

# Introducción de comandos en Vim

## Vim version 6.0

auxiliar para EOLibre

10 de noviembre de 2003

## Índice

<b>1. Modos en vim.</b>	<b>3</b>
<b>2. Objetos de texto.</b>	<b>3</b>
<b>3. Comandos de movimiento.</b>	<b>3</b>
3.1. Simples. . . . .	3
3.2. Marcas. . . . .	3
<b>4. Comandos de búsqueda.</b>	<b>4</b>
<b>5. Comandos de vista.</b>	<b>5</b>
5.1. Plegado de texto. . . . .	5
5.2. Ventanas. . . . .	5
<b>6. Órdenes.</b>	<b>6</b>
6.1. Comandos de manipulación. . . . .	6
6.1.1. Comandos de cambio a modo de inserción. . . . .	6
6.1.2. Comandos con permanencia en modo normal. . . . .	6
6.2. Línea de comandos (":"). . . . .	7
6.3. Ejemplos. . . . .	7
<b>7. Sobre este documento.</b>	<b>8</b>

## Índice de cuadros

1.	Objetos de texto. . . . .	3
2.	Comandos de movimiento simples. . . . .	4
3.	Comandos de movimiento por marcas. . . . .	5
4.	Comandos de movimiento por marcas. . . . .	5
5.	Comandos de vista. Plegado de texto. . . . .	5
6.	Comandos de vista. Ventanas. . . . .	6
7.	Comandos de cambio a modo de inserción. . . . .	6
8.	Comandos sin ámbito. . . . .	7
9.	Comandos con ámbito. . . . .	7
10.	Indicación de ámbito. . . . .	8
11.	Ejemplos. . . . .	9

## 1. Modos en vim.

- Modo normal, que sirve para introducir comandos (de movimiento, de búsqueda, etc...).
- Modo inserción, que permite escribir texto en sí.

En el modo normal hay que distinguir los comandos introducidos a ciegas de los introducidos después de un “:” (ver sección 6.2).

## 2. Objetos de texto.

Son porciones de texto que podemos manipular (borrar, copiar, etc.) como una unidad (ver cuadro 1).

---

aw (palabra)	delimitada por un espacio, una tabulación, una coma, etc...
iw (palabra)	idem, pero estrictamente la palabra sin el espacio siguiente.
as (frase)	delimitada por un punto (.) o una línea en blanco.
is (frase)	idem, pero estrictamente la frase sin el espacio siguiente.
ap (párrafo)	delimitado por una línea en blanco.
ip (párrafo)	idem, pero estrictamente el párrafo sin la línea en blanco siguiente.
ab (bloque de paréntesis)	es lo que encierra una pareja de (), ellos incluidos.
ib (bloque de paréntesis)	es lo que encierra una pareja de ().
aB (bloque de llaves)	es lo que encierra una pareja de {}, ellas incluidas.
iB (bloque de llaves)	es lo que encierra una pareja de {}.

---

Cuadro 1: Objetos de texto.

## 3. Comandos de movimiento.

### 3.1. Simples.

Básicamente permiten el movimiento cómodo y rápido a través del texto en modo normal (ver cuadro 2).

En los comandos de movimiento en que tenga sentido (todos menos “M” y “%”), se admite un número antes del propio comando que indica el número de veces que se aplica. Por ejemplo, “3w” indica moverse tres palabras adelante (como “www”); “3\$”, moverse al tercer final de línea (que eá en la segunda siguiente línea), etc. Nótese que un “2}” no va al segundo “}” que se encuentre vim, sino al “}” que cierra el bloque que contiene al bloque actual.

### 3.2. Marcas.

Permiten marcar un caracter del texto para poder posteriormente acceder a él de forma rápida (ver cuadro 3).

