

Vim

Claudio Jiménez Castillo

Grupo de Usuarios de Linux
Universidad Carlos III de Madrid.



5 de Marzo de 2006

Introducción.

¿Que es Vim?

- ▶ Vim ("VI IMproved"), basado en el Vi.
- ▶ Es un editor de textos.
- ▶ Desarrollado por Bram Moolenaar.
- ▶ Licencia GPL.

Ventajas

- ▶ Es Open Source.
- ▶ Es altamente configurable.
- ▶ Tiene resaltado para el código.
- ▶ Es charity-ware.
- ▶ No ocupa demasiado espacio.
- ▶ Uso muy similar al Vi.
- ▶ Permite muchos cambios con pocas combinaciones de teclas.
- ▶ Comodo de usar una vez se conoce.
- ▶ Existen versiones para muchos sistemas operativos.

Modos

Modos

Existen 3 modos:

Comandos Es el básico, a partir de aquí se ejecuta todo.

Insertar Es el modo que nos permite insertar texto.

Visual Este modo nos permite hacer selecciones de texto y realizar operaciones sobre el.

Modo comandos

Es el modo en el que se ejecutan todos los comandos del Vim. Es el modo base del que se parte para hacer la mayoría de las operaciones sobre el texto.

Modo insertar

Es el modo que nos permite insertar texto dentro del fichero.

Modo visual

Este modo nos permite hacer selecciones visuales del texto del fichero y hacer operaciones sobre el. Existen tres tipos de modo visual en razón al modo de selección:

- ▶ Por caracter: Se selecciona el texto por caracteres (pulsar `v`).
- ▶ Por línea: Se selecciona el texto por líneas (pulsar `V`).
- ▶ Por bloque: Se selecciona el texto por bloques (pulsar `^v`).

Comandos

Comandos Básicos

Existen muchísimos comandos, pero los mas básicos serian los siguientes:

- ▶ Movimientos
- ▶ Inserción
- ▶ Borrado
- ▶ Copiar/Pegar
- ▶ Búsquedas
- ▶ Otros

Comandos Básicos: Movimientos I

- ▶ **h**: Mueve el cursor un caracter a la izquierda.
- ▶ **j**: Mueve el cursor un caracter abajo.
- ▶ **k**: Mueve el cursor un caracter arriba.
- ▶ **l**: Mueve el cursor un caracter a la derecha.
- ▶ **w**: Mueve el cursor al principio de la siguiente palabra.
- ▶ **b**: Mueve el cursor el principio de la palabra.
- ▶ **e**: Mueve el cursor el final de la palabra.

Comandos Básicos: Movimientos II

- ▶ `0`: Mueve el cursor a la primera columna.
- ▶ `|`: Mueve el cursor a la primera columna.
- ▶ `n|`: Mueve el cursor a la columna `n`.
- ▶ `^`: Mueve el cursor a la primera columna no blanca.
- ▶ `$`: Mueve el cursor a la última columna.
- ▶ `-`: Mueve el cursor al comienzo de la línea anterior.
- ▶ `+`: Mueve el cursor al comienzo de la línea siguiente.

Comandos: Movimientos III

- ▶ %: Mueve el cursor al parentesis o llave asociado.
- ▶ (: Mueve el cursor al comienzo de la frase.
- ▶): Mueve el cursor al final de la frase.
- ▶ {: Mueve el cursor al comienzo del párrafo.
- ▶ }: Mueve el cursor al final del párrafo.
- ▶ [[: Mueve el cursor al comienzo de una sección.
- ▶]]: Mueve el cursor al final de una sección.

Comandos Básicos: Movimientos IV

- ▶ H: Mueve a la primera línea de la pantalla.
- ▶ nH: Mueve a la línea n de la pantalla.
- ▶ L: Mueve a la última línea de la pantalla.
- ▶ nL: Mueve a la línea n de la pantalla (empezando por abajo).
- ▶ ^B: Mueve una pantalla arriba.
- ▶ ^U: Mueve media pantalla arriba.
- ▶ ^D: Mueve media pantalla abajo.
- ▶ ^F: Mueve una pantalla abajo.
- ▶ ^Y: Mueve una línea arriba.
- ▶ ^E: Mueve una línea abajo.

Comandos Básicos: Movimientos V

- ▶ **1G**: Mueve al principio del fichero.
- ▶ **nG**: Mueve a la línea n del fichero.
- ▶ **G**: Mueve al final del fichero.

Comandos Básicos: Insercion

- ▶ I: Inserción al principio de la línea.
- ▶ i: Inserción en el caracter bajo el cursor.
- ▶ a: Inserción en el caracter siguiente.
- ▶ A: Inserción al final de la línea.
- ▶ o: Inserción en una nueva línea después de la actual.
- ▶ O: Inserción en una nueva línea previa a la actual.
- ▶ r: Sustituye el caracter bajo el cursor.
- ▶ R: Entra en modo reemplazo la posición del cursor.

