

vim

linux

/vía: <https://www.catswhocode.com/blog/vim-cheat-sheet-for-2016>

/vía: <http://www.catswhocode.com/blog/130-essential-vim-commands>

info

- [.vimrc](#)
- [1 al día por mlechon](#)

Plugins/extensions/personalización

Pendiente: modificación status line

<https://kadekillary.work/post/statusline/>

Pendiente: modificación status line

<https://gist.github.com/millermedeiros/1262085>

- gestión/instalación plugins en sus propios directorios (runtimepath):
<https://github.com/tpope/vim-pathogen>
 - crear dos directorios: `~/ .vim/autoload` (ubicación `vim-pathogen.vim`) y `~/ .vim/bundle` (home_directory de los plugins)
 - modificar `.vimrc` añadiendo `execute pathogen#infect()`
- `statline.vim` (requiere `pathogen.vim`):

```
cd ~/ .vim/bundle
git clone https://github.com/millermedeiros/vim-statline.git
```

config

```
# cambiar editor por defecto
$ sudo update-alternatives --config editor
```

Basics

- `:e filename` : Open filename for edition
- `:w` : Save file
- `:q` : Exit Vim
- `:q!` : Quit without saving
- `:x` : Write file (if changes has been made) and exit

- `: sav filename` : Saves file as filename
- `.` : Repeats the last change made in normal mode
- `5.` : Repeats 5 times the last change made in normal mode

Moving in the file

- `0` : move the cursor to the beginning of the line
- `G o $` : move the cursor to the end of the file
- `gg` : move the cursor to the beginning of the file
- `: 59` : move cursor to line 59. Replace 59 by the desired line number.
- `k` or Up Arrow : move the cursor up one line
- `j` or Down Arrow : move the cursor down one line
- `e` : move the cursor to the end of the word
- `b` : move the cursor to the beginning of the word
- `L` : move the cursor to the bottom of the screen
- `20|` : move cursor to column 20.
- `%` : Move cursor to matching parenthesis
- `[[` : Jump to function start
- `[{` : Jump to block start

Cut, copy & paste

- `y` : Copy the selected text to clipboard
- `p` : Paste clipboard contents
- `dd` : Cut current line
- `yy` : Copy current line
- `y$` : Copy to end of line
- `D` : Cut to end of line

Search

- `/word` : Search word from top to bottom
- `?word` : Search word from bottom to top
- `*` : Search the word under cursor
- `/\cstring` : Search STRING or string, case insensitive
- `/jo[ha]n` : Search john or joan
- `/\< the` Search the, theatre or then
- `/the\>` : Search the or breathe
- `/\< the\>` : Search the
- `/\< |.\>` : Search all words of 4 letters
- `/\//` : Search fred but not alfred or frederick
- `/fred\|joe` : Search fred or joe
- `/\<\d\d\d\d\>` : Search exactly 4 digits
- `/^\n\{3}` : Find 3 empty lines
- `:bufdo /searchstr/` : Search in all open files
- `bufdo %s/something/somethingelse/g` : Search something in all the open buffers and replace it with somethingelse

Replace

- comando:

```
: [rango]s[substitute]/{pattern}/{string}/[flags] [count]
```

- Rango:

- % : de la primera a la última línea
- 1,3 : de la línea 1 a 3
- 3,\$: de la línea 3 al final de fichero

- Flags:

- g : global (en el ámbito, todas las sustituciones posibles)
- i : case insensitive
- c : confirmación

- Pattern:

- string : busca la cadena
- ^ : principio de línea
- \$: final de línea

- Ejemplos:

