

# Usando el editor de texto *vi*

Juan José Álvarez Sánchez

Escuela Universitaria de Informática  
Universidad de Valladolid *campus* Segovia

24 de febrero de 2004

## 1. Introducción

Antes de empaparse en el uso del editor de texto *vi*, es conveniente conocer la diferencia entre un *editor de texto* y un *procesador de texto*. Un procesador de texto es un programa (paquete) informático diseñado para la redacción de documentos que contienen texto resaltado, gráficos, figuras, ecuaciones, tablas, etc. Como ejemplos de este tipo de procesadores se pueden citar MS WORD, Word Perfect y LaTeX, si bien la diferencias entre ellos son considerables.

MS WORD, por ejemplo, es un procesador de texto *WYSIWYG*, es decir, "what you see is what you get"; esto significa que lo que el usuario ve en la ventana donde está preparando el documento, es lo que obtendrá como resultado final en el papel impreso o cualquier otro medio que sirva como destino final del documento.

LaTeX, por su parte, es un procesador de texto basado en la compilación de un fichero fuente escrito en texto puro que contiene códigos y secuencias propias de un lenguaje de generación de documentos. Crear un documento con LaTeX es complicado debido a la necesidad de aprender su lenguaje y estructuras, sin embargo su potencial y calidad es muy grande, lo que ha ocasionado que sea éste el estándar en el cual se publican una gran cantidad de revistas, libros y publicaciones científicas en general; entre ellas, las más importantes como las de la IEEE, Addison-Wesley, etc.

Comparado con MS WORD, Word Perfect y LaTeX, *vi* puede parecer insignificante al ser un simple editor de texto. Sin embargo, el potencial de *vi* reside en su ubicuidad y utilidad: *vi* es un editor que está presente en todos los sistemas UNIX y Linux, y en ocasiones, cuando el sistema computacional no funciona del todo *vi* es el único editor disponible y se le puede utilizar para cambiar los ficheros de configuración del sistema a fin de restaurarlo.

*vi* no es el único ni el más simple de los editores de texto de UNIX, sin embargo, como se dijo antes, está disponible en todos los sistemas UNIX y el aprender a manejarlo le permitirá realizar tareas tales como introducir, modificar o borrar texto, buscar o sustituir cadenas, y copiar, cortar y pegar bloques de texto. Además, es configurable por el usuario.

## 2. Generalidades

*vi* funciona en dos modos: *modo inserción* y *modo instrucción*. El modo inserción es el modo normal de escritura en el cual *vi* actúa como una máquina de escribir donde todos los caracteres que se teclean se introducen como parte del documento (fichero), mientras que en el modo instrucción (command) los caracteres o teclas presionadas se interpretan como órdenes o directivas del editor que permiten borrar, copiar, salir, sustituir, buscar, mover el apuntador (cursor), etc.

En una sesión de *vi* se conmuta frecuentemente de un modo a otro y es conveniente saber en todo momento en que modo se encuentra el editor. Para entrar al modo inserción se presionan las teclas <i> o <a>; para salir del modo inserción e ir al modo instrucción se presiona la tecla <Esc>. La tecla <Esc> se puede presionar repetidamente sin que ocurra nada significativo, mientras que si se presionan repetidamente las teclas <i> o <a> solo el primer carácter actuará como una orden mientras que el resto de formarán parte del texto introducido.

El texto creado o editado por *vi* se manipula creando o modificando ficheros, respectivamente. Cuando se desea modificar un fichero (texto) existente se llama al fichero correspondiente a fin de que el editor

cargue una copia de éste en la sesión de edición. Ya sea que se esté creando o modificando un texto existente, el editor mantiene el texto que está manipulando en un área de almacenamiento llamada *memoria intermedia (buffer)*. Así, los cambios efectuados no se registran en el fichero hasta que se decida explícitamente guardar el texto, es decir, el contenido de la memoria intermedia. Se puede guardar el texto cuantas veces se desee y no es necesario salir del editor para hacerlo.

