

#Aula 01

Tópicos

- Por que aprender vim/vi
- Um pouco de história
- Editores de linha e visuais
- Criando um script com o 'ed'
- Editando o mesmo script com o 'ex'
- Entendendo o conceito de 'modos'
- As metas da primeira semana
- Comandos e movimentos essenciais

Por que aprender vim/vi

- O 'vi' é o editor padrão UNIX, *BSD (nvi) e GNU/Linux.
- Perfeita integração com a plataforma (shell).
- Dominar todo um ecossistema, em vez de um apenas editor.
- Incríveis poderes de edição ("pilhas incluídas").
- Induz bons hábitos de edição naturalmente.
- Altamente customizável (Vim Script + Bash).

Um pouco de história...

Ano	Quem/O que...	Editor
1971	Ken Thompson	'ed'
1976	Bill Joy	'ex'
1978	Link para modo visual do 'ex'	'vi'
1987	Tim Thompson	'steevie'
1990	Steve Kirkendall	'elvis'
1991	Bram Moolenaar	'vim'
1994	Keith Bostic	'nvi'

Editores de linha

ed

- Direcionado a teleimpressoras (tty).
- Modos de comando e de edição.
- Primeira implementação de REGEX.
- Origem do utilitário 'grep'.
- Lê a saída de comandos do shell (!)*.
- Salva com 'w' e sai com 'q'.

(*) Pelo menos na versão GNU.

ex

- Feito para terminais em vídeo.
- Introduz o ':' para entrada de comandos.
- Novas sintaxes para substituições com REGEX ('%s').
- Mapeamento de teclas de atalho (':map').
- Abreviações para autotexto (':ab').
- Marcadores de linhas (k / ').
- Modo de edição visual (':vi').
- Navega entre linhas com 'j' e 'k' no modo visual.

Editores visuais

vi

- Começou como um modo do 'ex'.
- Na versão 2.0 do 'ex', era um link simbólico.
- Recebeu atenção dedicada a partir de 1979.
- Teclas de movimentação, comandos e funções.
- Clonado por diversos projetos (elvis, stevie, etc).
- O clone 'stevie' deu origem ao Vim ('VI iMitation').
- O clone 'elvis' deu origem ao 'nvi', padrão nos *BSD.
- Até hoje, é o editor padrão da família UNIX.
- Mantém os comandos e o modo de edição 'ex'.

vim

- Desenvolvido a partir do 'stevie' (clone do 'vi').
- Inicialmente, significava 'VI iMitation'.
- Passou a ser 'VI iMproved' em 1993.

- Mantém um modo de compatibilidade com o 'vi'.
- Mantém um modo de edição de linha ('ex').
- Implementa um modo gráfico ('vim-gtk3'/'gvim').
- Implementa a própria linguagem de script.
- Ativamente desenvolvido até hoje.
- Disponível na maioria das distribuições GNU/Linux*.

(*) Instalação opcional.

Criando um script com editores de linha

Script `salve.sh`

```
#!/bin/bash
```

```
read -erp 'Digite o seu nome: '
```

```
if [[ $REPLY ]]; then
```

```
    echo "Salve, $REPLY!"
```

```
else
```

```
    echo 'Que sem graça... :-(
```

```
fi
```

Entendendo o conceito de 'modos'

