** - Força a saída e **descarta** todas as alterações desde o último salvamento. Use com cuidado!
- **zz** - Atalho para **:wq**. Pressione z duas vezes no Modo Normal.

- **zq** - Atalho para **:q!**. Força a saída sem salvar.

3. Edição Básica e Navegação

Manipulação de texto com copiar, recortar, colar e histórico de alterações. Estes são os comandos fundamentais para edição eficiente de arquivos.

Operações de Cópia, Recorte e Cola

Ação	Atalho (Modo Normal)	Explicação
Copiar Linha	yy	(Y)ank - Copia a linha inteira.
Copiar Palavra	yw	Copia uma palavra.
Copiar Seleção	y (no Modo Visual)	Copia o texto selecionado.
Recortar Linha	dd	(D)ele te - Recorta a linha inteira (vai para o registrador).
Recortar Palavra	dw	Recorta uma palavra.
Recortar Seleção	d (no Modo Visual)	Recorta o texto selecionado.
Colar Após o Cursor	p	(P)aste - Cola o conteúdo do registrador após o cursor.
Colar Antes do Cursor	P	Cola o conteúdo do registrador antes do cursor.
Copiar para o Clipboard (sistema)	" + y	(Y)ank - copia o texto para o ambiente externo do editor (Ctrl + C).
Colar do Clipboard (sistema)	" + p	(P)aste - cola o texto do ambiente externo no editor (Ctrl + V).

Obs.: versões antigas do vim não suportam essa função ('Ctrl + C' e 'Ctrl + V').

Histórico de Alterações

Ação	Atalho (Modo Normal)	Explicação
Desfazer	u	(U)ndo - Desfaz a última alteração. Pressione múltiplas vezes para desfazer as anteriores.
Refazer	Ctrl + R	(R)edo - Refaz uma alteração desfeita com u.
Repetir Última Ação	.	Repete o último comando de edição. Extremamente útil para automação!

Notas Importantes

- **Recortar é o mesmo que Deletar no Vim:** Quando você usa dd ou d, o texto vai para o “registrador” (área de transferência do Vim), funcionando como um recorte.
- **O ponto (.) é poderoso:** Se você inseriu texto, deletou uma linha, ou fez qualquer alteração, pressione . para repetir exatamente a mesma ação.

4. Busca e Substituição

Ferramentas rápidas para encontrar padrões e realizar substituições em massa.

Comandos de Busca

Ação	Comando/Atalho	Explicação
Buscar para Frente	/palavra	Busca pela palavra. Pressione n para a próxima ocorrência, N para a anterior.
Buscar para Trás	?palavra	Busca no sentido contrário do arquivo.
Ir para Próxima Ocorrência	n	Pula para a próxima ocorrência da última busca.
Ir para Ocorrência Anterior	N	Pula para a ocorrência anterior da última busca.

Comandos de Substituição

Ação	Comando	Explicação
Substituir na Linha	<code>:s/antigo/novo</code>	Substitui a primeira ocorrência na linha atual.
Substituir Globalmente	<code>:%s/antigo/novo/g</code>	Substitui todas as ocorrências no arquivo.
Substituir com Confirmação	<code>:%s/antigo/novo/gc</code>	Pede confirmação (y/n/a/q) para cada substituição.

Explicações Detalhadas

- `/palavra` - Pressione `Enter` para executar a busca. O cursor pulará para a primeira ocorrência.
- `n` - Vai para a próxima ocorrência. Pressione múltiplas vezes para navegar por todas.
- `N` - Vai para a ocorrência anterior (sentido contrário).
- `:%s/antigo/novo/g` - O `%` significa “todo o arquivo”, `s` é “substitute”, e `g` é “global” (todas as ocorrências).
- `:%s/antigo/novo/gc` - Ao pressionar `y`, substitui; `n` pula; `a` substitui todas as restantes; `q` cancela.

