

Atajos y comandos avanzados de Vim

Edison Achalma

Escuela Profesional de Economía, Universidad Nacional de San Cristóbal de Huamanga

Resumen

Este abstract será actualizado una vez que se complete el contenido final del artículo.

Palabras Claves: keyword1, keyword2

Tabla de contenidos

Introduction	4
1 Introducción a Vim	4
1.1 ¿Qué es Vim?	4
1.2 Ventajas de utilizar Vim como editor de texto	4
1.3 Configuración inicial de Vim	5
2 Modos de Vim	5
2.1 Modo normal	5
2.2 Modo de inserción	5
2.3 Modo de comando	6
3 Guardar documentos en Vim	6
4 Atajos de navegación y movimiento	6
4.1 Movimiento básico con las teclas h, j, k, l	6
4.2 Saltos rápidos en el archivo con gg y G	7
4.3 Navegación por palabras con w, b, e	7
4.4 Desplazamiento por páginas y ventanas	7
4.5 Otros movimientos del cursor	7
4.6 Navegación por el documento	8

Edison Achalma  <https://orcid.org/0000-0001-6996-3364>

El autor no tiene conflictos de interés que revelar. Los roles de autor se clasificaron utilizando la taxonomía de roles de colaborador (CRediT; <https://credit.niso.org/>) de la siguiente manera: Edison Achalma: conceptualización, redacción

La correspondencia relativa a este artículo debe dirigirse a Edison Achalma, Email: elmer.achalma.09@unsch.edu.pe

5	Atajos de edición y manipulación de texto	8
5.1	Inserción y edición de texto	8
5.2	Inserción de texto repetitivo o secuencial	9
5.3	Eliminación de caracteres, palabras y líneas	9
5.4	Copiar y pegar texto	9
5.5	Mover una línea de posición	10
6	Selección de texto y acciones en modo visual	10
6.1	Modo Visual	10
6.2	Selección de Palabras	10
6.3	Selección de Bloques	11
6.4	Otras Acciones	11
7	Crear marcas en un fichero	12
7.1	Crear Marcas	12
7.2	Trabajar con Marcas	12
7.3	Eliminar Marcas	12
8	Aplicación de sangrías a un texto	12
8.1	Aplicar Sangrías	12
8.2	Deshacer Sangría	12
8.3	Pegar y Ajustar a la Sangría Actual	13
8.4	Aplicar Sangría en Modo Visual	13
8.5	Aplicar Sangría en Modo Insertar	13
9	Deshacer y rehacer cambios	13
9.1	Deshacer Cambios	13
9.2	Restaurar Cambio	13
9.3	Rehacer Cambios	13
9.4	Deshacer y Rehacer por Tiempo	14
10	Eliminar caracteres y líneas	14
10.1	Eliminar Caracteres	14
10.2	Eliminar Palabras	14
10.3	Eliminar Líneas	14
10.4	Eliminar por Patrón	15
11	Búsqueda y reemplazo	15
11.1	Búsqueda de texto o frase	15
11.2	Navegación entre coincidencias de búsqueda	16
11.3	Reemplazo de texto	16
11.4	Reemplazo de caracteres	16
11.5	Búsqueda y reemplazo de palabras	17
11.6	Búsqueda y reemplazo en rangos específicos	17
11.7	Transformación de texto	17
12	Especificando rangos	18
12.1	Símbolos para especificar rangos	18
12.2	Especificación de rangos mediante marcas y números de línea	18

13 Juntar 2 líneas	18
14 Atajos de teclado para realizar pliegues manuales	18
15 Registros o portapapeles	19
15.1 Mostrar y gestionar los registros	19
15.2 Registros numéricos	19
15.3 Registros de lectura	20
15.4 Registros especiales	20
16 ¿Cómo usar la terminal de Linux dentro de Vim?	21
17 Encadenar comandos en Vim**	21
18 Trabajo con múltiples archivos y ventanas	21
18.1 Apertura de archivos	21
18.2 Cierre de archivos	22
18.3 Trabajar con pestañas	22
18.4 División de la ventana en paneles y cambio entre ventanas y paneles	23
19 Personalización y configuración de Vim	24
19.1 Personalización del archivo .vimrc	24
19.2 Uso de plugins y extensiones	24
19.3 Configuración de atajos de teclado personalizados	24
20 Consejos y trucos avanzados	25
20.1 Uso de marcadores	25
20.2 Autocompletado y omnicomplete	25
20.3 Trabajo con macros	25
20.4 Uso de scripts y automatización	25
21 Recursos adicionales	25
21.1 Referencias y documentación oficial de Vim	25
21.2 Comunidades y foros de usuarios de Vim	26
21.3 Libros y tutoriales recomendados sobre Vim	26
22 Publicaciones Similares	26

Atajos y comandos avanzados de Vim

1 Introducción a Vim

¿Estás buscando mejorar tu fluidez y productividad al editar texto en Vim? ¿Quieres dominar los atajos de teclado y comandos que te permitirán aprovechar al máximo este potente editor de texto? ¿Has llegado al lugar indicado!

En el mundo de la edición de texto, Vim destaca como una herramienta poderosa y altamente personalizable. Sin embargo, para muchos usuarios nuevos, su enfoque basado en modos y su amplio conjunto de comandos pueden resultar abrumadores al principio. Pero no te preocupes, estamos aquí para ayudarte a superar esa barrera inicial y llevar tus habilidades de Vim al siguiente nivel.

En este blog, exploraremos a fondo los atajos de teclado y comandos esenciales para usar Vim de manera eficiente. Te proporcionaremos una guía completa que abarcará desde los conceptos básicos hasta técnicas avanzadas, permitiéndote aprovechar al máximo este editor de texto icónico.

Descubrirás cómo navegar rápidamente por tus archivos, editar y manipular texto con fluidez, realizar búsquedas y reemplazos de manera eficiente, trabajar con múltiples archivos y ventanas, personalizar Vim según tus preferencias y mucho más. Además, te proporcionaremos consejos y trucos prácticos para optimizar tu flujo de trabajo y ahorrar tiempo en tus tareas diarias de edición de texto.

No importa si eres un principiante curioso o un usuario experimentado en busca de nuevas técnicas, mi objetivo es ayudarte a desbloquear todo el potencial de Vim y convertirte en un maestro de la edición eficiente.

¡Prepárate para desafiar tus límites, perfeccionar tus habilidades y descubrir un nuevo nivel de productividad con Vim! Sigue leyendo y comienza tu viaje hacia la maestría de Vim con nuestros atajos de teclado y comandos indispensables.

