

EDITOR VI: FUNDAMENTOS DE USO Y REFERENCIA BÁSICA DE COMANDOS.

Alejandra Granja Del Río (PAYUD)

alejandra.granja@uva.es

Iván Cabria Álvaro (PTUN)

ivan.cabria@uva.es

Departamento de Física Teórica, Atómica y Óptica

Facultad de Ciencias
Universidad de Valladolid
Paseo Belén 7
Campus Miguel Delibes
47011 Valladolid

Última actualización: 18 de diciembre de 2025



Este trabajo se distribuye bajo la licencia **Creative Commons Attribution 4.0 International (CC BY-NC-SA 4.0)**. <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Copyright © 2025 Alejandra Granja Del Río y Iván Cabria Álvaro

Este material se distribuye bajo la licencia Creative Commons Attribution 4.0 International (CC BY-NC-SA 4.0).



<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Editor vi: fundamentos de uso y referencia básica de comandos.

Antes de utilizar el editor **vi**, es necesario comprender el entorno desde el cual se ejecuta. En los sistemas UNIX y Linux, la interacción con el sistema se realiza habitualmente a través de una *consola* o *terminal*, que muestra un mensaje de espera denominado *prompt*. Este prompt indica que el sistema está listo para recibir órdenes escritas por el usuario.

El aspecto del *prompt* puede variar según el sistema y la configuración, pero suele terminar con símbolos como \$ o #. El símbolo \$ indica que se está trabajando como usuario normal, mientras que # señala que se tienen privilegios de administrador (ver Fig. 1). Tras el *prompt*, el usuario debe escribir el comando que desea ejecutar y pulsar la tecla **ENTER** para confirmarlo.



Figura 1: Ejemplo de un *prompt* de consola con \$ desde el que se ejecuta el editor vi.

Qué es vi

El editor **vi** es un programa destinado únicamente a la edición de texto plano. Carga el documento completo en memoria y ofrece un entorno muy básico, característico de los sistemas UNIX, donde se encuentra presente desde sus primeras versiones. No es un procesador de textos y, por tanto, no incorpora funciones avanzadas como estilos tipográficos, justificación automática o inserción de elementos gráficos.

Aunque muchos usuarios sin formación técnica lo consideran complicado, sobre todo quienes proceden de sistemas como DOS, **vi** resulta una herramienta muy eficaz. Su diseño permite utilizarlo en una amplia variedad

de terminales y ofrece gran agilidad para modificar texto: mover, copiar, borrar o añadir contenido con rapidez.

Para iniciar el editor **vi**, basta con escribir su nombre en el prompt, seguido opcionalmente del nombre del archivo que se desea editar. A partir de ese momento, el control pasa del sistema al editor, que se ejecuta dentro de la propia consola (ver Fig. 2).

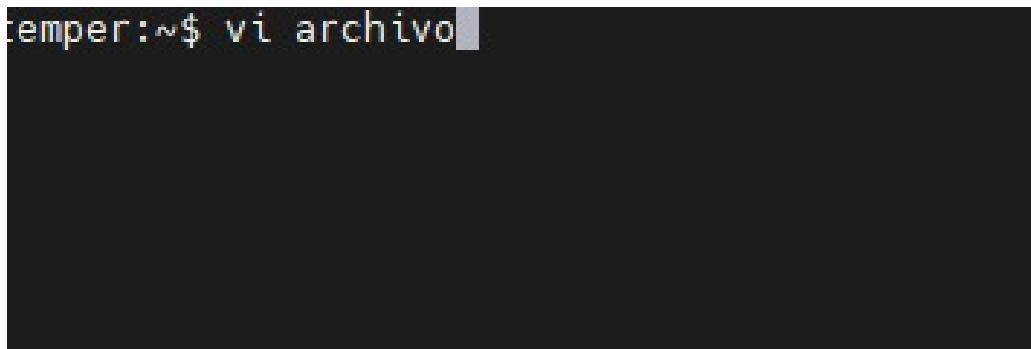


Figura 2: Cómo ejecutar **vi** para abrir un archivo desde el *prompt*. En este caso se abre el archivo de nombre “archivo”.

Inicio **vi**

Al ejecutar **vi**, se abre una ventana de edición en la pantalla del terminal. El cursor indica la posición actual, las líneas vacías se muestran con el símbolo `~` y, en la parte inferior, aparece una barra de estado con el nombre del archivo y la cantidad de caracteres que contiene (ver Fig 3).