5. Navegação por Saltos (Jump List)

O Vim mantém um histórico de “saltos” que você faz no arquivo. Use-o para navegar rapidamente.

Comandos de Jump List

Ação	Atalho (Modo Normal)	Explicação
Saltar para Posição Anterior	<code>Ctrl + O</code>	Volta para a posição do cursor antes do último “salto” (como o botão “Voltar” de um navegador).
Avançar para Posição Seguinte	<code>Ctrl + I</code>	Avança na lista de saltos (o oposto de <code>Ctrl+O</code>).

O que Conta como um “Salto”?

Um salto ocorre quando você:

- Realiza uma busca com `/` ou `?`
- Abre um novo arquivo com `:e` ou `:tabnew`
- Usa `gg` para ir ao início do arquivo
- Usa `G` para ir ao final do arquivo
- Usa `Ctrl+]` para ir para uma definição (com tags)

Exemplo Prático

1. Você está na linha 50 de um arquivo.
 2. Você busca `/função` e o cursor pula para a linha 200.
 3. Você abre outro arquivo com `:e outro_arquivo.txt`.
 4. Agora, pressione `Ctrl+O` para voltar à busca anterior (linha 200).
 5. Pressione `Ctrl+O` novamente para voltar à linha 50.
 6. Pressione `Ctrl+I` para avançar para a linha 200 novamente.
-

6. Gerenciamento de Buffers

Buffers são os arquivos carregados na memória. Gerencie-os para alternar rapidamente entre projetos e otimizar seu fluxo de trabalho.

O que é um Buffer?

Um **buffer** é uma área na memória que contém o conteúdo de um arquivo que você está editando. Quando você abre um arquivo com `vim arquivo.txt`, o Vim lê o conteúdo e o carrega em um buffer. Todas as suas edições são feitas nesse buffer, e o arquivo no disco só é modificado quando você salva.

Vantagem: Você pode ter vários buffers abertos ao mesmo tempo, cada um contendo um arquivo diferente, sem precisar fechar e reabrir o Vim.

Comandos de Gerenciamento de Buffers

Ação	Comando por Extenso	Abreviação	Atalho (Modo Normal)
Listar Buffers	:buffers	:ls	N/A
Ir para Próximo Buffer	:bnext	:bn	N/A
Ir para Buffer Anterior	:bprevious	:bp	N/A
Ir para Buffer por Número	:buffer <N>	:b <N>	N/A
Ir para Buffer por Nome	:buffer <nome>	:b <nome>	N/A
Ir para Buffer Alternativo	:buffer #	:b#	Ctrl + ^
Fechar Buffer Atual	:bdelete	:bd	N/A

Explicações Detalhadas

- **:ls** - Mostra todos os buffers abertos com seus respectivos números. Exemplo:
 - 1, 2, 3 são os números dos buffers.
 - % indica o buffer que está na janela atual.
 - # indica o buffer alternativo (o último que você visitou).
 - a indica que o buffer está ativo (carregado e visível).
- **:bn** - Vai para o próximo buffer na lista.
- **:bp** - Vai para o buffer anterior na lista.
- **:b 2** - Troca para o buffer número 2.
- **:b outro** - Troca para o buffer cujo nome comece com “outro” (use Tab para autocompletar).
- **ctrl+^** - Atalho rápido para alternar entre o buffer atual e o buffer alternativo. **Um dos atalhos mais úteis!**
- **:bd** - Fecha o buffer atual. Se o buffer tiver alterações não salvas, o Vim avisará. Use **:bd!** para forçar.

7. Janelas (Splits)

Divida a tela para ver e editar vários buffers simultaneamente. O prefixo para quase todos os comandos de janela é **ctrl+w**.

O que é uma Janela (Split)?

Uma **janela** (ou “split”) é uma área visível na tela onde você vê um buffer. Você pode ter múltiplas janelas abertas, cada uma mostrando um buffer diferente (ou diferentes partes do mesmo buffer).

