

Minicurso Vim

André William Régis

UFSC

2023

Material de apoio

```
$ git clone --recursive  
https://github.com/AndreKuru/Minicurso-Vim.git
```

practical-vim-examples

Principais modos

Normal: navegar e editar

Insert: inserção de texto

Visual: editar através de seleção

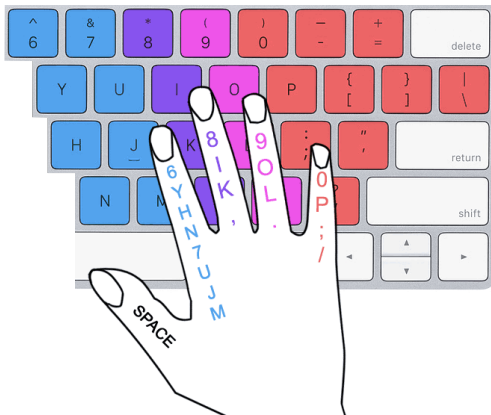
Command-line: retentor de usuários de primeira viagem

Movimentação básica

j: ↓

k: ↑

h: ←



Jogo para praticar:
vim-adventures.com

Imagem adaptada.
Disponível em: Wikimedia

Quando não se precisa manter a mão no mouse

u : undo change

dd : delete line

yy : yank line

p : put text



Imagem não adaptada.
Disponível em:
Unsplash

Mas se quiser pode

:set mouse=a

Experimentando cada modo

i: switch to Insert mode

v: switch to Visual mode

:: switch to Command-Line mode

Como sair do vim?

:q : quit

:q! : quit without writing

:w : write

:wq : write and quit

:x : write (if needed) and quit

Comando ponto

•: "Repeat last change [...]"

Mudanças comuns

x: delete [count] char(s)

d{motion}: delete

s: substitute [count] char(s)

c{motion}: change

[]: "[...] are optional."

{ }: "[...] must appear, but
which can take a number of
different values."

[count]: "An optional number
that may precede the
command to multiply or
iterate the command."

{motion}: "A command that
moves the cursor."

Movimentação na linha

0: start of the line

\$: end of the line

^: start non-blank of the line

g_: end non-blank of the line

Movimentação através de palavras

w: word forward

e: word end forward

b: word backward

ge: word end backward

word: "[...]
letters, digits and underscores,
or a sequence of other non-blank
characters [...]"

blank characters: "space and tab"

Movimentação através de PALAVRAS

W: WORD foward

E: WORD end foward

B: WORD backward

gE: WORD end
backward

WORD: "[...]
sequence of non-blank characters
[...]"

blank characters: "space and tab"

Buscas dentro da linha

f: find char
to the right

t: till before char
to the right

F: find char
to the left

T: till after char
to the left

;: "Repeat latest f, t, F or T [...]"

,: "Repeat latest f, t, F or T
in opposite direction [...]"

Buscas no arquivo

/: search forward for
the pattern

?: search backward for
the pattern

*****: search forward for
the nearest word

#: search backward for
the nearest word

n: "Repeat latest "/"or "?"[...]"

N: "Repeat latest "/"or "?"
in opposite direction [...]"

Operadores

c: change

d: delete

y: yank

>: shift right

<: shift left

=: autoindent

g~: swap case

gu: make lowercase

gU: make uppercase

Teclas especiais

<Enter> : Enter

<CR> : Carriage Return

<Esc> : Escape

<Up> : Up arrow key

<S-...> : Shift + ...

<S-Down> : Shift + Down arrow key

<A-...> : Alt + ...

Ctrl-A : Ctrl + a

<C-...> : Ctrl + ...

<C-x> : Ctrl + x

Aritmética simples

<C-a>: add [count] to the number at or after the cursor

<C-x>: subtract [count] to the number at or after the cursor

Nem tudo se transforma

i: insert text

a: append text

I: insert text at the start of the line

A: append text at the end of the line

o: obtain new line below

O: obtain new line
above

Correções rápidas

<C-h>: delete back one character (**<BS>**)

<C-w>: delete back one word

<C-u>: delete back to the start of line

Transições

<Esc>: switch to Normal mode

<C-]>: switch to Normal mode

<C-o>: switch to Insert Normal mode

Acessar registradores

<C-r>{register} : "Insert the contents of a register."