¡La eficiencia está a solo unos atajos de distancia!

1.1 ¿Qué es Vim?

Antes de sumergirnos en los fantásticos atajos de teclado y comandos de Vim, es importante comprender qué es exactamente Vim y por qué es tan popular entre los usuarios de edición de texto.

Vim, que significa “Vi Improved” (Vi mejorado), es un editor de texto altamente configurable y poderoso. Se basa en el antiguo editor de línea de comandos llamado Vi, que ha sido una herramienta estándar en los sistemas Unix durante décadas.

1.2 Ventajas de utilizar Vim como editor de texto

La principal ventaja de utilizar Vim es su enfoque en la eficiencia y la productividad. Con los atajos de teclado y comandos adecuados, puedes editar texto de forma más rápida y eficiente que nunca. No más tediosos movimientos del ratón o desplazamientos interminables. Con Vim, puedes mantener tus manos en el teclado y volar a través del texto como un verdadero experto.

Además de su enfoque en la eficiencia, Vim ofrece una serie de características que lo hacen destacar entre otros editores de texto. Algunas de las ventajas más notables incluyen:

1. **Modos de funcionamiento:** Vim tiene modos distintos, como el modo normal, el modo de inserción y el modo de comando. Cada modo tiene un propósito específico, lo que te permite realizar diversas acciones con facilidad y fluidez.

2. **Personalización y extensibilidad:** Vim es altamente personalizable. Puedes ajustar su apariencia, configurar atajos de teclado personalizados y aprovechar su amplia gama de complementos y extensiones para adaptarlo a tus preferencias y necesidades.
3. **Potentes capacidades de búsqueda y reemplazo:** Con Vim, puedes buscar y reemplazar texto de manera rápida y precisa. Sus comandos de búsqueda te permiten encontrar palabras o patrones específicos en todo el archivo o incluso en múltiples archivos a la vez.
4. **Edición en múltiples archivos:** Vim facilita el trabajo con múltiples archivos al mismo tiempo. Puedes abrir y editar varios archivos en diferentes ventanas o pestañas, lo que te permite alternar rápidamente entre ellos y mantener tu flujo de trabajo organizado.

1.3 Configuración inicial de Vim

Antes de sumergirnos en los atajos y comandos de Vim, es importante realizar una configuración inicial para adaptarlo a tus preferencias. Aquí hay algunos pasos clave para empezar:

1. **Instalación de Vim:** Si aún no tienes Vim instalado en tu sistema, debes descargarlo e instalarlo. Puedes encontrar la última versión en el sitio web oficial de Vim o utilizar un gestor de paquetes si estás en un sistema operativo compatible.
2. **Creación del archivo .vimrc:** El archivo `.vimrc` es donde puedes personalizar la configuración de Vim. Puedes establecer atajos de teclado personalizados, activar o desactivar características específicas, y mucho más. Es tu espacio para hacer que Vim se sienta como en casa.
3. **Exploración de los ajustes básicos:** En tu archivo `.vimrc`, puedes establecer algunas configuraciones iniciales básicas, como el número de líneas visibles, el formato de tabulación, el resaltado de sintaxis y los atajos de teclado predefinidos. Estos ajustes pueden mejorar tu experiencia de edición desde el principio.

¡Ahora estás listo para comenzar tu viaje con Vim! En los siguientes apartados, te sumergirás en los atajos de teclado y comandos esenciales que te ayudarán a aprovechar al máximo este poderoso editor de texto.

2 Modos de Vim

Cuando se trata de Vim, uno de los conceptos clave que necesitas entender son sus modos de funcionamiento. Estos modos determinan cómo interactúas con el texto y son fundamentales para utilizar Vim de manera eficiente.

2.1 Modo normal

El modo normal es el modo principal de Vim. Aquí es donde puedes navegar por el archivo, realizar ediciones rápidas y ejecutar comandos. En este modo, las teclas que presiones se interpretan como comandos, no como caracteres para ingresar. Es como convertir tu teclado en un control remoto para manipular el texto.

2.2 Modo de inserción

El modo de inserción es donde puedes escribir y editar texto como en cualquier otro editor de texto convencional. Cuando estás en este modo, las teclas que presiones se insertan directamente en el archivo. Es el modo en el que puedes escribir tus palabras y expresarte libremente.

2.3 Modo de comando

El modo de comando es donde puedes ejecutar comandos más avanzados en Vim. Puedes utilizar estos comandos para buscar y reemplazar texto, realizar cambios en el archivo y personalizar la configuración de Vim. Aquí es donde puedes aprovechar el verdadero poder de Vim y llevar tus habilidades de edición al siguiente nivel.

Para intercambiar entre cada uno de los modos usa los siguientes atajos de teclado:

Modo	Shortcuts or com- mands	Función
Modo normal	<code>Esc</code> o <code>Ctrl+c</code>	Este atajo te lleva al modo normal de Vim. Aquí puedes navegar por el archivo, realizar ediciones rápidas y ejecutar comandos.
Modo de inserción	<code>i</code>	Presionar <code>i</code> te lleva al modo de inserción. Aquí puedes escribir y editar texto como en cualquier otro editor de texto.
Modo de comando	<code>:</code>	Al presionar <code>:</code> accederás al modo de comando. Aquí puedes ejecutar comandos avanzados en Vim. Desde buscar y reemplazar texto hasta personalizar la configuración.

3 Guardar documentos en Vim

Cuando trabajas en Vim, es importante saber cómo guardar tus documentos para asegurarte de que tus cambios se guarden correctamente. Afortunadamente, Vim ofrece atajos de teclado rápidos y sencillos para realizar esta tarea. A continuación, se muestra una tabla con los atajos de teclado y su función correspondiente:

Shortcuts or commands	Función
<code>:w</code>	Presionar <code>:w</code> te permite guardar el archivo actual. Esto guarda los cambios que hayas realizado sin cerrar Vim.
<code>:w nombre_archivo</code>	Al utilizar <code>:w</code> seguido de un nombre de archivo, puedes guardar el archivo actual con un nuevo nombre. Por ejemplo <code>:w index.md</code>
<code>:wq</code> o <code>:x</code>	Si deseas guardar el archivo y salir de Vim al mismo tiempo, puedes utilizar <code>:wq</code> o <code>:x</code> . Esto guarda los cambios y cierra Vim en un solo paso.