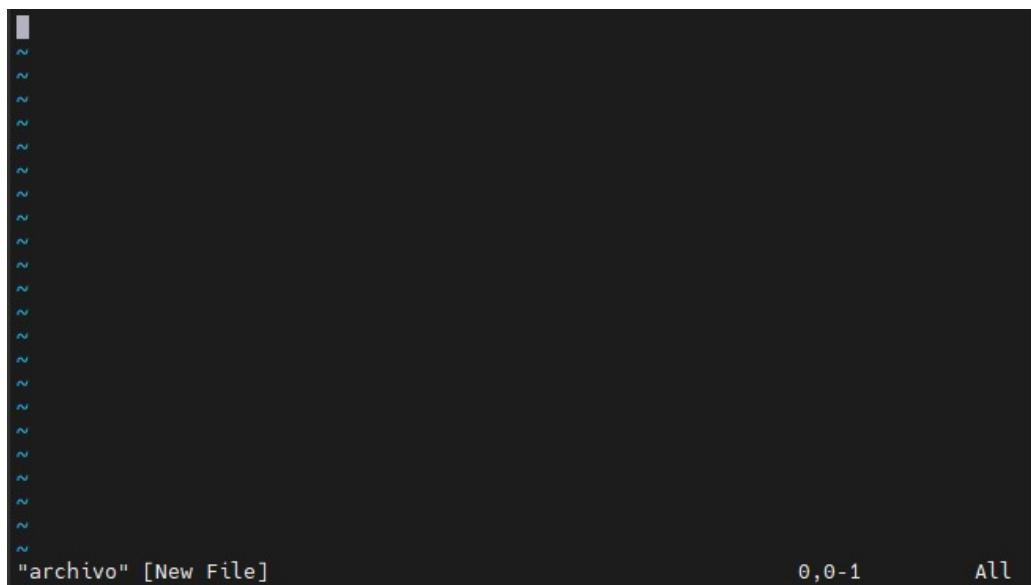


Figura 3: Ejemplo de la apertura de un archivo con **vi**.

Si se indica un nombre de archivo al iniciar **vi**, el programa mostrará su contenido si existe. En caso contrario, creará automáticamente un nuevo archivo, con ese nombre, en la ubicación indicada.

Formas de iniciar **vi**:

1. **vi**: abre el editor sin ningún archivo.
2. **vi archivo**: permite abrir un archivo existente o crear uno nuevo.
3. **vi archivo1 archivo2**: carga varios archivos en la misma sesión.
 - **:next o :n:** pasa al archivo siguiente.
 - **:prev o :N:** vuelve al anterior.
4. **vi +n archivo**: inicia la edición en la línea n.
5. **vi +/patrón archivo**: sitúa el cursor en la primera coincidencia del patrón.

Modos de operación en vi

El editor **vi** funciona principalmente en tres estados: modo **comando**, modo **inserción** y modo **Línea (ex)**.

1. **Modo inserción**: Permite escribir o añadir texto en el documento. Para entrar desde el modo comando, se usan teclas como **i**, **I**, **a**, **A**, **o**, **O**. Todo lo que se escriba aparece directamente en la pantalla. Para regresar al modo comando, se pulsa **ESC**.
2. **Modo Línea (ex)**: Permite escribir instrucciones en la franja inferior del terminal. Se activa comenzando la orden con “**:**” desde el modo comando y ofrece funciones adicionales como renombrar, guardar, mover o gestionar archivos. Se vuelve al modo **comando** al pulsar **ENTER** (para ejecutar) o **ESC** (para cancelar) (ver Tabla 1 y Figura 4).
3. **Modo comando**: Es el entorno inicial al iniciar **vi**, en el que el teclado no se utiliza para escribir texto, sino para ejecutar instrucciones. Cada pulsación activa una orden concreta que permite desplazarse por el documento, eliminar o duplicar contenido, buscar información, guardar cambios o finalizar la edición, ofreciendo un control directo y preciso del archivo. Es recomendable permanecer en este modo el mayor tiempo posible, ya que reduce errores al modificar el texto. Para acceder a él desde el modo inserción, se utiliza la tecla **ESC**.

En la práctica, el trabajo en **vi** consiste en pasar de un modo a otro constantemente: se navega y ejecutan comandos en el modo de **comando**, se redacta en el modo de **inserción** y, cuando se necesitan acciones más específicas, se recurre al modo de **Línea (ex)**. Esta organización hace que **vi** sea rápido, fiable y adecuado incluso para terminales muy sencillos.

Tabla 1:

Comando	Función que activa
:	Muestra la barra de órdenes del editor



Figura 4: Ejemplo para mostrar la barra de órdenes del editor vi con “:”.

Comandos del modo inserción

Los siguientes atajos permiten comenzar la escritura en distintas posiciones del documento (ver Tabla 2):

Tabla 2:

Comando	Descripción de la acción
i	Inicia la entrada de texto justo antes del punto actual
I	Comienza a escribir al principio de la línea
a	Añade texto inmediatamente después del cursor
A	Empieza la inserción al final de la línea activa
o	Abre una nueva línea inferior y coloca en modo inserción
O	Genera una nueva línea superior e inicia la escritura

Comandos del modo línea

A continuación se presentan las órdenes más habituales disponibles en este modo, organizadas según el tipo de acción que realizan.