Criando Splits

Ação	Comando por Extenso	Abreviação	Atalho (Modo Normal)
Split Horizontal	<code>:split <arquivo></code>	<code>:sp <arquivo></code>	<code>Ctrl+W s</code>
Split Vertical	<code>:vsplit <arquivo></code>	<code>:vsp <arquivo></code>	<code>Ctrl+W v</code>

Navegando Entre Janelas

Ação	Atalho (Modo Normal)	Explicação
Mover para Janela à Esquerda	<code>Ctrl+W h</code>	Move o cursor para a janela à esquerda.
Mover para Janela Abaixo	<code>Ctrl+W j</code>	Move o cursor para a janela abaixo.
Mover para Janela Acima	<code>Ctrl+W k</code>	Move o cursor para a janela acima.
Mover para Janela à Direita	<code>Ctrl+W l</code>	Move o cursor para a janela à direita.
Pular para Próxima Janela	<code>Ctrl+W w</code>	Pula para a próxima janela (circulando por todas).
Pular para Janela Anterior	<code>Ctrl+W p</code>	Pula para a janela anterior.

Gerenciando Janelas

Ação	Comando por Extenso	Abreviação	Atalho (Modo Normal)
Fechar Janela Atual	<code>:quit</code>	<code>:q</code>	<code>Ctrl+W q</code>
Manter Apenas Janela Atual	<code>:only</code>	N/A	<code>Ctrl+W o</code>

Exemplos Práticos

- **Abrir um arquivo em split horizontal:** `:sp arquivo.txt` (abre `arquivo.txt` em uma nova janela horizontal)
- **Abrir um arquivo em split vertical:** `:vsp script.js` (abre `script.js` em uma nova janela vertical)
- **Navegar entre splits:** Use `Ctrl+W + h/j/k/l` para mover entre as janelas como se fossem panes.

8. Redimensionamento de Janelas

Comandos para controlar o tamanho das janelas abertas. O prefixo é `Ctrl+W`.

Ajustando Largura e Altura

Ação	Atalho (Modo Normal)	Explicação
Aumentar Largura	<code>Ctrl+W ></code>	Aumenta a largura da janela atual em 1 coluna.
Diminuir Largura	<code>Ctrl+W <</code>	Diminui a largura da janela atual em 1 coluna.
Aumentar Altura	<code>Ctrl+W +</code>	Aumenta a altura da janela atual em 1 linha.
Diminuir Altura	<code>Ctrl+W -</code>	Diminui a altura da janela atual em 1 linha.
Igualar Tamanho	<code>Ctrl+W =</code>	Deixa todas as janelas com a mesma altura e largura.

Usando Números para Ajustes Maiores

Você pode prefixar os comandos com um número para fazer ajustes maiores:

- `10 Ctrl+W >` - Aumenta a largura em 10 colunas.
- `5 Ctrl+W +` - Aumenta a altura em 5 linhas.

Abrindo Netrw com Tamanho Específico

Ação	Comando	Explicação
Abrir Netrw com Largura Específica	<code>:30!explore</code>	Abre o Netrw em split vertical à esquerda com 30 colunas de largura.
Abrir Netrw com Largura Específica (Direita)	<code>:40Vexplore!</code>	Abre o Netrw em split vertical à direita com 40 colunas de largura.

Exemplo Prático

1. Você tem dois splits abertos lado a lado.

2. Você quer que o split da esquerda fique maior.
3. Pressione `Ctrl+W h` para mover para o split da esquerda.
4. Pressione `Ctrl+W >` múltiplas vezes para aumentar a largura, ou `10 Ctrl+W >` para aumentar 10 colunas de uma vez.

9. Abas (Tabs)

Abas são coleções de janelas, ideais para separar projetos ou tarefas. Navegue entre abas como em um navegador web.

O que é uma Aba (Tab)?

