

Vim, Vi e Venci

Editando códigos como um mestre!

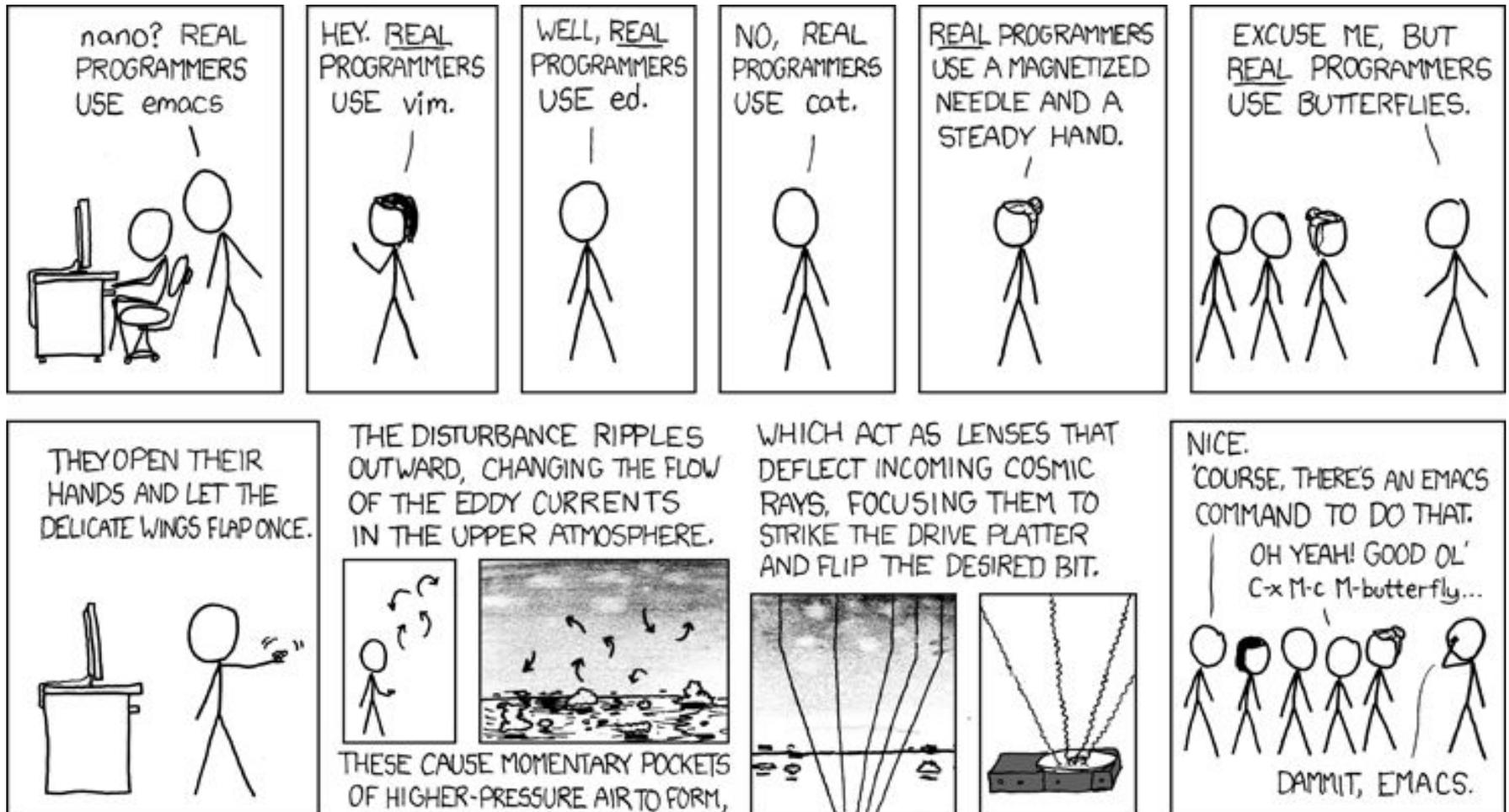
Acesse e siga as instruções em:

<http://lmtc.ufabc.edu.br/cursovim>

Prof. Fabrício Olivetti de França

Prof. Rodrigo Hausen

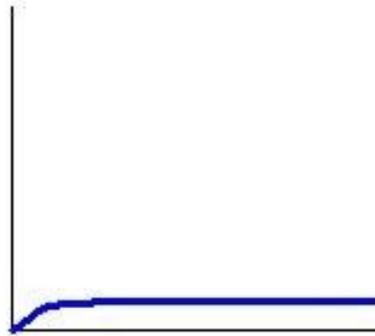
Por que aprender Vim?