4 Atajos de navegación y movimiento

En Vim, existen diversos atajos de teclado que te permiten una mejor navegación y movimiento. A continuación, se muestran las diferentes categorías de atajos junto con sus respectivas funciones:

4.1 Movimiento básico con las teclas h, j, k, l

Shortcuts or commands	Función
<code>h</code>	La tecla <code>h</code> te permite mover el cursor hacia la izquierda.

Shortcuts or commands	Función
j	La tecla j te permite mover el cursor hacia abajo.
k	La tecla k te permite mover el cursor hacia arriba.
l	La tecla l te permite mover el cursor hacia la derecha.

4.2 Saltos rápidos en el archivo con gg y G

Shortcuts or commands	Función
gg	Presionar gg te lleva al comienzo del archivo.
G	Presionar G te lleva al final del archivo.

4.3 Navegación por palabras con w, b, e

Shortcuts or commands	Función
w	Presionar w te lleva al inicio de la siguiente palabra.
b	Presionar b te lleva al inicio de la palabra anterior.
e	Presionar e te lleva al final de la palabra actual.

4.4 Desplazamiento por páginas y ventanas

Shortcuts or commands	Función
Ctrl + f	Presionar Ctrl + f te permite desplazarte hacia adelante por una página.
Ctrl + b	Presionar Ctrl + b te permite desplazarte hacia atrás por una página.
Ctrl + u	Presionar Ctrl + u te permite desplazarte hacia arriba por media página.
Ctrl + d	Presionar Ctrl + d te permite desplazarte hacia abajo por media página.
Ctrl + w	Presionar Ctrl + w + w te permite cambiar de ventana en Vim. Si tienes varias ventanas abiertas, este atajo te lleva a la siguiente ventana.

4.5 Otros movimientos del cursor

Shortcuts or commands	Función
W	Mover el cursor al inicio de la siguiente palabra.
B	Mover el cursor al inicio de la palabra anterior.
E	Desplazar el cursor al final de la palabra actual.
O	Posicionar el cursor al inicio de la línea actual.
\$ o Fin	Posicionar el cursor al final de la línea actual.
-	Mover el cursor al primer carácter que no sea un espacio en la línea actual.

Shortcuts or commands	Función
+	Mover el cursor al primer carácter que no sea un espacio en la línea siguiente.
-	Mover el cursor al primer carácter que no sea un espacio en la línea anterior.
shift + a	Posicionar el cursor al final de la línea actual y cambiar al modo insertar.
shift + 5	Mover el cursor del inicio al final de un paréntesis y viceversa.

4.6 Navegación por el documento

Shortcuts or commands	Función
5 + Enter	Dirigirse 5 líneas más abajo respecto a la línea actual.
10gg	Posicionar el cursor en la línea 10 del documento.
16 + Mayús + g	Dirigirse a la línea 16 del fichero de texto.
}	Hacer saltar el cursor al párrafo siguiente.
{	Hacer saltar el cursor al párrafo anterior.
gi	Posicionar el cursor en la última palabra editada del buffer actual.
H	Mover el cursor a la parte superior de la pantalla.
M	Mover el cursor a la parte media de la pantalla.
L	Mover el cursor a la parte inferior de la pantalla.
Ctrl+e	Mover una pantalla hacia abajo sin mover el cursor de posición.
Ctrl+y	Mover una pantalla hacia arriba sin mover el cursor de posición.

5 Atajos de edición y manipulación de texto

En Vim, existen diversos atajos de teclado que te permiten realizar acciones de edición y manipulación de texto de forma eficiente. A continuación, se muestran las diferentes categorías de atajos junto con sus respectivas funciones explicadas:

5.1 Inserción y edición de texto

Shortcuts or commands	Función
i	Presionar i te permite insertar texto en el lugar donde se encuentra el cursor.
a	Presionar a te permite agregar texto después del lugar donde se encuentra el cursor.
A	Presionar A te lleva al final de la línea actual para comenzar a insertar texto.
o	Presionar o te permite abrir una nueva línea debajo de la línea actual para comenzar a escribir.
Ctrl+ flecha derecha	Avanzar una palabra hacia la derecha en modo inserción.
Ctrl+ flecha izquierda	Avanzar una palabra hacia la izquierda en modo inserción.
I	Insertar texto al principio de una línea.

Shortcuts or commands	Función
A	Insertar texto al final de una línea.
O	Crear una línea en blanco justo encima de la línea actual y pasar al modo inserción.
ea	Insertar texto después de una palabra.
Ctrl + p	Autocompletar la palabra que tenemos escrita a medias con otra palabra anterior al cursor.
Ctrl + w + cursores	Cambiar entre las distintas ventanas abiertas en Vim.
Ctrl+g	Mostrar el número de líneas de un fichero.

5.2 Inserción de texto repetitivo o secuencial

Shortcuts or commands	Función
40i- + Esc + Enter	Inserta 40 guiones.
10i + hola mundo + Esc + Enter	Escribe “hola mundo” 10 veces.
i + achalma + Esc + 20.	Escribe “achalma” 20 veces.
:put=range(1,10)	Escribe los números del 1 al 10.

5.3 Eliminación de caracteres, palabras y líneas

Shortcuts or commands	Función
x	Presionar x te permite eliminar el carácter bajo el cursor.
dw	Presionar dw te permite eliminar la palabra desde el cursor hasta el final.
dd	Presionar dd te permite eliminar toda la línea actual.

5.4 Copiar y pegar texto

Shortcuts or commands	Función
yy	Presionar yy te permite copiar la línea actual.
2yy	Copia 2 líneas a partir de donde está el cursor.
y\$	Copia desde la posición actual del cursor hasta el final de la línea.
yw	Copia la palabra a partir de la posición actual del cursor hasta el final de la palabra.
yiw	Copia la palabra actual.
:10,20y	Copia desde la línea 10 hasta la línea 20.
120y	Copia la línea 120.
dd	Corta la línea completa.
2dd	Corta 2 líneas a partir de donde está el cursor.
d\$ o D	Corta desde la posición actual del cursor hasta el final de la línea.

Shortcuts or commands	Función
<code>dw</code>	Corta la palabra desde la posición actual del cursor hasta el final de la palabra.
<code>diw</code>	Corta la palabra actual.
<code>x</code>	Corta un solo carácter.
<code>p</code>	Pega una línea debajo de la posición actual del cursor.
<code>:129put</code>	Pega el contenido del portapapeles en la línea 129 del documento.
<code>shift + p</code>	Pega una línea arriba de la posición actual del cursor.