Guardar, salir y variantes

Cuando se trabaja con **vi**, las modificaciones no se realizan directamente sobre el archivo original. El editor carga una copia en un área temporal llamada «buffer», y todos los cambios se aplican allí. El archivo real solo se actualiza cuando el usuario decide guardar. Esto permite descartar toda la edición si no se desea conservar nada, pero también implica que, si no se guarda con frecuencia, los cambios podrían perderse en caso de fallo del sistema. Por ello, es aconsejable grabar a menudo. Estas órdenes controlan cuándo se guardan los cambios y cómo abandonar el editor. Recordar que con “**:**” se entra en modo **Línea** (ver Tabla 3):

Tabla 3:

Comando	Resultado
:q	Cierra el editor si nada fue modificado
:q!	Sale ignorando todos los cambios
:w	Guarda el contenido actual
:w archivo	Guarda con un nombre distinto
:x ó :wq ó ZZ	Guarda y finaliza la edición (equivalentes)
:wq!	guardar y salir del editor forzoso

Localización de texto

Tabla 4:

Comando	Acción que realiza
/texto	Busca la cadena hacia abajo en el fichero
?texto	Busca la cadena hacia arriba
n	Repite la búsqueda en la misma dirección
N	Repite la búsqueda en dirección inversa

Sustitución de cadenas

En **vi** se pueden reemplazar palabras o fragmentos de texto en distintas zonas del archivo (ver Tabla 5).

Tabla 5:

Comando	Qué modifica
:s/viejo/nuevo/	Cambia solo la primera coincidencia de la línea actual
:s/viejo/nuevo/g	Cambia todas las coincidencias de esa misma línea
:%s/viejo/nuevo/g	Reemplaza en todo el fichero
:1,5s/viejo/nuevo/g	Sustituye únicamente entre las líneas 1 y 5

Copiar, mover y numeración

Estas órdenes permiten reorganizar líneas y ajustar cómo se muestran en pantalla (ver Tabla 6).

Tabla 6:

Comando	Efecto
:1,2 co 3	Copia las líneas 1–2 justo después de la línea 3
:4,5 m 6	Traslada las líneas 4–5 a continuación de la línea 6
:set nu	Añade numeración visible a cada línea del documento
:set nonu	Oculta la numeración de líneas para una vista más limpia

Opciones de configuración para optimizar la edición en vi

Tabla 7:

Ajuste / Comando	Descripción resumida
:set ai	Habilita la indentación automática para facilitar la escritura estructurada
:set sw=#	Establece el ancho del desplazamiento al sangrar (reemplazar # por un valor numérico)
:set sm	Resalta el símbolo de apertura emparejado al introducir uno de cierre
:set ic	Desactiva la sensibilidad a mayúsculas/minúsculas en búsquedas
:set noic	Activa la distinción entre mayúsculas y minúsculas en las búsquedas

Comandos del modo comando

En este apartado se recopilan los comandos más usados del modo comando, clasificados por funcionalidad.

Desplazamiento del cursor

Tabla 8:

Comando	Función resumida
h / j / k / l	Mover el punto activo (izquierda, abajo, arriba, derecha)
0	Saltar al inicio de la línea actual
\$	Saltar al final de la línea actual
w / W	Avanzar a la siguiente palabra
b / B	Retroceder a la palabra previa
e	Ir al final de la palabra donde estás
G	Ir directamente a la última línea del archivo
nG	Ir a la línea número n
H / M / L	Ir al comienzo, centro o final de la pantalla visible
/c / ?c	Buscar el carácter c hacia delante / hacia atrás
f c / F c	Saltar a la próxima aparición del carácter c (adelante / atrás)
t c / T c	Igual que f/F pero quedándose antes del carácter
Ctrl+f / Ctrl+b	Pasar una pantalla completa adelante / atrás
Ctrl+d / Ctrl+u	Media pantalla hacia abajo / arriba

Eliminación de texto

Tabla 9:

Comando	Acción realizada
x	Suprime el símbolo bajo el cursor
X	Borra el carácter situado a la izquierda
dw	Elimina desde la posición actual hasta el final de la palabra
dd	Quita por completo la línea actual
ndd	Borra n líneas seguidas
d0	Borra desde el cursor hasta el comienzo de la línea
d\$	Borra desde el cursor hasta el final de la línea
d) / d(Suprime hasta el fin / inicio de la frase
d} / d{	Elimina hasta el fin / inicio del párrafo
ndw / ndb	Borra varias palabras hacia adelante / atrás
:1,10d	Elimina líneas del 1 al 10
:10,\$d	Borra desde la línea 10 hasta el final del documento

Copiar, cortar y pegar

Tabla 10:

Comando	Descripción
yy / Y	Copia la línea actual en el portapapeles interno
nyy	Copia n líneas seguidas
yw	Copia la palabra donde se encuentra el cursor
p	Pega el contenido después del cursor o línea actual
P	Pega el contenido antes del cursor o línea actual
ayy	Copia la línea en el registro a
add	Corta la línea en el registro a
ap / aP	Pega desde el registro a (después / antes)

Deshacer, repetir y recuperación

Tabla 11:

Comando	Qué hace
u	Revierte la modificación más reciente
U	Recupera la línea tal como estaba antes de modificarla
.	Repite el comando ejecutado anteriormente
:e!	Restaura el archivo desde el último guardado
“1p – ”9p	Recupera textos borrados almacenados en registros numerados