- :s/Bill/Steve/ : Replace the first occurrence of Bill by Steve in current line
- :s/Bill/Steve/g : Replace Bill by Steve in current line
- :2,35s/old/new/g : Replace all occurrences between lines 2 and 35
- :5,\$s/old/new/g : Replace all occurrences from line 5 to EOF
- :%s/old/new/g : Replace all occurrences of old by new in file
- :%s/onward/forward/gi : Replace onward by forward, case insensitive
- :%s/old/new/gc : Replace all occurrences with confirmation
- :%s/^/hello/g : Replace the beginning of each line by hello
- :%s\$/Harry/g : Replace the end of each line by Harry
- :%s/onward/forward/gi : Replace onward by forward, case insensitive
- :%s/ *\$/ /g : Delete all white spaces
- :%s/Bill/Steve/g : Replace Bill by Steve in all the file
- :%s/^M/ /g : Delete DOS carriage returns (^M)
- :%s#\s#\r#g : cambia espacios (\s) por saltos de línea (\r) en todo el texto. Usamos # como separador
- :%s/\r/\r/g : Transform DOS carriage returns in returns
- :%s#<[^>]\+>##g : Delete HTML tags but keeps text
- :%s/^(.*)\n\1\$/\1/ : Delete lines which appears twice
- :g/string/d : Delete all lines containing string
 - g/^\$/d : borra líneas en blanco → v/. /d
 - g/^\s*\$/d : borra líneas en blanco o con espacios/tabuladores
 - más ejemplos: https://vim.fandom.com/wiki/Power_of_g
- :v/string/d : Delete all lines containing which didn't contain string

- Ctrl+a : Increment number under the cursor
- Ctrl+x : Decrement number under cursor
- ggVGg? : Change text to Rot13

Case

- Vu : Lowercase line
- VU : Uppercase line
- g~~ : Invert case
- vEU : Switch word to uppercase
- vE~ : Modify word case

- `ggguG` : Set all text to lowercase
- `gggUG` : Set all text to uppercase
- `:%s/\<.\>/\u&/g` : Sets first letter of each word to uppercase
- `:%s/\<.\>/\l&/g` : Sets first letter of each word to lowercase
- `:%s/.*\>/\u&` : Sets first letter of each line to uppercase
- `:%s/.*\>/\l&` : Sets first letter of each line to lowercase

Read/Write files

- `:1,10 w outfile` : Saves lines 1 to 10 in outfile
- `:1,10 w >> outfile` : Appends lines 1 to 10 to outfile
- `:r infile` : Insert the content of infile
- `:23r infile` : Insert the content of infile under line 23

File explorer

- `:e .` : Open integrated file explorer
- `:Sex` : Split window and open integrated file explorer
- `:Sex!` : Same as Sex but split window vertically
- `:browse e` : Graphical file explorer
- `:ls` : List buffers
- `:cd ..` : Move to parent directory
- `:args` : List files
- `:args *.php` : Open file list
- `:grep expression *.php` : Returns a list of .php files contening expression
- `gf` : Open file name under cursor

Interact with Unix

- `:!pwd` : Execute the pwd unix command, then returns to Vi
- `!!pwd` : Execute the pwd unix command and insert output in file
- `:sh` : Temporary returns to Unix
- `$exit` : Returns to Vi

Alignment

- `:%!fmt` : Align all lines
- `!}fmt` : Align all lines at the current position
- `5!!fmt` : Align the next 5 lines

Tabs/Windows

cli

- `vim -p <archivo1> <archivo2>`

manejo

- `:tabnew` : Creates a new tab
- `:tabe[dit] <file>` : abre <file> en nuevo tab
- `:tabf[ind] <file>`: abre <file> en nuevo tab buscando por el path
 - `:set path=.,,*` : busca en la ubicación del archivo, en el directorio actual y en los subdirectorios
- `:tabclose`
- `:tabonly` : cierra todos los tabs menos el actual
- `:tab split`: duplica el tab actual

- `:tabdo %s/foo/bar/g` : Execute a command in all tabs
- `:tab ball` : Puts all open files in tabs
- `:new abc.txt` : Edit abc.txt in new window
- `:qa` o `wqa` : salir de todas las pestañas

movimiento

- `gt` : Show next tab
 - `:tabn`
 - `^PgUp`
- `gT` : Show previous tab
 - `:tabp` : Show previous tab
 - `^PgDw`
- `:tabs` : muestra todos los tabs
- `:tabfirst` : Show first tab
- `:tablast` : Show last tab

- `:tabm n` : reorganizar tabs
 - `0` : mover actual a la primera posición
 - `:` : mover actual a la última posición
 - `n` : mover actual a la posición **n**