“¡No te preocupes, también puedes copiar y pegar texto en VIM utilizando el modo visual! Sigue estos pasos sencillos:

1. Posiciona el cursor en el punto desde donde deseas comenzar a copiar.
2. Presiona la letra ‘v’ para entrar en el modo visual.
3. Utiliza los cursores o las teclas ‘j’, ‘k’, ‘h’ y ‘l’ para seleccionar el texto que deseas copiar, cortar o eliminar.
4. Si quieres copiar el texto seleccionado, simplemente presiona la tecla ‘y’. Si prefieres cortarlo, presiona ‘d’. Y si deseas eliminarlo por completo, usa la tecla ‘x’.
5. ¡Listo! Ahora puedes pegar el texto donde desees. Para ello, sal del modo visual y vuelve al modo normal, y luego presiona la tecla ‘p’.

5.5 Mover una línea de posición

Shortcuts or commands	Función
<code>:-1m.+1</code>	Mueve la línea justo encima de la línea actual a la línea inferior justo después de la línea actual.
<code>:19m17</code>	Mueve la línea 19 a la posición de la línea 17.
<code>:-4,.-2m.+8</code>	Mueve las líneas desde 4 líneas antes de la actual hasta 2 líneas antes de la actual, a una posición 8 líneas después de la posición actual del cursor.

6 Selección de texto y acciones en modo visual

6.1 Modo Visual

Shortcuts or commands	Función
<code>v</code>	Accede al modo visual.
<code>V</code>	Inicia el modo visual seleccionando toda una línea.
<code>Ctrl + v</code>	Inicia el modo Bloque visual.

6.2 Selección de Palabras

Shortcuts or commands	Función
<code>i w</code>	Selecciona la palabra donde se encuentra el cursor.
<code>a w</code>	Selecciona la palabra donde se encuentra el cursor, incluyendo el espacio que la precede.

6.3 Selección de Bloques

Shortcuts or commands	Función
<code>a b</code>	Selecciona un bloque de texto delimitado por paréntesis (). La selección incluye los paréntesis.
<code>i t</code>	Selecciona un bloque de texto delimitado por paréntesis (). La selección no incluye los paréntesis.
<code>a B</code>	Selecciona un bloque de texto delimitado por corchetes {}. La selección incluye los corchetes.
<code>i B</code>	Selecciona un bloque de texto delimitado por corchetes {}. La selección no incluye los corchetes.
<code>a t</code>	Selecciona un bloque de texto delimitado por etiquetas <> y </> incluyendo las etiquetas.
<code>i t</code>	Selecciona un bloque de texto delimitado por etiquetas <> y </> sin incluir las etiquetas.

6.4 Otras Acciones

Shortcuts or commands	Función
<code>o</code>	Mueve el cursor a la parte inicial de un bloque delimitado por (), {}, <>.. </>
<code>O</code>	Mueve el cursor a la parte final de un bloque delimitado por (), {}, <>.. </>
<code>j</code>	Selecciona una frase entera y se mueve a la siguiente línea.
<code>i s</code>	Selecciona una frase hasta el primer punto.
<code>i p</code>	Selecciona un párrafo completo.
<code>b</code>	Selecciona desde el cursor hasta el inicio de la palabra.
<code>e</code>	Selecciona desde el cursor hasta el final de la palabra.
<code>\$</code>	Selecciona desde el cursor hasta el final de la línea.
<code>^</code>	Selecciona desde el cursor hasta el primer carácter imprimible de la línea.
<code>awd\$ p</code>	Selecciona una palabra y la mueve al final de un párrafo.
<code>u</code>	Transforma todo el texto seleccionado en minúsculas.
<code>U</code>	Transforma todo el texto seleccionado en mayúsculas.
<code>></code>	Mueve la línea en la que se encuentra el cursor a la derecha. (Aplica sangría al texto.)
<code><</code>	Mueve la línea en la que se encuentra el cursor a la izquierda. (Aplica sangría al texto.)
<code>Esc</code>	Sale del modo visual.

7 Crear marcas en un fichero

7.1 Crear Marcas

Shortcuts or commands	Función
m + [a-z]	Crea una marca local con el nombre especificado. Puedes usar letras de a a z.

7.2 Trabajar con Marcas

Shortcuts or commands	Función
[a-z]	Posiciona el cursor en la marca especificada. Utiliza la letra correspondiente.
'	Regresa al inicio de la línea en la que se creó la marca.
'a	Mueve el cursor al inicio de la línea en la que se creó la marca.
'	Salta a la marca anterior. Si estás en la marca b, te trasladarás a la marca a.

7.3 Eliminar Marcas

Shortcuts or commands	Función
:delmarks a	Elimina la marca especificada (en este ejemplo, la marca a).
:delmarks!	Elimina todas las marcas locales o minúsculas del fichero.
:delmarks a-d	Elimina todas las marcas desde la marca a hasta la marca d.
:delmarks aBc	Elimina las marcas especificadas (en este ejemplo, las marcas a, B y c).

Nota: Si asignamos una marca con letras minúsculas, esta será una marca local y solo estará disponible mientras el fichero esté abierto. Si deseamos que la marca sea permanente, debemos asignarla con letras mayúsculas, creando así marcas globales.

8 Aplicación de sangrías a un texto

8.1 Aplicar Sangrías

Shortcuts or commands	Función
>>	Sangra el texto hacia la derecha.
3 >>	Realiza un sangrado en 3 líneas a partir de la posición actual.

8.2 Deshacer Sangría

Shortcuts or commands	Función
<<	Deshace la sangría, moviendo el texto hacia la izquierda.

8.3 Pegar y Ajustar a la Sangría Actual

Shortcuts or commands	Función
]p	Pega el contenido del portapapeles y lo ajusta a la sangría actual.

8.4 Aplicar Sangría en Modo Visual

Shortcuts or commands	Función
v + j + >	Selecciona una frase en modo visual y la sangra hacia la derecha.

8.5 Aplicar Sangría en Modo Insertar

Shortcuts or commands	Función
Ctrl+t	Aplica una sangría moviendo el texto hacia la derecha en modo insertar. X
Ctrl+d	Aplica una sangría moviendo el texto hacia la izquierda en modo insertar.

9 Deshacer y rehacer cambios

9.1 Deshacer Cambios

Shortcuts or commands o comando	Función
u	Deshace el último cambio realizado.
2u	Deshace los 2 últimos cambios realizados.

9.2 Restaurar Cambio

Shortcuts or commands o comando	Función
U	Restaura la última línea modificada. No se puede rehacer el cambio después de restaurarlo.

9.3 Rehacer Cambios

Shortcuts or commands o comando	Función
Ctrl + r or :redo	Rehace el último cambio realizado.
3 + Ctrl + r	Rehace los últimos 3 cambios deshechos.