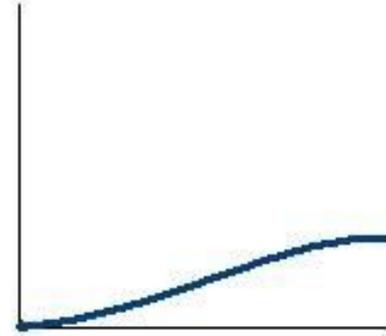
Curva de Aprendizado

Classical learning curves for some common editors

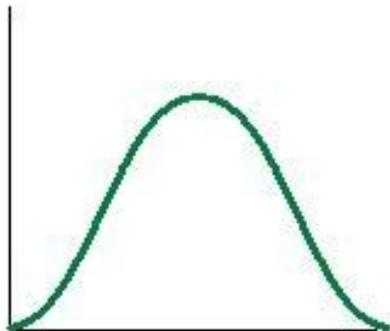
Notepad



Pico



Visual Studio



vi



emacs



Por que aprender Vim?

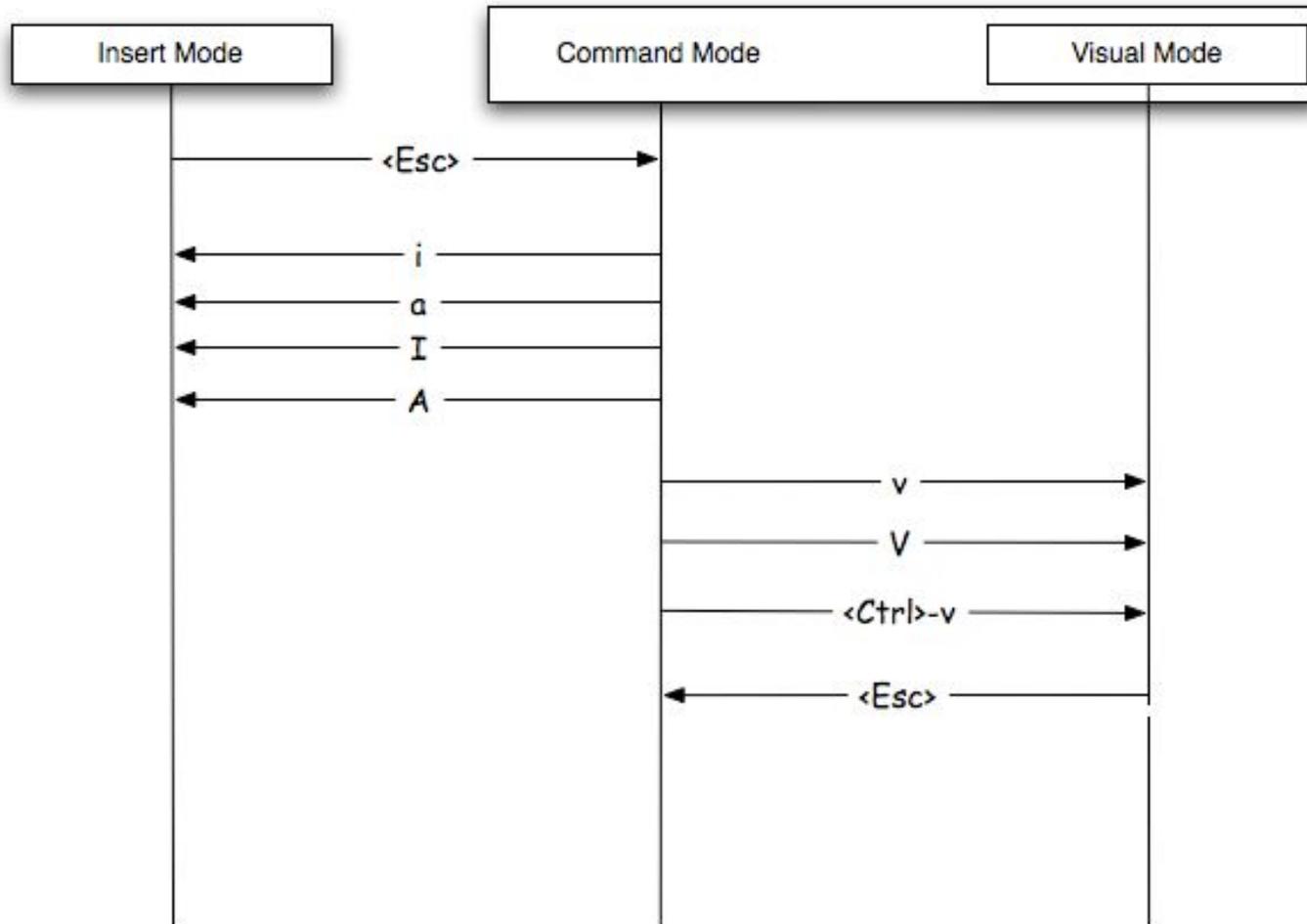
Principalmente pelo aumento na produtividade em:

- Escrever códigos-fonte
- Pré-processamento de textos e bases de dados

Por que aprender Vim?

- Não utiliza mouse
- Rápida criação de códigos
- Te fará um programador feliz :)

Os 3 modos de uso



Os 3 modos de uso

Modo de inserção: inserção textos (edição).

Modo de comando: execução de comandos.

Modo visual: seleção de textos.

Os 3 modos de uso

Por padrão o **Vim** inicia em modo de comando.

Modo Comando: Mover

Para mover o cursor pelo texto podemos usar as **setas** do teclado ou as teclas:



representando **esquerda, direita, cima, baixo** respectivamente.

Teste você mesmo!

Abra o arquivo “Textolnicial”:

\$ vim Textolnicial

Navegue pelo texto usando as setas e os comandos do slide anterior.

Modo Comando: Mover

Existem também os comandos para pular o cursor para certas posições:

- e** final da palavra (end)
- w** próxima palavra (word)
- b** início da palavra (beginning)

Modo Comando: Mover

Esse texto servirá como exemplo inicial dos comandos básicos do Vim.

e

Esse e texto servirá como exemplo inicial dos comandos básicos do Vim.

W

Esse texto servirá como exemplo inicial dos comandos básicos do Vim.

b

Esse texto servirá como exemplo inicial dos comandos básicos do Vim.

Modo Comando: Mover

Além disso, eles podem ser alterados da seguinte forma:

letra maiúscula (E, B, W) - ignora pontuação
g (ge, gb, gw, gE, gB, gW) - anda para trás

Modo Comando: Mover

Outros comandos:

0 (zero)	início da linha
\$	final da linha
^	primeiro símbolo imprimível da linha
%	(,), [,] ou {, } correspondente
{ }	começo e fim do parágrafo
()	começo e fim da sentença

Vamos testar

*Essa linha é um teste ...e mais... e mais
outro texto*

0

*_ Essa linha é um teste ...e mais... e mais
outro texto*



*Essa linha é um teste ...e mais... e mais
outro texto*

Modo Comando: Editar

Os comandos a seguir atuam no conteúdo da posição atual do cursor até a posição especificada pelo comando seguinte:

d - apaga o conteúdo e armazena no buffer

y - copia o conteúdo no buffer

c - altera o conteúdo (entrando em modo de inserção)

Modo Comando: Editar

r - substitui o caractere atual

x - deleta o caractere no cursor

X - deleta o caractere anterior

p - cola o conteúdo do buffer após a posição

P - cola o conteúdo do buffer antes da posição

. - repete último comando de edição

Modo Comando: Editar

dd - apaga a linha inteira

yy - copia a linha inteira

d3w - apaga as 3 próximas palavras

Vamos praticar

Esse texto servirá como exemplo inicial dos comandos básicos do Vim.

dw

texto servirá como exemplo inicial dos comandos básicos do Vim.

d2w

d como exemplo inicial dos comandos básicos do Vim.