Se dice que el editor *vi* es interactivo porque interacciona con el usuario mostrándole mensajes de estado y error. La última línea de la ventana (pantalla) del editor se llama *línea de estado* y sirve para este propósito. Por otra parte, la posición del apuntador (cursor) siempre indica su situación actual en la memoria intermedia y algunas de las órdenes que se introducen afectan el carácter que se encuentra en dicha posición. Naturalmente, *vi* tiene varias órdenes para mover el apuntador (cursor) por la memoria intermedia (ventana o pantalla).

### 3. Entrando en *vi*

Para iniciar *vi* simplemente se escribe su nombre en la línea de órdenes del intérprete. Si se sabe el nombre del archivo que se quiere crear o editar se puede lanzar este programa usando como argumento el nombre de dicho fichero. Por ejemplo: **vi nombre\_del\_fichero**.

Cuando se inicia *vi* la pantalla de la terminal se borra y aparece el contenido del fichero a editar. En el caso de que se trate de un fichero nuevo, o de que no se haya dado nombre alguno, en la primera columna de todas las líneas visibles de la pantalla, exceptuando la primera, se presenta el carácter tilde `~`. Este carácter es el indicador de línea de memoria intermedia vacía.

### 4. Agregando e insertando texto

Cuando arranca *vi* el editor se inicia siempre en el modo instrucción. Para insertar texto en la posición actual del apuntador (cursor) se presiona la tecla `<i>` seguida del texto que se desea insertar. Otra forma de insertar texto es mediante la acción de la tecla `<a>`; en este caso el texto introducido después de la orden se agrega en el carácter que sigue a la posición actual del apuntador (cursor).

Tanto `<i>` como `<a>` tienen como efecto que el editor conmute del modo instrucción al modo inserción; para salir del modo inserción hay que presionar la tecla `<Esc>`. Es importante señalar que en el modo inserción todas las teclas presionadas se consideran parte del texto a introducir, es decir, en este modo de operación no se pueden manipular ninguna de las teclas de posicionamiento del apuntador (cursor), avance de página, etc. La única excepción es la tecla `<BackSpace>` o `<←>`, la cual tiene como función borrar el carácter de pantalla inmediatamente anterior a la posición del apuntador.

Una importante variación de `<i>` y `<a>` son `<Shift-i>` y `<Shift-a>`, respectivamente. Si estando en el modo instrucción se presiona la tecla `<Shift-i>` el editor se posicionará al comienzo de la línea actual para permitir la inserción de texto en ese lugar, mientras que si se presiona `<Shift-a>` el indicador (cursor) se desplazará al final de la línea actual para agregar texto a continuación.

Para introducir una nueva línea de texto debajo o encima de la línea actual se presionan las teclas `<o>` y `<Shift-o>` respectivamente. Cada una de ellas abre una nueva línea en la memoria intermedia que permitirá agregar nuevo texto en la posición correspondiente.

### 5. Movimiento y posicionamiento del apuntador (cursor)

En todos los procesos de edición de texto es necesario posicionar el indicador con el fin de insertar texto adicional, borrar, corregir errores, agregar texto al final del documento, etc. Para este fin existen en *vi* una serie de instrucciones de posicionamiento del indicador. Dependiendo del sistema en el que esté instalado *vi* se pueden utilizar para este fin las teclas de flecha del teclado `<←>` `<→>` `<↑>``<↓>`, así como las teclas `<PageUp>` y `<PageDown>`.

Sin embargo, en un sistema que no esté adecuadamente configurado para *vi*, puede ocurrir que estas teclas introduzcan caracteres extraños. Para subsanar esto existen otras teclas diferentes a las flechas que le permiten desplazar el apuntador por el texto que conviene conocer. Debido a que cuando se desarrollo *vi* muchos teclados no disponían de las teclas de flecha se usaban y se usan las teclas `<h>`, `<j>`, `<k>` y `<l>`; `<h>` y `<l>` se usan para desplazar el apuntador un carácter hacia la izquierda y

derecha, respectivamente, mientras que <j> y <k> se usan para mover el apuntador una línea abajo y arriba, respectivamente.