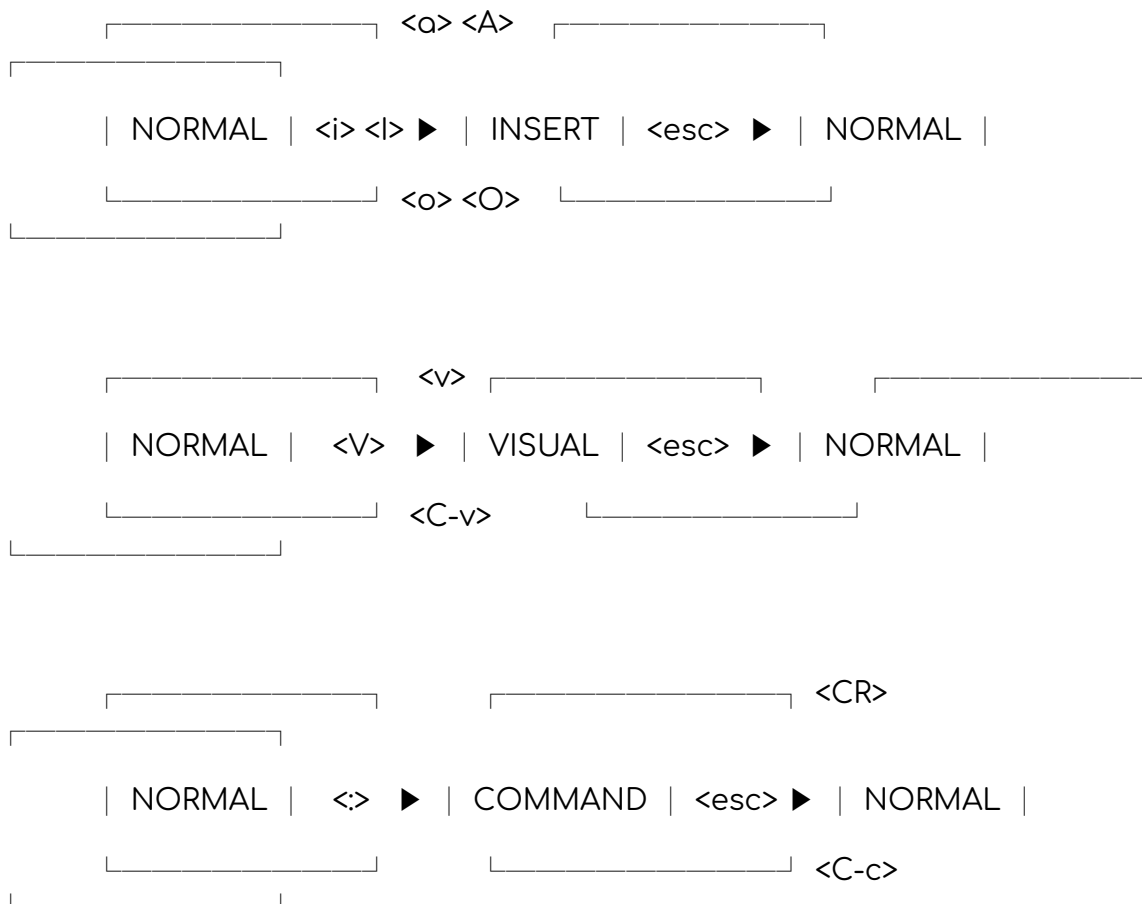
Por que o vim é modal?

- Foi uma solução brilhante para teleimpressoras.

Por que o vim continua sendo modal?

- Escrever e editar são tarefas diferentes.
- Nós passamos mais tempo editando do que escrevendo.
- Solução brilhante para desenvolver bons hábitos!

Os quatro principais modos...



As metas da primeira semana

- Utilizar o vi diariamente (não o vim!).
- Memorizar os principais comandos e movimentos.

Sobre as teclas de direção:

- Você pode usar as setas, se quiser.
- Mas memorizá-las é fundamental!
- Elas fazem parte de muitos comandos!

Dicas:

- Foque em se habituar com o vi, não na digitação.
- Nesta semana, apenas anote os "como fazer tal coisa".
- Abrace o desconforto!

Comandos e movimentos essenciais

Movimentação do cursor

k	<Up>
↑	↑
h ← → l	<Left> ← → <Right>
↓	↓
j	<Down>

- Início da linha: **0** ou **<Home>**
- Fim da linha: **\$** ou **<End>**
- Página anterior (back): **<C-b>** ou **<PgUp>**
- Próxima página (forward): **<C-f>** ou **<PgDown>**
- Início do buffer: **gg** ou **<C-Home>**
- Fim do buffer: **G** ou **<C-End>**

Seleção de texto

- Seleção de caracteres: **<v><direções>**
- Seleção de linhas: **<V><direções>** (para cima ou para baixo)
- Seleção em bloco: **<C-v><direções>**

Recortar (delete) e copiar (yank)

- Copiar seleção: **y**
- Recortar seleção: **d**
- Recortar a linha corrente: **dd**
- Copiar a linha corrente: **yy**
- Recortar várias linhas abaixo: **<#linhas>dd** ou **d<#linhas>j**
- Copiar várias linhas abaixo: **<#linhas>yy** ou **y<#linhas>j**
- Recortar várias linhas acima: **d<#linhas>k**
- Copiar várias linhas acima: **y<#linhas>k**
- Colar (put) depois do cursor: **p**
- Colar (put) antes do cursor: **P**