{register} :

+ : the clipboard contents

***** : the clipboard contents (X11: primary selection)

0 : contains the text from the most recent yank command

1-9 : contains the text deleted in ascending order by
the most recent delete or change command

= : the expression register

Inserir caracteres por código

`<C-v>191` : ÿ

`<C-k>{char1}{char2}`

`<C-v>u00bf` : ÿ

`<C-k>?l` : ÿ

`<C-v>u00b0` : °

`<C-k>12` : $\frac{1}{2}$

`<C-v>{nondigit}` : e.g. `<Home>`

`ga` : print the ascii value

Mais um modo

R: Replace mode

gR: Virtual Replace mode

r: replace single character

gr: virtual replace single character

Cada modo uma orientação

v: switch character-wise
Visual mode

V: switch line-wise
Visual Line mode

<C-v>: switch block-wise
Visual Block mode

<Esc>: switch to Normal mode

<C-[>: switch to Normal mode

gv: re-select last visual selection

o: go to other end of selected text

Seleção entre delimitadores

a) or **ab**: a pair of (parentheses)

a]: a pair of [brackets]

a} or **aB**: a pair of {braces}

a>: a pair of <>

a': a pair of 'single quotes'

a`: a pair of 'backticks'

a": a pair of ""double quotes"

at: a pair of tags

Seleção entre delimitadores

i) or **ib**: inside of (parentheses)

i]: inside of [brackets]

i} or **iB**: inside of {braces}

i>: inside of <>

i': inside of 'single quotes'

i': inside of 'backticks'

i": inside of "double quotes"

it: inside of tags

Seleção de objeto

aw : around of **w**ord

iw : inside of **w**ord

aW : around of **W**ORD

iW : inside of **W**ORD

as : around of **s**entence

is : inside of **s**entence

ap : around of **p**aragraph

ip : inside of **p**aragraph

Falando do ex

:print : print current line

:3print : print line 3

:2,4print : print lines 2 to 4

:'<,>print : print lines selected from visual mode

Comandos básicos

:*[range]*print

:*[range]*join

:*[range]*copy {address}

:*[range]*move {address}

***[range]* : {address}[, {address}]**

{address} :

number : absolute line number

. : current line

\$: last line in the file

% : current file

/ {pattern} [/] : next line [...]

? {pattern} [?] : previous line [...]

Comandos básicos

```
:[range]delete [x]
```

```
:[range]yank [x]
```

```
:[range]put [x]
```

```
[x] : "[into register x]"
```

```
a,b,c,d,e,f,g,h,  
i,j,k,l,m,n,o,p,  
q,r,s,t,u,v,w,x,  
y,z
```


Os mais populares

```
:[range]normal {address}
```

```
:[range]substitute/{pattern}/{string}/{flags}
```

```
:[range]global/{pattern}/[cmd]
```

[cmd] : command from command line

Repetição é a chave

@: execute the last ex command

@@: execute the last **@**

@{register}: execute the contents of register

q: record/stop macro

Comandos externos

`:shell` : start shell

`!!{cmd}` : execute `{cmd}` in the shell

Comandos externos integrados com o arquivo atual

`:read !{cmd}` : execute `{cmd}` in the shell
and put the output as text below cursor

`:[range]write !{cmd}` : execute `{cmd}` in the shell
with `[range]` lines as input

`:[range] !{filter}` : execute `{cmd}` in the shell
and put the output as text below cursor
with `[range]` lines as input

Rolar texto

<C-E>: scroll window
(Extra lines)

<C-Y>: scroll window
(Yrineu)

<C-D>: scroll window
Donwards

<C-U>: scroll window
Upwards

<C-F>: scroll window
Forwards

<C-B>: scroll window
Backwards

Saltando pelo texto

`[count]G`: go to line number

`%`: jump to matching parenthesis

`m{letter}`: set `{mark}` `{letter}` at cursor position

``{mark}`: jump to a `{mark}`

Navegando pelos saltos

<C-]>: "Jump to the definition of the keyword under the cursor."

<C-o>: go to older
cursor position in jump list.

<C-i>: go to ~~irineu~~ newer
cursor position in jump list.

Autocomplete

<C-n>: find **n**ext match

<C-p>: find **p**revious match

Recomendações

vim-adventures.com

```
$ vimtutor
```

```
:help
```

Referência



Drew Neil, *Practical vim: Edit text at the speed of thought*, Practical Vim (2015), 1–356.