Otras alternativas para mover el apuntador son:

- <Barra espaciadora> para mover el cursor a la derecha.
- <Intro>, <Enter> o <+> para ir al comienzo de la siguiente línea.
- <-> para ir al comienzo de la línea anterior.
- <0> para ir al principio de la línea actual.
- <\$> para ir al final de la línea actual.

Existen otras órdenes en *vi* que permiten mover el apuntador en relación a las palabras de una línea. Una palabra es para *vi* una secuencia de caracteres separada de otras por espacios o símbolos tales como *./*, */?/*, */,/* y */-/*. Estas instrucciones son:

**w** Avanza una palabra.

**b** Va al inicio de la palabra actual

**e** Va al final de la palabra actual

Tanto con las tres órdenes anteriores como con las instrucciones <h>, <j>, <k> y <l> es posible anteponer un número al carácter de tal manera que por ejemplo <3><w> ocasionará que el indicador (cursor) avance 3 palabras a la derecha.

Existen otras órdenes que permiten realizar grandes desplazamientos del apuntador sobre el texto; en cada uno de estos casos aparece el cursor al principio de la línea destino. Por ejemplo:

- <Shift-h> posiciona el apuntador en la primera línea de la ventana actual de texto.
- <Shift-m> coloca el apuntador en la línea que se encuentra en medio de la ventana actual de texto.
- <Shift-l> coloca el apuntador en la última línea de la ventana actual de texto.

Si se desea desplazarse por el fichero de ventana en ventana se pueden utilizar las teclas <Ctrl-f> para ir una ventana adelante y <Ctrl-b> para ir una ventana atrás. Si se desea ir rápidamente a la última línea del fichero (memoria intermedia) se puede presionar la combinación de teclas <Shift-g> y, en general, para ir a cualquier línea de la memoria intermedia se puede introducir el número de línea antes de presionar <Shift-g>. Por ejemplo, para ir directamente a la primera línea del fichero se presiona <1><Shift-g>.

## 6. Búsqueda de texto

*vi*, como casi todos los editores de texto, dispone de órdenes para buscar cadenas de texto. La búsqueda se puede realizar hacia adelante o hacia atrás desde la posición actual del apuntador (cursor) y se puede continuar buscando incluso cuando ya se ha llegado al inicio o al final del texto. Las órdenes para la búsqueda de texto en *vi* son

**/cadena** Busca la cadena hacia adelante con respecto a la posición actual.

**?cadena** Busca la cadena hacia atrás con respecto a la posición actual.

**<n>** Vuelve a buscar en la misma dirección de la última búsqueda.

**<Shift-n>** Vuelve a buscar en la dirección opuesta a la última búsqueda.

Cuando se utiliza una de estas instrucciones de búsqueda, la orden aparece escrita en la línea de estado. Si la cadena se encuentra en la memoria intermedia el indicador se posiciona al comienzo de dicha cadena, en caso contrario el editor muestra en la línea de estado el mensaje **Pattern not found**.

## 7. Cambio y sustitución de texto

Otra tarea muy frecuente en la edición es el cambio o sustitución de una cadena de texto por otra. Con *vi* se puede sustituir un carácter, una secuencia de caracteres, un segmento de línea o una línea completa. Las órdenes de cambio y sustitución de texto son:

<r> Sustituye solo un carácter; aquél en el que está posicionado el cursor.

<Shift-R> Sustituye tantos caracteres como se introduzcan a continuación.

<c><w> y <c><e> Sustituye los caracteres de la palabra actual, desde la posición del cursor hasta el final de la palabra.

<c><b> Cambia la palabra actual, desde el comienzo de la misma hasta el carácter anterior a la posición del apuntador (cursor).

<c><\$> y <Shift-c> Cambia una línea desde la posición del apuntador (cursor) hasta el final de la línea.

<c><c> Cambia la línea entera.

