

## Curso Básico de Editor Vi

Vi permite realizar sofisticadas operaciones de búsqueda y reemplazo de patrones que no se encuentran en la mayoría de los editores. Esto debido a que vi, es realmente la unión de dos grandes editores, *ex* y el propio *vi*. Puesto que *ex* fue muy potente en su época y aún lo es, *vi* hace uso del editor *ex* y propone otras mejoras que poco a poco se van actualizando, debido a las distribuciones que salen al mercado.

Al ser *ex* integrado al editor *vi*, todos los comandos que empiezan con dos puntos (:) son pasados a *ex* para su ejecución. La mayoría de los comandos de *ex* siguen el mismo patrón que los comandos para copiar, mover o borrar líneas de texto en *vi* (:x,yCz ó :[rango][comando][opciones]); los dos puntos quiere decir que se refiere a un comando *ex*, el rango se refiere al bloque de líneas que se va a tratar, si se omite, el comando se ejecuta sobre la línea actual nadamás, después sigue el comando en sí y al último las opciones o parámetros de dicho comando que también pueden ser omitidas.

Dentro del modo de comandos, la mayoría de comandos se escriben después de ':', pero para poder entenderlos mejor, se clasifican en tres casos y se listan los comandos principales.

### Comandos de navegación

\$	Ir al final de la línea actual
/<cadena>	Ir a la siguiente ocurrencia de la cadena especificada
?<cadena>	Ir a la anterior ocurrencia de la cadena especificada
O	Ir al principio de la línea actual
b	Desplazarse una palabra hacia atrás
w	Desplazarse una palabra hacia adelante
Ctl-B	Regresar pagina
Ctl-F	Avanzar pagina
G	Ir al final de la línea
NG	Ir a la línea n-ésima
N	Repite la última búsqueda
g!/patrón/	Ejecuta el comando g especificado en aquellas líneas que no contengan el patrón

### Comandos de edición

<<	Sangrar la línea hacia la izquierda
>>	Sangrar la línea hacia la derecha
a	Agregar texto a la posición actual del carácter
A	Agregar texto al final de la línea actual
d\$	Borrar desde la posición del cursor hasta el final de la línea
dd	Borrar la línea actual
dw	Borrar la siguiente palabra después del cursor
L	Insertar texto antes del cursor
J	Juntar la línea actual con la siguiente
Yy	Copiar la línea actual al búfer
P	Insertar texto almacenado en el búfer
U	Deshacer el último cambio
x	Borrar el carácter actual
r archivo	Inserta el texto del archivo especificado dentro del que se esta editando y a partir de la línea donde se encuentra el cursor
s/cad1/cad2/	reemplaza el texto de la cadena 1 por el texto de la cadena 2, por omisión se mueve a la siguiente línea

### Comandos especiales

:!<comando>	Ejecuta el archivo shell especificado con comando
:n,m/x/y/g	Reemplaza por y cada primera ocurrencia de x desde la línea n hasta la m
:s/x/y/	Reemplaza por y la primera ocurrencia de x en la línea actual
:q	Salir del editor vi
:q!	Salir del editor vi sin guardar cambios
:w	Guardar el archivo actual en disco
:w!	Guardar el archivo actual en disco, sin importar la protección contra escritura
:w archivo	Guardar el buffer de edición en el archivo especificado
:w! archivo	Guardar el buffer de edición en el archivo especificado sin importar la protección contra escritura
:set nu	Mostrar los números de línea
:set nonu	Desactivar la muestra de números de línea

Vi trabaja en dos modos principales: modo comandos y modo inserción, este último añadirá todos los caracteres tecleados al archivo, comenzando por el inmediato a la derecha, sin embargo, para pasar al modo comandos desde el modo inserción es necesario teclear ESC y en sentido contrario, para ir del modo comandos al modo inserción, teclear 'i'.