---

k,j,h,l	Arriba, abajo, izquierda, derecha.
w	Moverse al principio de la palabra siguiente.
b	Moverse al principio de la palabra anterior.
f<c>	Ir a la siguiente ocurrencia en la línea del caracter <c>.
t<c>	Ir antes de la siguiente ocurrencia en la línea del caracter <c>.
F<c>	Ir a la anterior ocurrencia en la línea del caracter <c>.
T<c>	Ir antes de la anterior ocurrencia en la línea del caracter <c>.
(	Principio de la frase.
)	Final de la frase.
{	Principio del párrafo.
}	Final del párrafo.
0	Principio de línea.
^	Primer caracter de la línea.
\$	Final de línea.
g0	Principio de línea de la pantalla (no la real).
g^	Primer caracter de la línea de la pantalla (no la real).
g\$	Final de línea de la pantalla (no la real).
gk	Línea arriba de pantalla (no la real).
gj	Línea abajo de pantalla (no la real).
gg	Principio de la primera línea del documento.
G	Principio de la última línea del documento.
H	Primera línea de la pantalla.
M	Principio de la línea en mitad de la pantalla.
L	Última línea de la pantalla.
n	Ir a la siguiente ocurrencia de la búsqueda.
N	Ir a la anterior ocurrencia de la búsqueda.
	Ir a la primera columna.
[{	Moverse al { que abre el bloque actual de llaves.
]}	Moverse al } que cierra el bloque actual de llaves.
[(	Moverse al ( que abre el bloque actual de paréntesis.
] )	Moverse al ) que cierra el bloque actual de paréntesis.
[*	Moverse al /* que abre el comentario.
]*	Moverse al */ que cierra el comentario.
%	Situado sobre un delimitador de bloque, mueve el cursor a su pareja.
<n>%	Ir al <n> % del documento.

---

Cuadro 2: Comandos de movimiento simples.

## 4. Comandos de búsqueda.

“\*”, “n” y “N” admiten un número como prefijo. Por ejemplo, “2n” se mueve a la segunda siguiente ocurrencia del patrón, es decir equivale a “nn” (ver cuadro 4).

---

m<c>	Crea la marca en la posición del cursor. <c> es un caracter alfabético, es decir, [A-Za-z].
'<c>	Va al principio de la línea que contiene la marca <c>.
'<c>	Va exactamente al caracter en que se originó la marca <c>.
''	Va al principio de la línea en que se encontraba el cursor antes del salto a la marca.
''	Va exactamente al caracter en que se encontraba el cursor antes del salto a la marca.

---

Cuadro 3: Comandos de movimiento por marcas.

---

/<patrón>	Busca  patrón . Avanza a su siguiente ocurrencia.
?<patrón>	Busca  patrón . Retrocede a la ocurrencia anterior.
*	Busca la siguiente ocurrencia de la palabra actual.
\#	Busca la anterior ocurrencia de la palabra actual.
n	Va a la siguiente ocurrencia de la búsqueda.
N	Va a la anterior ocurrencia de la búsqueda.

---

Cuadro 4: Comandos de movimiento por marcas.

## 5. Comandos de vista.

### 5.1. Plegado de texto.

El plegado consiste en el embutido de un conjunto de líneas en una sola. Me gusta especialmente el plegado por marcas (":set foldmethod=marker"), aunque existen otros. Para manejar esta característica ver cuadro 5.

---

zo	Abre el pliegue actual.
zO	Abre el pliegue actual y todos los que están dentro de él.
zc	Cierra el pliegue actual.
zC	Cierra el pliegue actual y todos los que están dentro de él.
zR	Abre todos los pliegues del documento.
zM	Cierra todos los pliegues del documento.
zf	Crea un pliegue manual.

---

Cuadro 5: Comandos de vista. Plegado de texto.

El último comando necesita un ámbito de actuación, pues crea un pliegue sobre la marcha que dura el tiempo que está abierto el documento. Se explicará en la sección 6.1.2 cómo deben usarse este tipo de comandos.