En todos los casos, excepto con la orden de reemplazo <r>, es necesario presionar <Esc> para volver al modo instrucción una vez completada la sustitución.

## 8. Supresión de texto

En todos los procesos de edición de texto se hace necesario suprimir texto con el fin de corregir errores o realizar correcciones en general. Con *vi* se puede suprimir un carácter, una palabra, un número consecutivo de palabras, una línea entera o un cierto número de ellas. Para suprimir texto es necesario posicionar el apuntador en el lugar donde se quiere realizar la operación y usar una de las siguientes órdenes en modo instrucción:

<x> Suprime el carácter que se encuentra en la posición actual del apuntador.

<d><w> Borra los caracteres desde la posición actual del indicador hasta el principio de la siguiente palabra.

<d><\$> Suprime los caracteres desde la posición actual del cursor hasta el final de la línea.

<Shift-d> Al igual que el anterior, borra el resto de la línea.

<d><d> Borra la línea actual por completo independientemente de la posición del cursor.

Algunas de las órdenes de supresión anteriores se pueden utilizar con varios caracteres, palabras o líneas. Por ejemplo:

- <4><x> borra cuatro caracteres.
- <3><d><w> borra tres palabras consecutivas.
- <8><d><d> borra ocho líneas consecutivas.

## 9. Copiar, cortar y pegar texto

Con *vi* no solo se puede seleccionar texto para pegarlo en otra parte del documento, sino que además, todos los caracteres, palabras o líneas que se suprimen se almacenan en una memoria genérica para que luego puedan también colocarse en otra parte del documento. Las operaciones de pegado de texto se realizan mediante las órdenes

<p> para pegar el texto a la derecha del apuntador (cursor).

<Shift-p> para pegar el texto a la izquierda del apuntador.

Las órdenes explícitas para copiar texto a la memoria genérica son las siguientes:

<y><w> Copia desde la posición actual del indicador hasta el delimitador final de la palabra.

<y><\$> y <Shift-y> Copian desde la posición actual del curso hasta el final de la línea.

<y><y> Copia la línea actual de principio a fin.

A algunas de las órdenes anteriores se puede anteponer un número para copiar una cierta cantidad de palabras o líneas.

## 10. Deshacer una orden

Con *vi* es posible deshacer la acción más reciente, siempre que el fichero no haya sido guardado en disco. Para realizar esto cerciórese de estar en el modo instrucción (presionando <Esc>) y presione la tecla <u>. Si se presiona <u> nuevamente, la corrección se deshace. Hay que recordar que en *vi* solo es posible deshacer la orden más reciente.

## 11. Repetición de órdenes

*vi* guarda la última orden que se utilizó en el editor con el fin de reutilizarla. Para repetir la última orden ejecutada cerciórese de estar en el modo instrucción y presione el carácter del punto <.>

## 12. Escritura de ficheros a disco

Las órdenes para guardar el contenido de la memoria intermedia, al igual que el resto de las órdenes de *vi*, se introducen en el modo instrucción. Existen en este editor de texto tres formas de guardar el texto en un fichero sin salir del editor de texto.

**:w** Esta opción es útil cuando se ha iniciado la sesión de *vi* usando un nombre de fichero; es decir, se inició la sesión escribiendo **vi nombre\_de\_fichero**. En este caso el contenido de la memoria intermedia se almacena en un fichero que tiene el nombre del fichero de entrada.

**:w nombre\_de\_fichero** Esta opción se usa cuando se inició la sesión de *vi* sin proporcionar un fichero de entrada y se hace necesario dar un nombre de fichero de almacenamiento en disco. Esta opción también se puede usar cuando se ha iniciado *vi* dando un fichero de entrada pero se desea guardar el texto en un fichero con otro nombre.