Para desplazarte a una línea en particular conviene que visualices el número de todas las líneas y después te desplaces, por ejemplo, para ir a la línea 45, borrar el carácter actual y después guardar el archivo:

```
:set nu
:45G
x
:w
```

aunque si el siguiente paso es salir del editor vi se puede teclear ":q" o mejor aún, combinar los comandos de manera que sea mas sencillo escribir una sola vez todos los comandos a ejecutar, en este caso ":wq" que guarda el archivo en disco y después sale del editor vi.

El *comando set* de ex permite habilitar o deshabilitar opciones que controlan el modo de funcionamiento de vi, para ver las opciones que se tienen actualmente habilitadas, basta con ejecutar ':set'. Algunos de los comandos set requieren de algún tipo de parámetro, generalmente son los que se dedican a definir características sobre el entorno de vi, a continuación se muestran algunos comandos:

autoindent	Hace que vi, en el modo de inserción, indente de forma automática cada línea con la precedente.
ignorecase	Hace que vi no tenga en cuenta mayúsculas/minúsculas en las búsquedas
number	Hace que vi muestre los números de línea
shiftwidth=width	Define el ancho de las columnas cuando esta habilitado el autoindentado y define la operación de los comandos de shift
showmatch	Hace que vi muestre parpadeando brevemente el paréntesis '(' o el corchete '[' cuando el cursor pase por

	un ')' o un ']', respectivamente.
tabstop=spaces	Define el número de espacios de indentación asociados en el carácter de TABULADOR.
wrapscan	Hace que vi busque desde el principio del archivo una vez que se haya llegado al final del mismo
wrapmargin=width	i autoindenta líneas que excedan de la anchura especificada.

### Caracteres con significado especial

^	Representa el principio de línea
\$	Representa el final de la línea.
%	Representa todas las líneas
.	Especifica un único carácter
*	Especifica cualquier expresión
\x	Especifica qu ex se entienda como un carácter ordinario, siendo x, cualquier valor tecleado
[lista]	Especifica una lista, pudiendo ser cualquiera de ellos, ej. [123456789]
[rango]	Especifica un rango de caracteres, ej. [0-9], se pueden especificar múltiples rangos como [0-9a-z]
\<patron	Identifica el patrón cuando este ocurre al principio de una palabra
\>patron	Identifica el patrón cuando este ocurre al final de una palabra

A continuación se ilustran algunos ejemplos, tratando de explicar algunos comandos mencionados anteriormente.

Para arrancar vi, ejecuta el comando:

vi archivo

donde archivo es el nombre del archivo que se desea editar, si se quiere saltar a una línea en específico, basta con incluir lo siguiente:

vi +num archivo

donde num es el número de línea en la que se desea iniciar, pero tal vez sea mas útil ir a la primer línea con una palabra en particular, por lo que escribiremos:

vi +/palabra archivo

donde palabra es la palabra que deseamos encontrar.

Para borrar una única línea, ejecuta:

:xd

donde x representa el número de línea a borrar, pero tal vez no sea suficiente y desees borrar todo un bloque, para ello, ejecuta:

:x,yd

donde 'x' es la primer línea y la 'y' es la última línea que deseas borrar

Para copiar un texto que se encuentra en un bloque de líneas, basta con ejecutar:

`:x,ymz`

donde 'x' y 'y' son los limitadores del bloque de líneas que se va a copiar y z es la línea anterior a donde se copiará el texto. Si por el contrario, solo se desea mover el texto, se debe ejecutar:

`:x,ytz`

donde la 'x', 'y' y 'z' funcionan de la misma manera que en el comando anterior.

Si se desea remplazar el texto hola por el texto adios en todo el archivo, se ejecuta:

`:%s/hola/adios/g`

donde % representa todo el archivo y la g permite buscar varias ocurrencias en la misma línea, ya que el comando s solo encuentra una ocurrencia por línea y hace caso omiso si existen otras en la misma.

Otra aplicación de g es cuando se desea borrar todas las líneas que contengan una cadena como sigue:

`:1,$g/cadena/d`      ó      `:%g/cadena/d`

También se puede arrancar un shell de la siguiente manera:

`!:sh`