### 5.2. Ventanas.

**vim** permite dividir la pantalla en varias ventanas para editar varios ficheros a la vez. Una

vez creadas, el modo de cambiar de ventana es sencillo siguiendo el cuadro 6.

---

Ctrl+wk	Va a la ventana superior.
Ctrl+wj	Va a la ventana inferior.
Ctrl+wh	Va a la ventana de la izquierda.
Ctrl+wl	Va a la ventana inferior.

---

Cuadro 6: Comandos de vista. Ventanas.

## 6. Órdenes.

Bajo esta sección se describen los comandos que sirven para alterar el texto ya escrito o introducir nuevo.

### 6.1. Comandos de manipulación.

Se engloban aquí los comandos ciegos (los que se teclean sin que llegue a haber echo en pantalla). Principalmente cabe destacar dos tipos: los comandos que no necesitan ámbito de aplicación (p.e. pegar un texto o deshacer la acción previa) y los comandos que sí lo necesitan (p.e. borrar, porque hay que indicarle a **vim** qué es lo que queremos borrar, ¿no?).

Además, hay unos pocos comandos que nos permiten cambiar al modo inserción para que sigamos escribiendo texto, mientras que otros alteran el texto y mantienen el modo normal.

#### 6.1.1. Comandos de cambio a modo de inserción.

---

a	Inserta tras la posición del cursor.
A	Inserta al final de la línea actual.
i	Inserta en la posición del cursor.
I	Inserta al comienzo de la línea actual.
R	Reemplaza.
o	Abre línea tras la actual e inserta.
O	Abre línea antes de la actual e inserta.
c	Sustituye e inserta.
<ESC>	Regresar a modo normal.

---

Cuadro 7: Comandos de cambio a modo de inserción.

Ninguno de los comandos tiene ámbito, excepto “c”, del que se explicará más adelante su uso (ver cuadro 7).

#### 6.1.2. Comandos con permanencia en modo normal.

Al igual que los comandos de movimiento admiten como prefijo un número que multiplica el efecto del comando.

**Comandos sin ámbito.** Ver cuadro 8.

## Comandos con ámbito.

**Lista de comandos.** Al contrario que los comandos sin ámbito (la mayoría de los que se han listado hasta ahora), los comandos que se listan a continuación no pueden ser usados tecleando su símbolo exclusivamente. En el próximo apartado se indicará cómo deben usarse (ver cuadro 9).

**¿Cómo se indica el ámbito?** Todos estos comandos (junto con “c”) no son suficientes de por sí: necesitan que se les especifique un ámbito de actuación. Existen cinco medios de lograrlo (ver cuadro 10).

### 6.2. Línea de comandos (“:”).

Por hacer.

### 6.3. Ejemplos.

Ver cuadro 11.

---

p	Pega el texto del buffer tras la posición del cursor.
[p	Idem, pero respetando la indentación del entorno en que se pega.
P	Pega el texto del buffer tras la posición del cursor.
[P	Idem, pero respetando la indentación del entorno en que se pega.
u	Deshace.
Ctrl+R	Rehace.
.	Repite el último comando.
r<c>	Reemplaza el caracter actual por el caracter <c>.
x	Borra el caracter actual.
J	Fusiona la línea actual con la siguiente.
Ctrl+P	Completa la palabra buscando una que empiece igual antes.
Ctrl+N	Completa la palabra buscando una que empiece igual después.
gf	Abre el fichero cuyo nombre es el de la palabra actual.

---

Cuadro 8: Comandos sin ámbito.

---

d	Corta (es decir, borra y coloca en el buffer).
y	Copia (al buffer).
gu	Convertir en minúsculas.
gU	Convertir en mayúsculas.
>	Indentar.
<	Des-indentar.
gq	Corta las líneas al ancho prefijado por “textwidth”.

---

Cuadro 9: Comandos con ámbito.