**:w! nombre\_de\_fichero\_existente** En algunas ocasiones, al usar la opción **:w nombre\_de\_fichero** del inciso anterior puede ocurrir que ya exista un fichero con el nombre que se desea utilizar para el fichero que se está editando, en cuyo caso el editor despliega en la línea de estado un mensaje:

**File exist - use "w! nombre\_de\_fichero to overwrite.**

Este es un mensaje preventivo para que no se sobrescriba el fichero existente, sin embargo, si realmente es esto lo que se desea hacer, hay que usar la orden **:w! nombre\_de\_fichero\_existente**

## 13. Salida de *vi*

Se puede salir del editor de texto *vi* de varias formas, recordando que para hacerlo hay que estar en el modo instrucción habiendo presionado la tecla <Esc>. Las varias formas de salir de *vi* son:

**:q** Esta opción se usa cuando no se han hecho cambios en el fichero editado o bien cuando los cambios realizados han sido almacenados en un fichero. En caso contrario el editor mostrará un mensaje preventivo y nos impedirá salir.

**:q!** Esta opción se usa cuando se desea abandonar el editor aún cuando se hayan hecho cambios en la memoria intermedia y estos cambios no se hayan registrado en un fichero. En este caso se pierden todas las modificaciones realizadas desde la última vez que se almacenó el texto en un fichero.

**:wq , :x y ZZ** En estos casos se almacena el contenido de la memoria intermedia en el fichero previamente indicado y se termina la sesión del editor.

## 14. Definición del entorno de *vi*

El editor de texto *vi* dispone de una gran cantidad de opciones de configuración que el usuario puede decidir usar; las opciones de configuración de *vi* son de tipo booleano, numérico o alfanumérico. Para activar una opción booleana se tecléa en la línea de estado del editor **:set opción**, mientras que para desactivarla se tecléa **:set noopción**. Para establecer una opción de tipo numérico se tecléa **:set opción=número**, mientras que para establecer una opción de tipo alfanumérica se tecléa **:set opción=cadena**.

Algunas de las opciones de configuración de *vi* son:

**showmode o smd** Cuando se activa se presenta un mensaje en la línea de estado que indica en que modo se encuentra el editor en cada momento.

**number o nu** Cuando se activa se numeran las líneas de texto del documento.

**autoindent o ai** Cuando se activa las nuevas líneas de texto que se generan se alinean con la anterior.

**showmatch o sm** Activando este parámetro se resalta el paréntesis abierto que corresponde al paréntesis cerrado que se introduce en el texto.

**wrapmargin o wm** Con esta opción se puede fijar el número de caracteres desde el margen derecho con los cuales el editor rompe la línea actual introduciendo un retorno de carro.

## 15. El fichero **.exrc**

Generalmente, siempre que se desea usar el editor *vi* es conveniente que éste tenga la configuración deseada por el usuario. Si para llegar a esa configuración el usuario necesita cambiar o establecer una gran cantidad de parámetros, el hacerlo en cada sesión de *vi* puede ser tedioso debido a que al salir del editor se pierde la configuración establecida.

Una posibilidad para evitar el trabajo de configurar *vi* para cada sesión es mediante el uso de un fichero de configuración. El fichero particular que sirve para configurar *vi* se llama **.exrc** y debe estar localizado en el directorio de inicio del usuario (v.g., \$HOME). El contenido de este fichero son órdenes del tipo **set** como las que se introducen en la línea de estado, con la particularidad de que no deben ir precedidas de los dos puntos **:/**.

## 16. Ejecución de órdenes del intérprete (shell) de UNIX desde *vi*

Es posible ejecutar órdenes del intérprete (shell) de UNIX desde la línea de estado de *vi* sin perder el texto que se está editando. Para ello se usa la orden **!orden**; en este caso, el símbolo **!** le indica al editor que lo que sigue es una orden del intérprete (shell) de UNIX.

Más aún, el texto enviado a la salida estándar como consecuencia de la orden se puede integrar al documento que se está editando mediante la sintaxis **:r!orden**; el texto se insertaría en la línea que sigue a la posición actual del apuntador (cursor). La sintaxis **:r fichero** se puede utilizar para integrar el contenido de "fichero" en la línea siguiente a la posición actual del apuntador (cursor) en el documento que se está editando.