Entrar no modo INSERT

- Iniciar inserção depois do cursor: **a**
- Iniciar inserção antes do cursor: **i**
- Iniciar inserção na linha abaixo: **o**
- Iniciar inserção no fim da linha: **A**
- Iniciar inserção no início da linha: **I**

- Iniciar inserção na linha acima: **O**
- Voltar ao modo NORMAL para executar um comando: **<C-o>**
- Voltar ao modo NORMAL: **<esc>**

Desfazer/refazer

- Desfazer: **U**
- Refazer: **<C-r>**

Abrir e fechar arquivos

- Abrir arquivo pela linha de comandos:

:~\$ vi <arquivo>

Podemos digitar
vim README.md

- Abaixo será mostrado o nome do arquivo aberto.
- Número de Linhas Ex.: 104L
- Número de caracteres Ex.: 5578C
- Linha e coluna atual é mostrada assim: 1,1
- É mostrado no VIM em qual região do arquivo está localizado.

- Abrir ou criar arquivo:

:e <arquivo>

- Fechar arquivo corrente sem salvar:

:bd!

- Fechar arquivo corrente:

:bd

Para criar um novo arquivo através do VIM, digitamos o comando vim seguido do nome .tipo do arquivo.

Ex.: **vim exemplo.txt**

O caractere ~ no início de cada linha no vim está indicando que aquela é uma linha vazia.

- Ao abrir o VIM inicialmente ele é iniciado no modo de comandos, onde os caracteres, teclas são interpretados como comandos do VIM.
- Para digitar um texto no arquivo aberto no VIM é preciso entrar no modo de INSERÇÃO.
- Para entrar no modo de INSERÇÃO digitamos a letra i, logo após o VIM mostrará a palavra INSERT indicando que podemos inserir o texto.
- Para Sair do modo de INSERÇÃO apertamos ESC.
- Para Salvar o que for digitado no arquivo criado utilizamos o seguinte atalho: **ESC + SHIFT + : + w** para poder salvar as modificações e logo em seguida utilizamos o atalho para sair do vim

Salvar e sair

- Salvar:

`:w [nome]<cr>`

- Salvar como:

`:sav <novo_nome><cr>`

- Salvar e sair:

`:x [nome]<cr> ou :wq [nome]<cr>`

- Sair sem salvar:

`:q!`

- Sair:

`:q`

Atalho para sair do VIM, Apertar **ESC+SHIFT+ : + q**

- Sair do arquivo sem salvar as modificações do arquivo **ESC + SHIFT + : + q!**

Os quatro principais modos

NORMAL	INSERT	NORMAL
NORMAL	VISUAL	NORMAL
NORMAL	COMMAND	NORMAL

<esc> Tecla utilizada para voltar ao modo normal

: Teclando dois pontos saímos do modo normal e entramos no modo COMMAND

vim -u NONE o vim entra no modo compatível com o vi

No modo NORMAL

0 Vai para o início da linha.

\$ Vai para o final da linha

<Ctrl + b> Vai para a página anterior

<Ctrl + f> Vai para a próxima página

gg Vai para o início do buffer

G Vai para o fim do buffer

^ O circunflexo vai para o primeiro caractere legível da linha

Seleção de Texto

v Seleção de caracteres

V Seleção de linhas (Para cima ou para baixo)

V2j Seleciona a linha atual mais duas linhas

Recortar (delete) e copiar (yank)

- Seleções

Copiar: tecla y

Recortar: tecla d

- Linhas

Copiar a linha corrente: tecla yy

Recortar a linha corrente: tecla dd

Recortar várias linhas abaixo

<#linhas>dd Nesta opção estamos repetindo o comando - Abaixo

d<#linhas>Deletar quantidade(N) de linhas abaixo - Na faixa informada

Copiar várias linhas abaixo

<#linhas>yy Nesta opção estamos repetindo o comando - Abaixo

y<#linhas>Copiar quantidade(N) de linhas abaixo - Na faixa informada

Recortar várias linhas acima
d<#linhas>k

Copiar várias linhas acima
y<#linhas>k

Desfazer / Refazer MODO NORMAL
Desfazer: U
Refazer: <Ctrl-r>

Abrir e fechar arquivos
vim <arquivo> Abrir arquivo pela linha de comandos
:e Abrir ou criar arquivo
:bd! Fechar arquivo corrente sem salvar
:bd Fechar arquivo corrente