Uma **aba** é um layout de janelas. Cada aba pode conter seu próprio conjunto de splits. Isso é diferente de buffers (que são arquivos em memória) e janelas (que são visualizações na tela). As abas permitem organizar completamente diferentes contextos de trabalho.

Criando e Abrindo Abas

Ação	Comando por Extenso	Abreviação	Atalho (Modo Normal)
Nova Aba Vazia	<code>:tabnew</code>	N/A	N/A
Nova Aba com Arquivo	<code>:tabnew <arquivo></code>	<code>:tabe <arquivo></code>	N/A

Navegando Entre Abas

Ação	Comando por Extenso	Abreviação	Atalho (Modo Normal)
Ir para Próxima Aba	<code>:tabnext</code>	<code>:tabn</code>	<code>gt</code>
Ir para Aba Anterior	<code>:tabprevious</code>	<code>:tabp</code>	<code>gT</code>
Ir para Aba Número N	<code>:tabnext <N></code>	<code>:tabn <N></code>	<code><N>gt</code>
Ir para Primeira Aba	<code>:tabfirst</code>	N/A	N/A
Ir para Última Aba	<code>:tablast</code>	N/A	N/A

Gerenciando Abas

Ação	Comando por Extenso	Abreviação	Atalho (Modo Normal)
Fechar Aba Atual	<code>:tabclose</code>	<code>:tabc</code>	N/A
Manter Apenas Aba Atual	<code>:tabonly</code>	N/A	N/A
Mover Aba para Posição	<code>:tabmove <posição></code>	N/A	N/A

Exemplos Práticos

- **Abrir um novo projeto em uma aba:** `:tabnew projeto2.txt`
- **Navegar entre abas:** Use `gt` para próxima e `gT` para anterior.
- **Ir para a terceira aba:** `3gt`
- **Mover a aba atual para o final:** `:tabmove`
- **Mover a aba atual para ser a primeira:** `:tabmove 0`

10. Netrw: Gerenciamento de Arquivos

Explorador de arquivos nativo do Vim. Crie, renomeie, delete, copie e mova arquivos e diretórios diretamente no editor.

Abrindo o Netrw

Ação	Comando	Explicação
Abrir Netrw na Janela Atual	<code>:Explore</code> ou <code>:E</code>	Abre o Netrw mostrando o diretório atual.
Abrir Netrw em Split Vertical (Esquerda)	<code>:Lexplore</code> ou <code>:Lex</code>	Abre o Netrw em split vertical à esquerda (ótimo para sidebar).
Abrir Netrw em Split Vertical (Direita)	<code>:Vexplore</code> ou <code>:Vex</code>	Abre o Netrw em split vertical à direita.
Abrir Netrw em Split Horizontal	<code>:Sexplore</code> ou <code>:Sex</code>	Abre o Netrw em split horizontal.

Operações no Netrw

Os comandos abaixo são executados no Modo Normal dentro da janela do Netrw, com o cursor sobre o arquivo/diretório desejado.

Ação	Atalho (no Netrw)	Explicação
Criar Diretório	d	Pressione d e digite o nome do novo diretório.
Criar Arquivo	%	Pressione % e digite o nome do novo arquivo.
Renomear	R	Renomeia o arquivo/diretório sob o cursor. Digite o novo nome.
Deletar	D	Deleta o arquivo/diretório sob o cursor (pede confirmação).
Marcar Arquivo	mf	(Mark File) - Marca arquivo para operações em lote. Um * aparecerá.
Desmarcar Arquivo	mu	(Mark Unseen) - Remove a marcação do arquivo.

Exemplos Práticos

1. **Abrir o Netrw:** Digite :Lexplore para abrir como sidebar à esquerda.
2. **Criar um novo diretório:** Pressione d e digite novo_projeto.
3. **Criar um novo arquivo:** Pressione % e digite script.py.
4. **Renomear um arquivo:** Pressione R sobre o arquivo e digite o novo nome.
5. **Deletar um arquivo:** Pressione D sobre o arquivo e confirme.