**wcw amostra<ESC>wwwPId
ces<ESC>**

*c como amostra inicial dos exemplos básicos do
Vim.*

Modo Comando: Oops

u - desfaz última alteração

Ctrl - R - refaz última alteração desfeita

Vamos praticar

*System.out.println("vou concatenar a variavel
"+var1+" com a "+var2+" gerando a funcao
"+f+"("+var1+", "+var2+"");*

f+s + <ESC>;.;;.;;.;;.;;.;;.;;.

```
System.out.println("vou concatenar a variavel "  
+ var1 + " com a " + var2 + " gerando a funcao  
" + f + "(" + var1 + "," + var2 + " ")");
```

Vamos praticar

f+ próxima ocorrência do caractere “+”

;
repete última busca

Modo Inserção

i - começa a inserir na posição anterior ao cursor

a - insere na posição posterior

I - insere antes do primeiro símbolo da linha

A - insere após o último símbolo da linha

o - cria uma nova linha abaixo do cursor

O - cria uma nova linha acima do cursor

ESC - retorna ao modo de comandos

Modo Inserção

Ctrl-p - auto completa palavra

Ctrl-x Ctrl-l - auto completa linha

Modo Visual

Para entrar no modo visual:

v - modo de seleção por caractere

V - modo de seleção por linha

Ctrl - V - modo de seleção por bloco

Busca e Substituição

Um dos pontos fortes do Vim é seu sistema de localizar e substituir strings.

A busca é feita pelo comando “/” e a substituição pelo comando “:s”.

Busca

Vamos procurar pela palavra “texto”:

/texto

ao digitar o comando “/” seguido de <ENTER>
ele irá para a próxima ocorrência.

Busca com REGEX

O Vim aceita expressões regulares:

`/te.t.`

O “.” representa qualquer caractere, então ele procura por toda string composta por “te” + qualquer caractere + “t” + qualquer caractere.

Busca com REGEX

Outras expressões regulares:

^ - início da linha

\$ - fim da linha

\+ - 1 ou mais do caractere anterior

* - 0 ou mais do caractere anterior

? - 0 ou 1 ocorrência

[] - lista de caracteres aceitos

Vamos testar

Encontrar o nome das frutas que começam com a letra “m” e terminam com a letra “a”:

/m.\+a

:(

Vamos testar

Encontrar o nome das frutas que começam com a letra “m” e terminam com a letra “a”:

```
/^m.\+a
```

```
:(
```

Vamos testar

Encontrar o nome das frutas que começam com a letra “m” e terminam com a letra “a”:

```
/^m.\+a,
```

```
:/
```

Vamos testar

Encontrar o nome das frutas que começam com a letra “m” e terminam com a letra “a”:

```
^(m.\+a)$
```

:) `\< \>` delimita uma única palavra

Vamos testar

```
^<b(an\?)+>
```

Toda palavra (`< >`) que começa com “b” e repete “an” ou “a” 1 ou mais vezes.

Para saber mais

[http://learnvimscriptthehardway.stevelosh.com/
chapters/31.html](http://learnvimscriptthehardway.stevelosh.com/chapters/31.html)

Substituição

`:%s/busca/substitui/gci`

s - comando de substituição

% - no arquivo inteiro (sem isso ele faz apenas na linha atual)

g - todas as ocorrências (sem ele, substitui apenas a primeira ocorrência)

i - caso insensitivo

Vamos testar

```
:%s/\/bacon/g
```

substitui toda ocorrência de palavras que começam com “m” e terminam com “a”, por bacon.

Salvando e saindo



I Am Developer

@iamdeveloper



Following

I've been using Vim for about 2 years now, mostly because I can't figure out how to exit it.

Salvando e saindo

No modo de comando:

:w grava arquivo

:w nome grava arquivo com outro nome

:q sai do Vim

:q! sai do Vim e ignora modificações

:wq salva o arquivo e sai do Vim

Editando códigos

Para os próximos comandos vamos abrir o código **confuso.c**

```
$ vim confuso.c
```

Comandos úteis

:set number mostra o número das linhas

:syntax on sintaxe colorida

Vamos praticar!

- 1) Vamos renomear a variável “n” para “numero” e “i” para “passo”

Vamos praticar!