9.4 Deshacer y Rehacer por Tiempo

Shortcuts or commands o comando	Función
:earlier 1h	Deshace todos los cambios realizados en la última hora.
:later 30m	Rehace los cambios realizados en la última media hora.

10 Eliminar caracteres y líneas

10.1 Eliminar Caracteres

Shortcuts or commands o comando	Función
x	Elimina un carácter en modo normal.
3x	Elimina 3 caracteres a partir de donde está el cursor.
X	Borra un carácter a la izquierda del cursor.
r	Reemplaza un carácter.
s	Elimina un carácter y cambia al modo insertar.

10.2 Eliminar Palabras

Shortcuts or commands o comando	Función
dw	Borra la palabra desde el cursor hasta el final.
cw	Borra la palabra desde el cursor hasta el final y cambia al modo insertar.
diw	Borra la palabra completa bajo el cursor.
ciw	Borra la palabra completa bajo el cursor y cambia al modo insertar.

10.3 Eliminar Líneas

Shortcuts or commands o comando	Función
dd	Borra/corta una línea completa.
shift + d	Borra desde el cursor hasta el final de la línea.
d0	Borra desde el cursor hasta el inicio de la línea.
cc o S	Borra toda la línea y cambia al modo insertar.
shift + c	Borra desde el cursor hasta el final de la línea y cambia al modo insertar.

Shortcuts or commands o comando	Función
c0	Borra desde el cursor hasta el inicio de la línea y cambia al modo insertar.
2dd	Borra/corta 2 líneas a partir de donde está el cursor.
2cc	Borra/corta 2 líneas a partir de donde está el cursor y cambia al modo insertar.

10.4 Eliminar por Patrón

Shortcuts or commands o comando	Función
:%d	Borra todas las líneas del archivo.
dgg	Borra desde el cursor hasta el inicio del archivo.
dG	Borra desde el cursor hasta el final del archivo.
: [2] , [4]	Borra las líneas de la 2 a la 4.
: 1 , .-1d	Borra todas las líneas antes de la línea actual.
: .+1, \$d	Borra todas las líneas después de la línea actual.
:g/<patrón>/d	Borra todas las líneas que contienen un patrón específico.
:g!/<patrón>/d	Borra todas las líneas que no contienen un patrón específico.
:g/^A/d	Borra todas las líneas que comienzan con la letra A.
:g/^\$/d	Borra todas las líneas vacías en el documento.
: .-2, .+8d	Borra las líneas desde 2 líneas encima hasta 8 líneas debajo de la posición actual.

¡Hay otra forma de eliminar texto en Vim utilizando el modo visual!

Si quieres utilizar marcas para eliminar texto, simplemente sigue estos pasos:

1. Ve a la primera línea que deseas borrar y crea una marca, por ejemplo, presiona ma.
2. Luego, coloca el cursor en la última línea que deseas eliminar.
3. Finalmente, presiona d'a y todas las líneas desde la posición actual del cursor hasta la línea marcada con la marca a se eliminarán.

Por otro lado, si prefieres eliminar texto en el modo visual, puedes consultar el apartado de “Atajos de teclado para copiar y pegar texto en Vim”. ¡Es una forma rápida y eficiente de seleccionar y eliminar el texto que desees!

11 Búsqueda y reemplazo

En Vim, la capacidad de buscar y reemplazar texto de manera eficiente es fundamental. A continuación, se presentan las diferentes categorías de acciones de búsqueda y reemplazo, junto con sus respectivos atajos de teclado y comandos:

11.1 Búsqueda de texto o frase

Shortcuts or commands	Función
/palabra_a_buscar	Presionar / seguido del texto que deseas buscar te lleva a la primera aparición de ese texto en el archivo.
n	Presionar n te lleva a la siguiente aparición de la búsqueda realizada previamente.
N	Presionar N te lleva a la aparición anterior de la búsqueda realizada previamente.
?palabra_a_buscar	Busca todas las palabras que contienen un texto determinado y coloca el cursor en la primera ocurrencia anterior a la ubicación actual del cursor.
n	Después de realizar una búsqueda, “n” lleva al siguiente resultado de búsqueda.
Shift+n	Similar a “n”, pero se desplaza hacia atrás en los resultados de búsqueda.
ggn	Va a la primera aparición de la palabra buscada.
Gn	Va a la última aparición de la palabra buscada.
3/palabra_a_buscar	Buscará todas las palabras que contengan “palabra_a_busca” y colocará el cursor en la tercera ocurrencia.

11.2 Navegación entre coincidencias de búsqueda

Shortcuts or commands	Función
*	Presionar * te lleva a la siguiente aparición de la palabra en la que se encuentra el cursor.
#	Presionar # te lleva a la aparición anterior de la palabra en la que se encuentra el cursor.

11.3 Reemplazo de texto

Shortcuts or commands	Función
:%s/buscar/reemplazar/g	Presionar :%s/buscar/reemplazar/g te permite reemplazar todas las apariciones de “buscar” por “reemplazar” en todo el archivo.
:s/buscar/reemplazar/g	Presionar :s/buscar/reemplazar/g te permite reemplazar solo la primera aparición de “buscar” por “reemplazar” en la línea actual.

11.4 Reemplazo de caracteres

Shortcuts or commands o fórmula	Función
r	Permite reemplazar un carácter cuando estás en modo normal.
R	Activa el modo de reemplazo para cambiar los caracteres deseados. No saldrás del modo de reemplazo hasta que presiones la tecla ESC.
ró	Reemplaza la letra señalada por el cursor por “ó”.

Shortcuts or commands o fórmula	Función
10ró	Reemplaza las 10 letras a partir de la posición actual del cursor por “ó”.

11.5 Búsqueda y reemplazo de palabras

Shortcuts or commands o fórmula	Función
:s/achalma/Achalma	Reemplaza todas las palabras que contienen “achalma” por “Achalma” solo en la línea actual.
:s/achalma/Achalma/g	Reemplaza todas las palabras que contienen “achalma” por “Achalma” en la línea del cursor sin pedir confirmación.
:%s/achalma/Achalma/g	Reemplaza todas las palabras que contienen “achalma” por “Achalma” en todo el documento sin pedir confirmación.
:%s/achalma/Achalma/gc	Reemplaza todas las ocurrencias de “achalma” por “Achalma” en el archivo pidiendo confirmación.
:%s!Achalma!ubuntu/scripts	Reemplaza todas las palabras que contienen “Achalma” por “ubuntu/scripts” utilizando el delimitador “!” y sin distinguir entre mayúsculas y minúsculas.
:%s/\<geek\>/Linux	Reemplaza la palabra exacta “geek” por “Linux” en todo el documento pidiendo confirmación.

