

Charla de VIM (VI Improved)

e-GHOST -ESIDE's GNU Hi-tech
Open Source Team
< www.e-ghost.deusto.es >

STenyaK -Bruno González
< stenyak@gmail.com >



INDICE

- 1 - Introducción
- 2 - Novatos
- 3 - Power Users
- 4 - Desarrolladores

1 - Introducción

2 - Novatos

3 - Power Users

4 - Desarrolladores

¿Qué es VIM?

1

- Editor de texto, no gráfico (en ppio).

2

- Clon de VI, ampliado.

3

- Autor principal: Bram Moolenaar

4

Puntos positivos

- 1 • Potencia, escaso gasto de recursos.
- 2 • Disponibilidad: estándar en Unix (bash -o vi, dselect, less, ...).
- 3 • Especializado: “Do one thing, do it well”
- 4 • Velocidad: touch-typing.

Puntos negativos

1

- Inclínada curva de aprendizaje.

2

- Falta de intuitividad.

3

- Es sólo un editor.

4

- Orientado al teclado (en principio).

1 - Introducción

2 - Novatos

3 - Power Users

4 - Desarrolladores

Instalación de VIM 7

- Herramientas de tu distribución Linux

1 # apt-get install vim

emerge vim

synaptic

2 # ...

- A mano:

3 \$ <http://www.vim.org/download.php>

\$./configure --prefix=/home/x/usr

4 \$ make && make install

\$ PATH=/home/x/usr/bin:\$PATH

Guía de supervivencia

- Abrir un fichero:

1 \$ vim fichero.txt

- Modificar contenidos:

2 i<modificar fichero><esc>

- Salir guardando cambios:

3 :wq

- Salir sin guardar cambios:

4 :q!

Guía de supervivencia

- Ayuda:

1

\$ man vim

:help <lo-que-sea>

(se puede usar autocompletado con tab)

2

- Ayuda interactiva (esencial!):

\$ vimtutor

3

- Para todo lo demás, googlear ;-)

4

Modos

• Modo comando:

1 <esc>

• Modo insercion:

i I a A o O c <movimiento> ...

2 • Modo reemplazo

r R

3 • Modo visual:

v V <c-v>

• Modo completado:

4 <c-x> <c-...>

Movimientos básicos

1 • Una posición: hjkl

• Una "palabra": w W b B e E

2 • Inicio fichero: gg

• Fin de fichero: G

• Línea X: <X>gg

3 • Inicio pantalla: H

• Fin de pantalla: L

4 • Centrar en pantalla (no es mov.): zz

Cortar-Copiar-Pegar

- Visualmente: $v \quad V \quad <c-v> \quad <mov.>$

1

<u>copiar</u>	<u>cortar</u>
y	x

- Línea actual: yy

dd

- Genérico: $y<mov.> \quad d<mov.>$

3

E incluso tras cerrar vim....:

- Pegar antes: P

4

- Pegar después: p

- Pegar portapapeles X: $i<s-ins>$

Buscar-reemplazar

- Iniciar búsqueda genérica:

1 /<palabra>

?<palabra>

- Iniciar búsqueda palabra bajo cursor:

2 * #

- Continuar búsqueda:

3 n N

- Reemplazar regexp interactivo:

:%s/expresion1/expresion2/gc

- Reemplazar regexp automáticamente:

4 :%s/expresion1/expresion2/g

Deshacer

- Deshacer cambio:

1 u

- Rehacer cambio:

2 <c-r>

3

4



- 1 - Introducción
- 2 - Novatos
- 3 - Power Users**
- 4 - Desarrolladores

Repetir tareas

- Una cifra delante de un comando:

1 Borrar 15 líneas: 15dd

Borrar 3 palabras: 3dw

Bajar 5 líneas: 5j

2 Deshacer 2 niveles: 2u

100iNo hablaré más en clase
<esc>

3 • Repetir último comando: .

4 • Grabar macro: q<registro><...>q

• Reproducir macro: @<registro>

Historial

- Historial de comandos:

1 q:<mov.>

- Historial de búsqueda:

2 q/<mov.>

3 [estos dos son modificables!]

- Historial de portapapeles (pegar):

4 “<registro numérico>p

Comandos varios

1 • Crear marca: m<registro>

• Ir a marca: '<registro>

• Último cambio: "

[pueden usarse como mov. con mas comandos!]