- 1) Vamos renomear a variável “n” para “numero” e “i” para “passo”

```
:%s/\<n\>/numero/gc
```

```
:%s/\<i\>/passo/gc
```

Vamos praticar!

2) Vamos remover a instrução (1<<numero) de dentro do comando “for” e atribuir uma variável para seu valor:

- a) posicione o cursor no “(“
- b) digite: **c%repeticoes <ESC>**
- c) na linha anterior digite **p**
- d) no começo da linha entre em modo de inserção e digite “**repe**” e aperte **ctrl-p**
- e) saia do modo de edição e aperte **>>**

Vamos praticar!

3) A função `ffs(passo)` é repetida diversas vezes, vamos armazenar seu retorno em uma variável e substituir a ocorrência pelo nome dessa variável.

Vamos praticar!

a) `/ffs<ENTER>`

b) `y%kpbbbi<ENTER>`

c) `digito = <ESC>`

d) `j:s/ffs([[[:alpha:]]\+)/digito/g`

Vamos praticar!

4) Mover a instrução: `abs(...)%3 +1` para uma variável de nome “destino”

/abs

v%→%%←cdestino<ESC>

ko<ESC>p0idestino = <ESC>

>>.

Vamos praticar!

5) Vamos repetir para retirar a parte abs() para a variável “movimento”

/abs

c%movimento

ko<ESC>pimovimento = <ESC>>>.

Vamos praticar!

5) $(\text{passo}/(1 \ll \text{digito}) + 1) \% 3$ para “inverso”

$\% \%$ →

$v\%$ → → **cinverso**

ko<ESC>p0iinverso = <ESC>>>.

Vamos praticar!

5) (numero%2 == digito%2) para “par”

%%→%%

c%par<ESC>

ko<ESC>p0ipar = <ESC>>>.

Vamos praticar!

6) Vamos criar duas funções:

“procedimento” que recebe a variável numero como parâmetro e faz todo o laço “for”

e

“qualdestino” que recebe como parâmetro numero, passo, digito e retorna destino.

Vamos praticar!

Crie as seguintes linhas na linha 4:

```
int procedimento(int numero);
```

```
int qualdestino(int numero, int passo, int digito);
```

Vamos praticar!

Na primeira linha faça:

A<BKSPC>{<ENTER>}<ENTER><ESC>

Em seguida faça:

j.

Vamos praticar!

Copie os códigos correspondentes utilizando:

SHIFT+V

d

p

Vamos praticar!

Na primeira linha da função `qualdestino()`
insira:

```
int inverso;
```

```
int par;
```

```
int movimento;
```

digite: **kk3J**

Vamos praticar!

Na mesma linha:

:s;/,/gc

:s/int//gc

Vamos praticar!

Finalmente:

```
:%s/([^;{}>]\)$\1;/gc
```

```
gg=G
```

Vamos praticar!

Salve com **:w NovoCodigo.c**

E saia com **:q!**



Give me a
computer fast
enough to run
GCC and **VIM** on
which to code,
and i shall code
the world

Archimedes

Outros comandos interessantes

Janelas:

`:split <nome arquivo>`

`:vsplit <nome arquivo>`

`<CTRL>-W <CTRL>-W`

Outros comandos interessantes

* - busca para frente por palavra sob o cursor

- busca para trás

:Ex - abrir arquivo com interface

guu, gUU - linha em caixa baixa/alta

~ - inverte caixa do caractere

ggg?G - rot13

xp - swap de caracteres

Outros comandos interessantes

<CTRL-A>, <CTRL-X> -
incrementa/decrementa número no cursor

<CTRL-R>=5*5 - insere 25 no texto (insert mode)

G=gg - reidentifica documento

Outros comandos interessantes

“<SPACE>ayy - copia conteúdo no registro de nome “a”

“<SPACE>Ayy - append do conteúdo no registro de nome “a”

“<SPACE>ap - cola conteúdo do registro de nome “a”

Para avançar

vimtutor

<http://www.vimgolf.com/>

http://vim.wikia.com/wiki/Best_Vim_Tips

Practical Vim

<https://www.cs.oberlin.edu/~kuperman/help/vim/home.html> (windows.html é muito útil!)