Salvar e sair
:w [nome]<enter> Salvar
:sav <novo_nome> <enter> Salvar como
:x [nome] ou :wq [nome]<enter> Salvar e sair
:q! Sair sem salvar
:q Sair

Repositório
https://codeberg.org/blau_oraujo/vim-keys

- Selecione o texto estando no modo visual e entre no modo Insert para substituir o que foi selecionado e apagado.
:v e Tecla direcional para selecionar o texto e depois tecle c

Realizar substituição de uma determinada ocorrência de texto.

- Selecione o texto Shift + v
- Digite s para substituir, informe a ocorrência de texto, informe a o texto que irá substituir a ocorrência anterior, substitui todas as vezes que o texto aparece.

: '<, '>s/vi/VI/g

#AULA 02

Revisão da aula 01

Modos

- Normal <esc>
- Insert a A i l o O
- Visual v V <C-v>
- Command :

Movimento

- Caractere h j k l (+ setas)
- Linha 0 <Home> \$ <End>
- Página <C-b> <PgUp> <C-f> <PgDn>
- Buffer gg <C-Home> G <C-End>

Recortar e Copiar

- Seleções d y
- Linhas dd yy
- Colar p P

Máquina do tempo

- Desfazer U
- Refazer <C-r>

Salvar e Sair

- Salvar :w [nome]<cr>
- Salvar como :sav <nome><cr>
- Salvar e sair :x [nome]<cr> (ou :wq)
- Sair sem salvar :q!
- Sair :q

Abrir e fechar arquivos

- Abrir / criar arquivo :e <arquivo>
- Fechar arquivo :bd
- Fechar sem salvar :bd!

- Listar todos os buffers abertos no vim
:ls

Colocar processo em segundo plano utiliza-se <Ctrl-z>

Retomar o processo em segundo plano digita-se fg

A dica acima pode servir para momentos em que abrimos um arquivo ou mais no vim e queremos parar o processo em execução para executarmos outro comando no terminal.

Existem visualizações de buffers no vim, são janelas e abas

Para fechar qualquer janela no vim utiliza-se o comando
:q

Opções da linha de comando

- Abertura de arquivos
vim arquivo1 arquivo2
- Abertura de arquivos em janelas
vim -o arquivo1 arquivo2 ... #horizontal
vim -O arquivo1 arquivo2 ... #vertical

Essa opção horizontal ou vertical diz respeito a linha divisória das janelas ou ela é vertical ou horizontal

Para fechar todas as janelas de uma vez utilize
:qa

Abrir 4 arquivos em quatro janelas horizontais
vim -o teste.txt teste2.txt /etc/shells /etc/os-release

Abrir 4 arquivos em quatro janelas verticais
vim -O teste.txt teste2.txt /etc/shells /etc/os-release

Abrir 4 arquivos em duas janelas verticais
vim -O2 teste.txt teste2.txt /etc/shells /etc/os-release

Abrir 4 arquivos em duas janelas horizontais
vim -o2 teste.txt teste2.txt /etc/shells /etc/os-release

Abrir arquivos 4 arquivos em abas
vim -p java.txt python.txt /etc/shells /etc/os-release

Abrir 2 abas
vim -p2 java.txt python.txt

A movimentação por essas abas ocorre da seguinte forma:

<Ctrl-Page Up> Próximo
<Ctrl-Page Down> Anterior

ou <gt> Próximo
ou <gT> Próximo

<Ctrl-w-s> Abre outra janela na mesma aba

Comando para editar o buffer seguinte no lugar do buffer atual

`:bn`

Ou utilizar o comando `:ls` para ver o nome do arquivo e abrir o arquivo utilizando `:b` e número do arquivo mostrador por `:ls`

Exemplo:

`:b4` Abre o buffer quatro para ser editado

O Comando `:only` vai tornar a janela com o buffer corrente o único buffer visível.