/via: http://vim.wikia.com/wiki/Using_tab_pages

Window splitting

- `:e filename` : Edit filename in current window
- `:split filename` : Split the window and open filename
- `ctrl-w up arrow` : Puts cursor in top window
- `ctrl-w ctrl-w` : Puts cursor in next window
- `ctrl-w_` : Maximize current window vertically
- `ctrl-w|` : Maximize current window horizontally
- `ctrl-w=` : Gives the same size to all windows
- `10 ctrl-w+` : Add 10 lines to current window
- `:vsplit file` : Split window vertically
- `:sview file` : Same as `:split` in readonly mode
- `:hide` : Close current window
- `:nly` : Close all windows, excepted current
- `:b 2` : Open #2 in this window

Auto-completion

- `Ctrl+n Ctrl+p` (in insert mode) : Complete word
- `Ctrl+x Ctrl+l` : Complete line
- `:set dictionary=dict` : Define dict as a dictionary
- `Ctrl+x Ctrl+k` : Complete with dictionary

Marks

- `m {a-z}` : Marks current position as {a-z}
- `' {a-z}` : Move to position {a-z}
- `' '` : Move to previous position

Abbreviations

- `:ab mail mail@provider.org` : Define mail as abbreviation of mail@provider.org

Text indent

- `ctrl-t, ctrl-d` : Indent/un-indent in insert mode
- `>>` : Indent
- `<<` : Un-indent
- `=%` : Indent the code between parenthesis
- `!GVG=` : Indent the whole file

cifrado

- `:X` : pregunta por contraseña para cifrar el fichero a la hora de guardar
- `:setlocal cm?` : show encryption method for the current file (cm abreviación de cryptmethod)
- `:setlocal cm=zip` : weak (default for backwards-compatibility)
- `:setlocal cm=blowfish` : better
- `:setlocal cm=blowfish2` : best (requires Vim version 7.4.399 or higher)

Modelines

[pendiente, wiki](#)

mirar e integrar

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

permite modificar el comportamiento del editor (emacs o vim - u otros?) con líneas con estructura

activar

por defecto (y razones de seguridad) viene desactivado. Se han de cumplir estas condiciones:

- set modeline está activo (y no set nomodeline)
- modelines \neq 0
- no eres root

:verbose set modeline? modelines?: mirar si está activo

```
:e $MYVIMRC
set modeline
set modeline=5
```

SETs

búsquedas

- set (no)ignorecase [ic]
- set (no)hlsearch [hls] : marca la búsqueda
- set (no)incsearch [is]
- set (no)showmatch [sm]
- set (no)ignorecase [ic]
- set (no)incsearch [is]

indentación

- :set (no)autoindent [ai] : Turn on auto-indent
- :set smartindent [si] : Turn on intelligent auto-indent
- :set shiftwidth=4 : Defines 4 spaces as indent size

Syntax highlighting

- :syntax on : Turn on syntax highlighting
- :syntax off : Turn off syntax highlighting
- :set syntax=perl : Force syntax highlighting

ayudas

- set (no)cursorline [cul]: línea del cursor «subrayada»
 - modificar estilos (ojo, también incluye para columna):

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

```
" \c para activar/desactivar
:hi CursorLine   cterm=NONE ctermbg=darkred ctermfg=white
guibg=darkred guifg=white
:hi CursorColumn cterm=NONE ctermbg=darkred ctermfg=white
guibg=darkred guifg=white
:noremap <Leader>c :set cursorline! cursorcolumn!<CR>
```

- set (no)number [nu]: numeración de líneas

- `set (no) ruler [ru]`: muestra información de posición del cursor y % de la página
- `set (no) invlist` : muestra caracteres especiales
- `set (no) paste` : evita autoindentación de código «pegado»
 - `set pastetoggle=<F2>`
 - `<code vim .vimrc>noremap <F2> :set invpaste paste?<CR>`
- `set (no) list` : visualiza caracteres «ocultos» según establecido
 - `: set listchars=space:·,eol:$,tab:>- ,trail:~,extends:,>,precedes:< [lcs]`
 - `: help listchars [lcs]`

```
set pastetoggle=<F2> set showmode</code>
```

otros

- `:set smartcase` Ignore case in searches excepted if an uppercase letter is used
- `:set key=<password>` : otra manera de establecer contraseña (se ve la pass, se puede poner en fichero .vimrc)

From:

<https://miguelangel.torresegea.es/wiki/> - miguel angel torres egea

Permanent link:

<https://miguelangel.torresegea.es/wiki/linux:vim:comandos?rev=1575536951>

Last update: **05/12/2019 01:09**