---

Movimiento	Añadir después de la orden un comando de movimiento: extiende el ámbito desde la posición del cursor hasta la posición a la que llevaría el comando de movimiento el cursor. Por ejemplo, “ <b>d\$</b> ” borra hasta el final de la línea actual; “ <b>gUG</b> ” pasa a mayúsculas todas las palabras hasta el final del archivo, etc...
Objeto de texto	Añadir después de la orden un objeto del texto: el ámbito de actuación de la orden es dicho objeto. Por ejemplo, “ <b>dap</b> ” borra todo el párrafo actual; “ <b>daB</b> ” borra todo el bloque encerrado entre llaves.
Visual	En los dos casos anteriores la definición del ámbito se hace a ciegas y sólo se puede realizar o con un único movimiento o con un único objeto. Este tercer método nos permite ver lo que seleccionamos y, a la vez, hacer elecciones complejas reuniendo juntos varios movimientos y elecciones de objetos de texto a la vez. Consiste en presionar la letra “ <b>v</b> ”, a continuación movernos usando los comandos de movimientos (lo que hará que lo seleccionado aparezca en vídeo inverso) y, al completar la selección, teclear la orden. Por ejemplo, “ <b>v}apd</b> ” borra (“ <b>d</b> ”) hasta el final del párrafo (“ <b>}</b> ”) y todo el párrafo siguiente ( <b>ap</b> ). En vez de “ <b>v</b> ” puede usarse “ <b>V</b> ” para trabajar con líneas completas o “ <b>Ctrl+v</b> ” para trabajar con bloques de texto. Por ejemplo, “ <b>~v2j2ld</b> ” borraría un área de 2 filas (“ <b>2j</b> ”) por dos columnas (“ <b>2l</b> ”).
Duplicidad	Los comandos de este tipo de un sólo carácter (y incluso algunos de dos) suelen admitir su duplicidad, es decir, “ <b>dd</b> ” a partir de “ <b>d</b> ”. En este caso, lo que ocurre es que el comando se aplica a un ámbito predefinido (normalmente la línea). Así, “ <b>dd</b> ” corta una línea, “ <b>yy</b> ” copia una línea, “ <b>&gt;&gt;</b> ” indenta una línea, etc...
Línea de comandos	En este método se usa la línea de comandos. Consiste en teclear dos puntos (:) lo que nos posibilita entrar en la línea de comandos, escribir el rango y, por último, especificar el comando. Por ejemplo, “ <b>:10,30d</b> ” borra de la línea 10 a la línea 30 del documento.

---

Cuadro 10: Indicación de ámbito.

## 7. Sobre este documento.

Este documento ha sido tipografiado con el paquete de macros  $\text{\LaTeX} 2_{\epsilon}$  en una arquitectura i686 con sistema operativo GNU/Linux. Este documento está en constante revisión y actualización. Si observas alguna errata o error de sintaxis, por favor comunícalo para poder corregirlo.

Se permite la copia total o parcial de éste documento siempre que se indique la procedencia y autor del mismo, además de esta nota al final.



---

xp	Trabuca dos caracteres. Los hay que escriben muy mla. ;-)
[{d}]	Se mueve al principio de un bloque y borra hasta el final del mismo.
v10ly	Copia los 10 caracteres que siguenb al cursor.
2d2ap	Borra dos veces dos párrafos, es decir, borra cuatro párrafos.
y1	Copia el caracter actual.
caw	Sustituye la palabra actual, es decir, la borra y entra en modo inserción.
>iB	Indentar un bloque de llaves.
72	Va a la columna 72.
5G	Va a la línea 5.
5gg	Va a la línea 5.
zfaB	Pliega todo el bloque de paréntesis.
guu	Pone en minúsculas toda la línea.
gUU	Pone en mayúsculas toda la línea.
78 F r<ENTER>	Corta la línea actual por la columna 78.
gqqq	Formatea sólo la línea actual.

---

Cuadro 11: Ejemplos.