Comandos Básicos: Borrado

- ▶ `x`: Borra el caracter actual.
- ▶ `nx`: Borra `n` caracteres.
- ▶ `X`: Borra un caracter anterior al actual.
- ▶ `nX`: Borra `n` caracteres anteriores al actual.
- ▶ `dd`: Borra una linea.
- ▶ `ndd`: Borra `n` lineas.

Comandos Básicos: Copiar/Pegar

- ▶ `y`: Copia el caracter actual, o la selección que tengamos hecha.
- ▶ `Y`: Copia la línea actual.
- ▶ `yy`: Copia la línea actual.
- ▶ `yny`: Copia `n` líneas.
- ▶ `p`: Pega en la línea siguiente.
- ▶ `P`: Pega en la línea anterior.

Comandos Básicos: Búsquedas

- ▶ `/<cadena>`: Busca la cadena hacia adelante.
- ▶ `?<cadena>`: Busca la cadena hacia detrás.
- ▶ `n`: Repite la ultima búsqueda.
- ▶ `N`: Repite la búsqueda invirtiendo el sentido.

Comandos Básicos: Otros

- ▶ ZZ: Sale y guarda el fichero.
- ▶ :wq: Sale y guarda el fichero.
- ▶ :x: Sale y guarda el fichero.
- ▶ :view <file>: Abre un fichero para solo lectura.
- ▶ :w: Escribe el fichero.
- ▶ :q!: Sale del programa.
- ▶ u: Deshace la última operación.
- ▶ ^r: Rehace el último deshacer.
- ▶ J: Une dos líneas.
- ▶ .: Repite la última operación realizada.

Otros comandos

- ▶ Sustituciones
- ▶ Abreviaturas
- ▶ Macros
- ▶ Marcas
- ▶ Plegado
- ▶ Grabación de comandos
- ▶ Comandos del sistema
- ▶ Configuraciones
- ▶ Combinaciones

Otros comandos: Sustituciones

- ▶ `:%s/<expreg>/<cadena>/`: Sustitulle la primera vez que encuentre la expresion regular en una linea, por la cadena.
- ▶ `:%s/<expreg>/<cadena>/g`: Sustitulle todas las veces que encuentre la expresion regular por la cadena.
- ▶ `:<rango>s/<expreg>/<cadena>/g`: Sustituye todas las veces que encuentre la expresión regular dentro del rango, por la cadena.

Otros comandos: Abreviaturas

Son prácticas para escritura rápida y para predefinirlas en un fichero dependiendo el documento que vayas a editar.

- ▶ `:abbreviate <cadena1> <cadena2>`: Abrevia la cadena 2 como la cadena 1.
- ▶ `:abbreviate`: Muestra las abreviaturas actuales.

Otros comandos: Macros

Las macros son secuencias de comandos que se asignan a un caracter.

- ▶ `:map C <cadena>`: define un macro.
- ▶ `C`: ejecuta el macro definido en `C`.

ejemplo: `:map <F5> icadena de prueba <ESC>`

Otros comandos: Marcas

Las marcas se utilizan para memorizar posiciones dentro del fichero.

- ▶ `mC`: Marca una posición con el caracter C.
- ▶ `'C`: Va a la posición marcada con C.
- ▶ `:marks`: Muestra todas las marcas que existen en el documento.

Las marcas se pueden usar también en el rango:

`: 'C, 'Ds/casa/casita/g`

Otros comandos: Plegado

El plegado se utiliza para ocultar parte del texto del archivo que no queremos ver.

- ▶ `:<rango> fold`: Pliega el rango.
- ▶ `:<rango> foldddopen <comando>`: Ejecuta el comando sobre todo lo que no esté dentro de un pliegue cerrado.
- ▶ `:<rango> folddoclosed <comando>`: Ejecuta el comando sobre todo lo que esté dentro de un pliegue cerrado.
- ▶ `:<rango> foldopen[!]`: Abre los pliegues en el rango.
- ▶ `:<rango> foldclosed[!]`: Cierra los pliegues en el rango.
- ▶ Para `foldopen` y `foldclose` el `!` sirve para abrir también pliegues anidados.

Otros comandos: Plegado II

Algunos comandos utiles para el plegado.