11.6 Búsqueda y reemplazo en rangos específicos

Shortcuts or commands o fórmula	Función
:5,12s/foo/bar/g	Cambia “foo” por “bar” entre las líneas 5 y 12 (ambas incluidas).
: 'a, 'bs/foo/bar/g	Cambia “foo” por “bar” entre las marcas “a” y “b”.
:22s/Linux/Achalma	Reemplaza todas las palabras que contienen “Linux” por “Achalma” en la línea 22, distinguiendo entre mayúsculas y minúsculas.
:.s/Achalma/Linux	Reemplaza todas las palabras que contienen “Achalma” por “Linux” en la línea actual, distinguiendo entre mayúsculas y minúsculas.

11.7 Transformación de texto

Shortcuts or commands	Función
g+u+u	Transforma una frase completa a minúsculas.
g+U+U	Transforma una frase completa a mayúsculas.

12 Especificando rangos

12.1 Símbolos para especificar rangos

Símbolo	Significado
%	Aplica la acción a todo el documento
\$	Aplica la acción a la última línea del documento
.	Representa la línea actual en la que se encuentra el cursor

12.2 Especificación de rangos mediante marcas y números de línea

Rango	Significado
'a, 'b	Bloque de líneas entre las marcas 'a' y 'b'
:17,20d	Borra las líneas 17, 18, 19 y 20
:-2, +8y	Copia el contenido desde 2 líneas encima hasta 8 líneas debajo

13 Juntar 2 líneas

Shortcuts or commands o fórmula	Función
Shift + j	Une 2 líneas
gJ	Une la línea debajo del cursor con la actual sin dejar espacio entre ellas
:join	Fórmula a utilizar si se desean unir 2 líneas

14 Atajos de teclado para realizar pliegues manuales

Atajos de teclado	¿Qué hacen?
zf	Con este atajo puedes definir un pliegue en Vim.
<movimiento>	
zf4j	Puedes plegar las 4 líneas que se encuentran debajo del cursor.
zf2}	Este atajo pliega los 2 párrafos que están justo debajo del cursor.
zf2{	Con este atajo puedes plegar los 2 párrafos que están justo encima del cursor.
zo	Abre el pliegue en el que está posicionado el cursor.
zR	Despliega todos los pliegues en el documento.
zd	Elimina el pliegue que se encuentra encima del cursor.
zE	Elimina todos los pliegues en el documento.
zc	Cierra el pliegue que se encuentra encima del cursor.
zM	Cierra todos los pliegues existentes en el documento.
za	Abre o cierra el pliegue en el que se encuentra el cursor.
zj	Desplaza el cursor al siguiente pliegue.

Atajos de teclado	¿Qué hacen?
zk	Desplaza el cursor al pliegue anterior.

Nota: Para realizar pliegues de forma manual, debes agregar la siguiente línea al archivo de configuración `~/.vimrc: set foldmethod=manual`.

15 Registros o portapapeles

15.1 Mostrar y gestionar los registros

Shortcuts or commands	Función
<code>:registers</code>	Muestra el contenido almacenado en cada uno de los 48 registros o portapapeles de Vim.
<code>o :reg</code>	
<code>"1yy</code>	Copia la línea actual al registro numérico 1.
<code>"1p</code>	Pega el contenido del registro numérico 1.
<code>"ayy</code>	Copia la línea actual al registro nominal a. Si utilizas “a” en minúscula, se borrará completamente el contenido anterior del portapapeles a.
<code>"Ayy</code>	Añade la línea actual al registro nominal a. Si utilizas “A” en mayúscula, el contenido previamente guardado en a no se borrará, sino que se añadirá.
<code>"ap</code>	Pega el contenido del registro nominal a.
<code>:registers</code>	Muestra el contenido guardado en el registro a.
<code>a o :reg</code>	
<code>a</code>	
<code>:reg 4c</code>	Muestra el contenido almacenado en el registro numérico 4 y el registro nominal c.
<code>" +yy</code>	Copia la línea actual al portapapeles del sistema operativo.
<code>" +p</code>	Pega el contenido almacenado en el portapapeles del sistema operativo en el documento que estás editando en Vim.
<code>" *p</code>	Pega el texto seleccionado en tu navegador u otro software de tu sistema operativo. No es necesario que el texto esté copiado en el portapapeles del sistema.
<code>Ctrl+r+a</code>	Pega el contenido del registro a mientras estás en modo de inserción de texto.
<code>Ctrl+r "</code>	Pega el contenido del registro predeterminado del portapapeles de Vim en modo de inserción.
<code>Ctrl+r +</code>	Pega el contenido almacenado en el portapapeles del sistema operativo en modo de inserción.
<code>Ctrl+r *</code>	Pega el texto seleccionado en cualquier aplicación de tu sistema operativo en el modo de inserción de texto. No es necesario que el texto esté copiado en el portapapeles del sistema.

Nota: Hay 26 registros nominales, desde “a” hasta “z”.

15.2 Registros numéricos

Shortcuts or commands	¿Qué hacen?
"0yy	Copia la línea actual al registro numérico 0.
"0p	Pega el contenido del registro numérico 0.
"2yy	Copia la línea actual al registro numérico 2.
"2p	Pega el contenido del registro numérico 2.
...	...
"9yy	Copia la línea actual al registro numérico 9.
"9p	Pega el contenido del registro numérico 9.

15.3 Registros de lectura

Registro	Contenido almacenado
":	Guarda el último comando ejecutado.
%"	Contiene la ruta del archivo de Vim que estás editando.
".	Contiene el último texto insertado.

Estos registros de lectura te permiten acceder a información relevante, como el último comando ejecutado, la ruta del archivo en edición y el último texto insertado. Puedes utilizar estos registros en tus flujos de trabajo de Vim para mejorar tu productividad.

15.4 Registros especiales

Registro	Contenido almacenado
"_ "	Es el registro del “agujero negro”. Cuando eliminamos una palabra usando el atajo “_diw”, la palabra borrada se almacenará en este registro y el registro por defecto "" quedará vacío.
": "	Almacena el último comando ejecutado.
"- "	Contiene el contenido que hemos borrado o modificado y no es mayor a una línea de longitud.
"\	Guarda el contenido de la última búsqueda.
"=	Permite realizar operaciones matemáticas en Vim. Por ejemplo, puedes hacer una suma como =2+2 en modo normal y luego pegar el resultado con la tecla p.