No Vim a Janela é apenas um frame ou quadro para a visualização de buffers

- Abertura de arquivos em abas
`vim -p arquivo1 arquivo2`

- Abertura de diretórios
`vim caminho/`

Abrir todos os arquivos no diretório
`vim *`

Entrar no diretório corrente
`vim .`

v - Abre o arquivo do diretório na Vertical
h - Abre o arquivo do diretório na horizontal
p - Mostra um preview do arquivo

Tecle `:e ./`
E o netrw será aberto

Digite `:30Lex`

Para que de um lado tenhamos um buffer vazio e do outro o netrw
Para fechar execute novamente `:30Lex`

O número refere-se a largura da coluna lateral esquerda que vai ser aberta
Poderia ser `:20Lex`

Buffers, janelas e abas

Buffer

- O conteúdo de um arquivo na memória.
- Por padrão, buffers precisam ser salvos antes de serem ocultados!
- Isso pode ser mudado com:

`:set hidden`

Antes é preciso configurá-lo dentro do `.vimrc`

Dica para o `'vimrc'`:

Inserir a linha

`set hidden`

Criar o `./vimrc` no diretório no meu diretório principal

`vim ~/.vimrc`

`Ctrl + 6` Alternar entre os dois últimos buffers que estamos trabalhando

Janelas e abas

- Formas de exibição de buffers.
- Fecha-las não afeta os buffers exibidos.
- O comando `:q` fecha janelas e abas!
- O comando `:qa` fecha todas as janelas e abas!

Comandos básicos para operações com buffers

Abrir, criar e recarregar

- Abrir arquivo em um novo buffer:

`:e[dit] [arquivo]`

`:fin[d] <arquivo>`

- Novo buffer na janela corrente:

`:enew`

- Nova janela horizontal com um buffer vazio:

`:new`

- Nova janela vertical com um buffer vazio:

`:vnew`

- Nova aba com um buffer vazio:

:tabnew

Navegação entre buffers

- Listar buffers:

:ls

- Circulação:

:bn " Próximo buffer

:bp " Buffer anterior

- Navegação:

:b N " Ir para buffer de número N

:b NOME " Ir para buffer de nome NOME

- Alternar entre dois buffers:

<C-6>

Salvar e fechar buffers

- Salvar buffers:

:w[rite] [arquivo] " salvar buffer corrente

:wa[ll] " salvar todos os buffers

:sav[eas] <arquivo> " salvar como...

- Descarregar buffer (fechar):

:bd[elete] " Deletar o buffer corrente

:%bd[elete] " Deletar todos os buffers

:[%]bd! " Deletar buffers sem salvar

Os buffers descarregados não são excluídos da memória: eles são apenas removidos da lista de buffers.

Salvar buffer como 'root'

:w !sudo tee %

- Como funciona:

! " Executa um comando do shell.

:w! " Faz um pipe do conteúdo do buffer
" para um comando do shell.

sudo " Executa comandos como outro usuário.

tee " Escreve dados recebidos por pipe
" em stdout e em um arquivo.

% " Nome do arquivo.

Dica para o '.vimrc':

```
cmap w!! w !sudo tee > /dev/null %
```

Comandos básicos para janelas e abas

Abrir arquivos em janelas

:sp[lit] [arquivo] " Nova janela horizontal.

:vsp[lit] [arquivo] " Nova janela vertical.

Abrir arquivos em abas

:tabe[dit] [arquivo] " Nova aba.

:tabs ball " Todos os buffers em abas.

Fechar todas as janelas (a visualização)

:on[ly]

Fechar todas as abas (a visualização)

:tabo[nly]

Dica para o '.vimrc':

```
:set splitbelow
```

```
:set splitright
```

Movimentação entre janelas

<C-w> <seta direção> " Ir para janela na direção.
<C-w> w " Circular entre as janelas.

Movimentação entre abas

gt ou <C-PgDown> " Próxima aba.
gT ou <C-PgUp> " Aba anterior.

Outras dicas de comandos com janelas e abas

Abrir arquivo no texto sob o cursor

gf " Abrir buffer para editar o arquivo.
<C-w> f " Abrir o arquivo em nova janela (hor).
<C-w> v gf " Abrir o arquivo em nova janela (ver).
<C-w> gf " Abrir o arquivo em nova aba.

Sincronização de rolagem

É possível sincronizar a rolagem de duas janelas verticais contendo textos com as mesmas quantidades de linhas:

:set scrollbind " Executar em ambas as janelas!