2 • Ver fichero actual: 2<c-g>

• Completar filename: <c-x><c-f>

3 • Cambiar case: ~

• Ir a fin de línea: \$

4 • Ir a inicio de línea: 0

Comandos varios

- Ventanas:

1

Crear fila: `<c-w>s`

Crear columna: `<c-w>v`

2

Cerrar: `<c-w><c-c>`

Ocultar: `:hide
`

3

Dejar actual: `<c-w><c-o>`

4

Navegar: `<c-w>h j k l`

Más info: `:help window
`



Comandos varios

- Buffers:

1 Crear buffers: `$ vim -o *.cpp *.h`

2 Siguiete buffer: `:next
`

Anterior buffer: `:prev
`

3 Más info: `:help buffers
`
transparencias para
desarrolladores (**4**).

4 <-----|

Personalizar vim

La central de vim (tips & scripts):

1 < www.vim.org >

2 • Configuración de usuario:

```
$ vim ~/.vimrc
```

```
$ ls ~/.vim
```

3 • Scripts:

```
$ ls ~/.vim/plugins
```

4

1 - Introducción

2 - Novatos

3 - Power Users

4 - **Desarrolladores**

“Modo IDE”

- 1 • Personalización de VIM creada por un servidor (STenyaK :P), a través de vimrc.
- 2 • Muy mejorable (ampliar, bugfixes, script).
• Muy warro (shortcuts overrideados O:-).
- 3 • Activar/desactivar: `<f1>`
• Ventana de buffer actual: `<c-w>` `<c-e>`
- 4 • Necesita algunos plugins para funcionar:
TList, CppOmniComplete, WinManager.

Sesiones

•¿Qué son?

1 Permite almacenar en un fichero el estado de vim (archivos abiertos, variables modificadas, preferencias actuales, ...)

2 •Guardar sesión: `:mksession! fichero`

3 •Cargar sesión: `$ vim -S fichero`

4 A veces hay problemas con ciertos plugins, hay que tener cuidado :(

Folding (*"plegado"*?)

- 1 • Colapsar actual: ck
map ck \$]}V%kzf
- 2 • Expandir actual: cj
map cj z0
- 3 • Colapsar recursivo: CK
map CK ckckckckckckck o:-)
- 4 • Expandir recursivo: CJ
map CJ z0

Navegación a través del código

1 • Creación de índice de tags (imperfecto!):

```
$ ctags-exuberant -l --c++-kinds=+p --fields=+iaS --extra=+q -f tags *cpp *h
```

2 • Ir a posición de símbolo actual: fj

...en nueva ventana: Fj

• Volver a posición anterior: fk

...desde nueva ventana: FK

3 • Siguiente ocurrencia: fn

• Anterior ocurrencia: fp

4 • Ver pila de posiciones: fm

Autocompletado (aka IntelliSense)

- 1 •Actualizar tags manual: <f6>
•Actualizar tags make: <f5>
- 2 •Modo tradicional: <c-x> <c-ofn>
•Activar “Modo IDE”: <f1>
- 3 Mostrar lista (no al inicio): <tab>
Siguiendo elemento: <tab>
Usar elemento:

- 4 Cancelar completado: <esc>

Navegar tags

- 1 • Activar “Modo IDE”: `<f1>`
- 2 • Siguiente tag: `<c-l>`
• Anterior tag: `<c-h>`
- 3 • Ir a ventana de tags: `<c-w>` `<c-r>`
- 4 • Navegación: `<como en vim>`
• Buscar tag: `<como en vim>`
• Ir a tag: `
`

Navegar ficheros

- 1 • Activar “Modo IDE”: `<f1>`
- 2 • Ir a ventana de filesystem: `<c-w><c-e>`
- 2 • Navegación: `<como en vim>`
- 2 • Buscar fichero `<como en vim>`
- 3 • Abrir fichero: `
`
- 3 • Abrir directorio: `
`
- 3 • Nivel superior: `-`
- 4 • Más info: `?`

Navegar buffers

- 1
 - Activar “Modo IDE”: `<f1>`
 - Siguiente buffer: `<c-j>`
 - Anterior buffer: `<c-k>`
- 2
 - Ir a ventana de buffers: `<c-w>` `<c-d>`
- 3
 - Navegacion: `<como en vim>`
 - Buscar buffer: `<como en vim>`
 - Mostrar fichero: `
`
- 4
 - Cerrar buffer: `d`
 - Más info: `?`

Binding a “make”

- 1 • Ejecutar make: `:make
`
• Limpiar: `:make clean
`
• Etc... `:make whatever
`
- 2 • Lista de errores: `:clist
`
- 3 • Siguiete error: `:cnext
`
• Anterior error: `:cprev
`
• Último error: `:clast
`
- 4 • Recomendable comando “`colorgcc`” :-)

Binding a “gdb”

•gdbvim (script #84)

1 •vimDebug (script #663):

Iniciar gdb: <f12>

Cerrar gdb: <f11>

2 Next: <f6>

Step: <f7>

3 Continue: <f8>

Añadir bp: \b

Quitar bp: \c

4 •Ojo con bucles infinitos! gdb, ddd, kdbg..

Varios

- 1 • Ir a definición de símbolo: `gd`
- 1 • Ir a inicio-fin de `()`, `[]`, `{}`: `%`
- Moverse por párrafos, frases, etc.:
2 `:help object-motions
`
- Indentar izq/der: `<` `>`
- 3 • Indentar auto: `=`
- Inc/decrementar número: `<c-a>` `<c-x>`
- 4 • Comparar ficheros: `$ vimdiff file1 file2`

Fin de la charla :P

¿Dudas?

**Transparencias y .vimrc
disponibles en:**
[por determinar]

**Críticas, alabanzas,
insultos, preguntas y
spam a:**
STenyaK (Bruno González)
stenyak@gmail.com



:qa!