11. Netrw: Marcação e Cópia de Arquivos

Operações avançadas de cópia de arquivos marcados no Netrw, permitindo gerenciamento eficiente de projetos.

Marcar Arquivos (Pré-requisito)

Ação	Atalho (no Netrw)	Explicação
Marcar Arquivo	mf	(Mark File) - Marca o arquivo/diretório sob o cursor. Um * aparecerá ao lado do nome.
Desmarcar Arquivo	mu	(Mark Unseen) - Remove a marcação do arquivo (remove o *).

Nota: Você pode marcar **múltiplos arquivos** antes de executar uma operação de cópia ou movimentação.

Copiar Arquivos Marcados

Ação	Atalho (no Netrw)	Explicação
Copiar Arquivos Marcados	mc	(Mark Copy) - Copia os arquivos marcados para o diretório atual.

Processo Completo de Cópia

1. **Posicione o cursor** sobre os arquivos/diretórios que deseja copiar.
2. **Marque cada um** pressionando mf. Um * aparecerá ao lado de cada nome marcado.
3. **Navegue até o diretório de destino** usando as setas ou pressionando Enter para abrir pastas.
4. **Pressione mc** (Mark Copy). O Netrw pedirá confirmação para copiar os arquivos marcados para o local atual.
5. **Pressione u** para atualizar a visualização e ver os arquivos copiados.

Exemplo Prático

```

1 Estrutura Original:
2 src/
3   ├── arquivo1.txt
4   ├── arquivo2.txt
5   └── arquivo3.txt
6
7 backup/
8   └── (vazio)
9

```

```

10 Processo:
11 1. Abra o Netrw: :Lexplore
12 2. Navegue para src/
13 3. Marque arquivo1.txt com mf
14 4. Marque arquivo2.txt com mf
15 5. Navegue para backup/
16 6. Pressione mc para copiar
17 7. Pressione u para atualizar
18
19 Resultado:
20 src/
21 |— arquivo1.txt
22 |— arquivo2.txt
23 |— arquivo3.txt
24
25 backup/
26 |— arquivo1.txt
27 |— arquivo2.txt

```

12. Netrw: Movimentação de Arquivos

Operações de movimentação de arquivos marcados no Netrw. Inclui avisos importantes sobre a irreversibilidade das operações.

Mover Arquivos Marcados

Ação	Atalho (no Netrw)	Explicação
Mover Arquivos Marcados	mm	(Mark Move) - Move os arquivos marcados para o diretório atual.

Processo Completo de Movimentação

1. **Posicione o cursor** sobre os arquivos/diretórios que deseja mover.
2. **Marque cada um** pressionando **mf**. Um ***** aparecerá ao lado de cada nome marcado.
3. **Navegue até o diretório de destino** usando as setas ou pressionando **Enter** para abrir pastas.
4. **Pressione mm** (Mark Move). O Netrw moverá os arquivos marcados para o local atual.
5. **Pressione u** para atualizar a visualização. Os arquivos desaparecerão do local original.

Exemplo Prático

```
1 Estrutura Original:
2 projeto_antigo/
3   └─ arquivo1.txt
4   └─ arquivo2.txt
5   └─ arquivo3.txt
6
7 projeto_novo/
8   └─ (vazio)
9
10 Processo:
11 1. Abra o Netrw: :Lexlore
12 2. Navegue para projeto_antigo/
13 3. Marque arquivo1.txt com mf
14 4. Marque arquivo2.txt com mf
15 5. Navegue para projeto_novo/
16 6. Pressione mm para mover
17 7. Pressione u para atualizar
18
19 Resultado:
20 projeto_antigo/
21   └─ arquivo3.txt
22
23 projeto_novo/
24   └─ arquivo1.txt
25   └─ arquivo2.txt
```

Avisos Importantes

Operações Irreversíveis Operações de arquivo no Netrw são **ações diretas no sistema de arquivos** e **NÃO podem ser desfeitas** com u (undo do Vim). Tenha cuidado ao deletar ou mover arquivos importantes!