Estos registros especiales en Vim ofrecen funcionalidades adicionales, como el registro del “agujero negro” para evitar que palabras borradas se almacenen en el registro por defecto, el almacenamiento del último comando ejecutado y el contenido borrado o modificado. También puedes realizar operaciones matemáticas y acceder al contenido de la última búsqueda.

Recuerda que todos los registros de Vim se almacenan permanentemente en el archivo ~/.viminfo, lo que significa que no se borran al cerrar Vim. Esto te permite mantener tus registros guardados y acceder a ellos en sesiones posteriores.

16 ¿Cómo usar la terminal de Linux dentro de Vim?

Shortcuts or commands	Función
<code>:term</code>	Divide la pantalla en dos. Una parte muestra Vim y la otra muestra la terminal. Puedes alternar entre las ventanas usando el atajo <code>Ctrl+w</code> .
<code>:!ls</code>	Ejecuta el comando <code>ls</code> dentro de Vim. Puedes usar otros comandos, como <code>sudo apt update</code> , en su lugar.
<code>:r</code>	Inserta la salida del comando ejecutado dentro del archivo de Vim que estás editando.
<code>!<comando></code>	

17 Encadenar comandos en Vim**

Shortcuts or commands	Función
<code>:w \ q</code>	Guarda los cambios en el documento y lo cierra.
<code>:17,20y \ 122put</code>	Copia las líneas del 17 al 20 y las pega en la línea 122.

Usando el símbolo de tubería `|`, puedes encadenar la ejecución de comandos en Vim para realizar acciones de forma simultánea. Estos ejemplos muestran cómo guardar los cambios en un documento y cerrarlo en una sola línea de comando, así como copiar un rango de líneas y pegarlo en una ubicación específica del documento.

Esta técnica te permite agilizar tu flujo de trabajo al evitar tener que introducir comandos por separado. Puedes aprovechar la potencia de Vim combinando múltiples acciones en una sola línea de comando utilizando la tubería `|`.

18 Trabajo con múltiples archivos y ventanas

En Vim, tienes la capacidad de trabajar con múltiples archivos y ventanas para organizar tu flujo de trabajo de manera eficiente. A continuación, se presentan las diferentes acciones relacionadas con la apertura, cierre y navegación entre archivos y ventanas, junto con sus respectivas Combinación de teclas o comandos:

18.1 Apertura de archivos

Shortcuts or commands	Función
<code>:e</code>	Presionar <code>:e</code> seguido del nombre del archivo te permite abrir ese archivo en una nueva pestaña.
<code>:badd +</code>	Añadir un buffer sin que se muestre en pantalla.
<code>nombre_fichero/buffer</code>	
<code>:ls</code>	Lista y enumera todos los buffers/archivos que tenemos abiertos.
<code>:bn</code>	Ir al siguiente archivo/buffer que tenemos abierto (<code>Buffer_next</code>).
<code>:bp</code>	Ir al archivo/buffer anterior que tenemos abierto (<code>Buffer_previous</code>).
<code>:bl</code>	Mostrar en pantalla el último buffer abierto.

Shortcuts or commands	Función
:bf	Mostrar en pantalla el primer buffer abierto.
:b_número_buffer	Cambiar al buffer especificado por su número. Por ejemplo, :b2 para cambiar al buffer número 2.
:bd	Cerrar únicamente el archivo/buffer actual que estamos editando.
:bd2	Cerrar el buffer número 2.
:bd	Cerrar el buffer del archivo específico. Por ejemplo, :bd
nombre_archivo	nombre_archivo.py cerrará el buffer del archivo “nombre_archivo.py”.
:ba	Mostrar todos los buffers en pantalla.

18.2 Cierre de archivos

Llega el momento de decir adiós a Vim y cerrar el editor de texto. Pero, ¿cómo puedes hacerlo de manera rápida y sencilla? Vim también tiene atajos de teclado específicos para salir del programa. Echa un vistazo a la siguiente tabla con los atajos de teclado y sus funciones correspondientes:

Shortcuts or commands	Función
:q	Presionar :q te permite salir de Vim si no hay cambios sin guardar en el archivo actual.
:q!	Si realizaste cambios en el archivo y deseas salir de Vim sin guardarlos, puedes utilizar :q!. Esto te permite cerrar Vim sin guardar los cambios.
:wq o :x	Si deseas guardar los cambios y salir de Vim al mismo tiempo, puedes utilizar :wq o :x. Esto guarda los cambios y cierra Vim en un solo paso.
:wqa!	El comando :wqa! te permite guardar todos los archivos abiertos en Vim y salir, incluso si hay cambios sin guardar.
:qa!	Si deseas cerrar Vim sin guardar los cambios en ninguno de los archivos abiertos, puedes utilizar :qa!. Esto cierra Vim de inmediato y descarta todos los cambios realizados.

18.3 Trabajar con pestañas

Las pestañas en Vim son una forma práctica de organizar y alternar entre múltiples archivos abiertos. Con las siguientes combinaciones de teclas, podrás trabajar con pestañas de manera eficiente:

Shortcuts or commands	Función
:tabnew	Abrir el archivo en una nueva pestaña.
nombre_archivo	
:tabc	Cerrar la pestaña actual.
:tabo	Cerrar todas las pestañas, excepto la actual.
gt o :tabnext	Cambiar a la siguiente pestaña.
gT o :tabprev	Cambiar a la pestaña anterior.

Shortcuts or commands	Función
Número gt	Cambiar a la pestaña con el número especificado.
:tabs	Listar y enumerar todas las pestañas abiertas.
:tabmove N	Mover la pestaña actual a la posición N (1-indexed).
:tabedit nombre_fichero	Abrir un nuevo archivo en una pestaña nueva. Por ejemplo, :tabedit archivo1.txt abrirá archivo1.txt en una nueva pestaña.
:tabfind nombre_fichero	Buscar y abrir un archivo en una pestaña nueva. Por ejemplo, :tabfind archivo2.txt buscará y abrirá archivo2.txt en una nueva pestaña.
:tab ball	Abrir todos los buffers en pestañas separadas.
:tab split	Dividir la ventana actual en dos pestañas.
:tabfirst	Cambiar a la primera pestaña.
:tablast	Cambiar a la última pestaña.
:tabm 3	Mover la pestaña actual a una posición específica. Por ejemplo, :tabm 3 moverá la pestaña actual a la posición 3.
:tabm 0	Mover la pestaña actual a la última posición.
:tabm	Mover la pestaña actual a la siguiente posición.
:tabclose! o :q!	Cerrar la pestaña actual sin guardar cambios.
:tabclose o :q	Cerrar la pestaña actual con confirmación.
:tabonly	Cerrar todas las pestañas excepto la actual.
vim -p nombre_archivo_1	Abrir múltiples archivos en pestañas separadas. Por ejemplo, vim -p archivo1.sh archivo2.sh abrirá archivo1.sh y archivo2.sh en
nombre_archivo_2	pestañas separadas.