- ▶ `zd`: Borra el pliegue sobre el que se encuentre.
- ▶ `zD`: Borra el pliegue sobre el que se encuentre y todos los anidados.
- ▶ `zE`: Borra todos los pliegues del archivo.
- ▶ `zo`: Abre el pliegue sobre el que se encuentre.
- ▶ `zO`: Abre el pliegue sobre el que se encuentre y todos los anidados.
- ▶ `zc`: Cierra el pliegue sobre el que se encuentre.
- ▶ `zC`: Cierra el pliegue sobre el que se encuentre y todos los anidados.
- ▶ `za`: Cambia de estado (cerrado/abierto) del pliegue sobre el que se encuentre.
- ▶ `zA`: Cambia de estado (cerrado/abierto) del pliegue sobre el que se encuentre y todos los anidados.

Otros comandos: Grabación de comandos

La grabación de comandos nos permite grabar una secuencia de comandos que ejecutamos para repetirla posteriormente con una sencilla combinación de teclas.

- ▶ qC: Empieza a grabar una sucesión de comandos que ejecutes.
- ▶ q: Termina de grabar.
- ▶ @C: Ejecuta de nuevo la misma sucesión de comandos.

Otros comandos: Comandos del sistema

En Vim es posible ejecutar comandos del sistema sobre parte o todo el texto que estamos editando.

- ▶ `:<rango>!<comando>`: Ejecuta el comando con entrada estándar el rango que se le ha especificado y lo sustituye por la salida del comando.

Otros comandos: Configuraciones

El comando `set` nos sirve para configurar nuestro Vim.

- ▶ `:set all`: Muestra todas las configuraciones.
- ▶ `:set nu/nonu`: Muestra el numero de linea.
- ▶ `:set hlsearch/nohlsearch`: Añade o quita remarcado a las búsquedas.
- ▶ `:set filetype=<tipo>`: Especifica el tipo de fichero que se esta editando.
- ▶ `:set textwidth=n`: Especifica el ancho del documento.

Otros comandos: Combinaciones

Existen algunas combinaciones simples que se usan con cierta frecuencia.

- ▶ `xp`: Cambia de orden dos caracteres.
- ▶ `ddp`: Cambia de orden dos líneas.

Ventanas

Ventanas I

- ▶ `:split`: Abre una nueva ventana dividiendo en horizontal.
- ▶ `:vsplit`: Abre una nueva ventana dividiendo en vertical.
- ▶ `:split <file>`: Abre el fichero en una nueva ventana.
- ▶ `:n split`: Abre una ventana de tamaño n.
- ▶ `:sview`: Combinación entre `:split` y `:view`.

Ventanas II

- ▶ `^W w`: Cambia a la ventana siguiente.
- ▶ `^W j`: Cambia a la ventana de abajo.
- ▶ `^W k`: Cambia a la ventana de arriba.
- ▶ `^W -`: Disminuye el tamaño de la ventana.
- ▶ `^W +`: Aumenta el tamaño de la ventana.
- ▶ `^W =`: Iguala el tamaño de las ventanas.

Para programadores

Para programadores

- ▶ Configuraciones.
- ▶ Make.
- ▶ Ctags.

Configuraciones

- ▶ `:syntax on/off`: Activa o desactiva el reconocimiento de sintaxis.
- ▶ `:set autoindent/noautoindent`: Activa o desactiva el auto indentado.
- ▶ `:set cindent/smartindent/autoindent`: Selecciona el tipo de auto indentado que se usa.

Make |

- ▶ `:make <argumentos>`: Ejecuta el make con los argumentos que se le pasen.
- ▶ `:cnext`: Mueve el cursor a la posición del error siguiente (línea y fichero).
- ▶ `:cprevious`: Mueve el cursor a la posición del error anterior (línea y fichero).
- ▶ `:clast`: Mueve al último error.
- ▶ `:crewind`: Mueve al primer error.
- ▶ `:cnfile`: Mueve al primer error del próximo fichero.

Make II

- ▶ `:cc`: Muestra el error actual.
- ▶ `:clist`: Muestra la lista de errores.
- ▶ `:clist <rango>`: Muestra el rango de errores pedido.
- ▶ `:clist!`: Muestra también los mensajes de información del compilador.

Ctags

Búsqueda de definición de las funciones con ctags.

- ▶ `^]`: Va a la definición de la función.
- ▶ `^t`: Vuelve a la posición donde estaba.

Para terminar

Referencias

- ▶ `vimtutor`
- ▶ <http://www.vim.org> - Web oficial de Vim.
- ▶ <http://www.truth.sk/vim/vimbook-OPL.pdf> - Libro de Vim.
- ▶ <http://gul.es/mailman/listinfo/gul> - Lista de correo del gul.
- ▶ <http://bullium.com/support/vim.html>

Dudas

...

Agradecimientos

- ▶ Gracias a Jesús Espino por dejarme sus transparencias, en las que estan basadas estas.
- ▶ Gracias a todos los que han ayudado para que estos cursos hayan podido salir adelante.
- ▶ Y gracias a ustedes por venir.

Fin