Diferença: Undo do Vim vs. Sistema de Arquivos

- u** (undo) no Vim** desfaz apenas **edições de texto**, não operações de arquivo.
- Para recuperar arquivos deletados, você precisará usar ferramentas do sistema operacional (como `rm -r` no Linux ou Trash no macOS) ou backups.

Dica de Segurança Antes de executar operações em lote no Netrw, considere:

- Fazer um backup dos arquivos importantes.
- Usar um sistema de controle de versão (como Git) para rastrear mudanças.

- Testar a operação com um arquivo menos importante primeiro.

13. Configurações Essenciais (Bônus)

Opções de configuração que melhoram a experiência de uso do Vim. Adicione essas linhas ao seu arquivo `.vimrc`.

Configurações Básicas

Ação	Comando de Configuração	Explicação
Mostrar Comandos Parciais	<code>set showcmd</code>	Exibe teclas de comandos incompletos no canto inferior direito. Muito útil para aprender.
Numeração de Linha	<code>set number</code>	Mostra o número de cada linha na margem esquerda.
Numeração Relativa	<code>set relativenumber</code>	Mostra a distância das outras linhas em relação à linha atual. Ótimo para movimentos como <code>10j</code> .
Realçar Resultados da Busca	<code>set hlsearch</code>	Destaca todas as ocorrências encontradas em uma busca com <code>/</code> ou <code>?</code> .
Tamanho Padrão do Netrw	<code>let g:netrw_winsize = 30</code>	Define a largura padrão do Netrw para 30 colunas quando aberto com <code>:Lexlore</code> .

Exemplo de Arquivo `.vimrc`

```
1 " =====
2 " Configurações Essenciais do Vim
3 " =====
4
5 " Mostrar comandos parciais
6 set showcmd
7
8 " Destacar linha e coluna na posição do cursor.
9 set cursorcolumn
10 set cursorline
11
```

```

12 " tema nativo com bons contrastes.
13 colorscheme slate " Se essa não estiver disponível, pressione Tab no comando ':colorscheme '
    para ver os temas disponíveis.
14
15
16 " Numeração de linhas
17 set number
18 set relativenumber
19
20 " Realçar busca
21 set hlsearch
22
23 " Tamanho padrão do Netrw
24 let g:netrw_winsize = 30

```

Onde Salvar o Arquivo .vimrc

Localizações por Sistema Operacional

Sistema	Caminho
Linux/macOS (Vim)	~/.vimrc
Linux/macOS (NeoVim)	~/.config/nvim/init.vim
Windows (Vim)	%USERPROFILE%_vimrc
Windows (NeoVim)	%USERPROFILE%\AppData\Local\nvim\init.vim

Como Recarregar as Configurações

Depois de editar o arquivo .vimrc, você pode recarregar as configurações de duas formas:

1. **Dentro do Vim:** Digite `:source ~/.vimrc` (ou o caminho correto para seu sistema).
2. **Feche e reabra o editor:** As configurações serão aplicadas automaticamente na próxima inicialização.

Explicações Detalhadas

- **set showcmd** - Quando você digita um comando incompleto (como `d` esperando um movimento), o Vim mostra o que você digitou no canto inferior direito. Muito útil para aprender.
- **set number** - Mostra números de linha. Útil para referência e para usar comandos como `:10` (ir para a linha 10).
- **set relativenumber** - Mostra números relativos. A linha atual mostra 0, a linha acima mostra 1, a linha abaixo mostra 1, etc. Isso torna movimentos como `5j` (descer 5 linhas) muito mais intuitivos.

- **set hlsearch** - Quando você busca algo com `/`, todas as ocorrências são destacadas. Pressione `:nohlsearch` para desativar o destaque temporariamente.
 - **let g:netrw_winsize = 30** - Define a largura padrão do Netrw. Quando você abre com `:Lexlore`, o Netrw ocupará 30 colunas de largura.
-

Resumo Rápido de Atalhos Essenciais

Modos

- `i` - Entrar no Modo de Inserção (antes do cursor)
- `a` - Entrar no Modo de Inserção (após o cursor)
- `v` - Entrar no Modo Visual (caractere)
- `V` - Entrar no Modo Visual (linha)
- `Ctrl+V` - Entrar no Modo Visual (bloco)
- `Esc` - Voltar para o Modo Normal

Edição

- `yy` - Copiar linha
- `dd` - Deletar/Recortar linha
- `"+Y` - Copiar texto selecionado para o clipboard
- `"+p` - Colar texto do clipboard
- `p` - Colar após
- `P` - Colar antes
- `u` - Desfazer
- `Ctrl+R` - Refazer
- `.` - Repetir última ação

Busca e Navegação

- /palavra - Buscar para frente
- ?palavra - Buscar para trás
- n - Próxima ocorrência
- N - Ocorrência anterior
- Ctrl+0 - Saltar para posição anterior
- Ctrl+I - Avançar para posição seguinte

Buffers

- :ls - Listar buffers
- :bn - Próximo buffer
- :bp - Buffer anterior
- :b <N> - Buffer por número
- Ctrl+^ - Alternar entre buffers

Janelas (Splits)

- Ctrl+W s - Split horizontal
- Ctrl+W v - Split vertical
- Ctrl+W h/j/k/l - Navegar entre splits
- Ctrl+W q - Fechar split
- Ctrl+W = - Igualar tamanho

Abas

- :tabnew - Nova aba
- gt - Próxima aba
- gT - Aba anterior
- <N>gt - Ir para aba N
- :tabc - Fechar aba

Netrw

- `:Explore` - Abrir Netrw
- `:Lexlore` - Netrw em split vertical (esquerda)
- `d` - Criar diretório
- `%` - Criar arquivo
- `R` - Renomear
- `D` - Deletar
- `mf` - Marcar arquivo
- `mc` - Copiar marcados
- `mm` - Mover marcados

Sessão

- `:w` - Salvar
 - `:q` - Sair
 - `:wq` ou `:x` - Salvar e sair
 - `:q!` - Sair sem salvar
 - `zz` - Salvar e sair (atalho)
 - `zQ` - Sair sem salvar (atalho)
-

Dicas Finais

1. **Comece pelo básico:** Domine os modos e os comandos essenciais antes de explorar funcionalidades avançadas.
2. **Use `showcmd`:** Adicione `set showcmd` ao seu `.vimrc` para ver os comandos que você está digitando. Isso acelera o aprendizado.
3. **Pratique diariamente:** Use o Vim para editar todos os seus arquivos por uma semana. Sua memória muscular se desenvolverá rapidamente.
4. **Use `.` para repetir:** O atalho `.` é extremamente poderoso. Se você fez uma edição, pressione `.` para repeti-la na próxima linha.

5. **Organize com abas e splits:** Use abas para separar projetos e splits para ver múltiplos arquivos lado a lado.
 6. **Netrw é poderoso:** Você não precisa de plugins para gerenciar arquivos. O Netrw nativo é suficiente para a maioria das tarefas.
 7. **Customize seu .vimrc:** Comece com as configurações básicas e adicione mais conforme suas necessidades crescerem.
 8. **Use Jump List:** `Ctrl+O` e `Ctrl+I` são seus melhores amigos para navegar rapidamente entre posições.
-

Referências

- Documentação Oficial do Vim
 - Documentação Oficial do NeoVim
 - Vim Tips Wiki
-

Última atualização: Novembro de 2025

Autor: Tutorial Completo de Vim/NeoVim

Licença: Livre para uso e distribuição