18.4 División de la ventana en paneles y cambio entre ventanas y paneles

Shortcuts or commands	Función
:sp	Presionar :sp te permite dividir la ventana horizontalmente, creando un nuevo panel en la parte superior.
:vsp	Presionar :vsp te permite dividir la ventana verticalmente, creando un nuevo panel a la derecha.
Ctrl+ww	Presionar Ctrl+w seguido de v dividirá la ventana actual en dos verticales.
Ctrl+ws	Presionar Ctrl+w seguido de s dividirá la ventana actual en dos horizontales.
Ctrl+w+(h/j/k/l) o flecha	Cambiar a la ventana vecina en una dirección específica. Ctrl+w seguido de una de las teclas h, j, k o l cambiará a la ventana vecina en la dirección indicada.
Ctrl+ww	Presionar Ctrl+w dos veces cambiará a la siguiente ventana.
Ctrl+w	Presionar Ctrl+w cambiará a la ventana anteriormente seleccionada.
Ctrl+wq	Presionar Ctrl+w seguido de q cerrará la ventana actual.
Ctrl+wx	Presionar Ctrl+w seguido de x intercambiará la posición de las ventanas.
Ctrl+wr	Presionar Ctrl+w seguido de r rotará las ventanas hacia la derecha.
Ctrl+wR	Presionar Ctrl+w seguido de R rotará las ventanas hacia la izquierda.

Shortcuts or commands	Función
<code>:split</code>	Dividir la ventana actual horizontalmente con un archivo. Por ejemplo, <code>nombre_archivo:split archivo2.txt</code> dividirá la ventana actual horizontalmente y abrirá <code>archivo2.txt</code> .
<code>:resize 40</code>	Cambiar el tamaño de la ventana actual a 40 líneas.
<code>:vertical</code>	Cambiar el ancho de la ventana actual a 80 columnas.
<code>resize 80</code>	
<code>:vertical</code>	Reducir el ancho de la ventana actual en 5 columnas.
<code>resize -5</code>	
<code>:vertical</code>	Aumentar el ancho de la ventana actual en 5 columnas.
<code>resize +5</code>	
<code>Ctrl+w > y</code>	Presionar <code>Ctrl+w</code> seguido de <code>></code> o <code><</code> ajustará el tamaño de la ventana hacia la derecha o izquierda, respectivamente.
<code>Ctrl+w <</code>	
<code>Ctrl+w + y</code>	Presionar <code>Ctrl+w</code> seguido de <code>+</code> o <code>-</code> aumentará o reducirá el tamaño de la ventana verticalmente, respectivamente.
<code>Ctrl+w -</code>	
<code>Ctrl+w =</code>	Presionar <code>Ctrl+w</code> seguido de <code>=</code> igualará el tamaño de todas las ventanas.
<code>vim -o</code>	Abrir múltiples archivos en ventanas separadas. <code>vim -o archivo1.txt</code>
<code>archivo1</code>	<code>archivo2.txt</code> <code>archivo3.txt</code> abrirá <code>archivo1.txt</code> , <code>archivo2.txt</code> y
<code>archivo2</code>	<code>archivo3.txt</code> en ventanas separadas.
<code>archivo3</code>	

19 Personalización y configuración de Vim

Una de las ventajas más destacadas de Vim es su alta capacidad de personalización y configuración. Puedes adaptar el editor según tus preferencias y necesidades. A continuación, te presento las diferentes opciones para personalizar y configurar Vim, junto con sus respectivas características:

19.1 Personalización del archivo `.vimrc`

El archivo `.vimrc` es donde puedes definir tus preferencias de Vim. Puedes personalizar aspectos como el color del fondo, el tamaño de la fuente, las opciones de resaltado de sintaxis y mucho más. Edita este archivo para que Vim se ajuste a tu estilo y preferencias personales. ¡Haz de Vim tu editor único!

19.2 Uso de plugins y extensiones

Vim cuenta con una amplia gama de plugins y extensiones que te permiten ampliar sus funcionalidades. Puedes agregar complementos para la administración de proyectos, resaltado de sintaxis avanzado, autocompletado y mucho más. Explora la comunidad de plugins de Vim y encuentra las herramientas que se adapten mejor a tus necesidades. ¡Potencia tus capacidades editoriales!

19.3 Configuración de atajos de teclado personalizados

Vim te permite personalizar los atajos de teclado según tus preferencias. Puedes asignar tus propias combinaciones de teclas para realizar acciones específicas, como guardar un archivo, buscar y reemplazar, o incluso para ejecutar comandos personalizados. Configura tus atajos de

teclado para trabajar de manera más eficiente y adaptada a tu flujo de trabajo. ¡Crea tu propio mapa de atajos!

20 Consejos y trucos avanzados

En esta sección, exploraremos algunos consejos y trucos avanzados para aprovechar al máximo Vim. Estos recursos te permitirán mejorar tu flujo de trabajo y aumentar tu productividad. ¡Descubre cómo darle un impulso a tu experiencia con Vim!

20.1 Uso de marcadores

Los marcadores son una herramienta útil para marcar y acceder rápidamente a ubicaciones específicas en un archivo. Puedes colocar marcadores en líneas específicas y luego saltar entre ellos con facilidad. Usa `ma` para establecer un marcador en la posición actual y `m[a-z]` para asignar un marcador a una letra específica. Luego, utiliza `'a` para saltar al marcador. ¡No pierdas nunca la pista de tus puntos clave!

20.2 Autocompletado y omnicomplete

Vim ofrece características avanzadas de autocompletado que te ayudarán a escribir código más rápido y eficientemente. Utiliza `<Ctrl + n>` y `<Ctrl + p>` para navegar por las opciones de autocompletado mientras escribes. Además, con la función de omnicomplete, puedes obtener sugerencias contextuales y completar palabras automáticamente presionando `<Ctrl + x>` seguido de `<Ctrl + o>`. ¡Escribe código de forma más rápida y precisa!

20.3 Trabajo con macros

Las macros te permiten grabar y reproducir una serie de comandos en Vim. Esto es útil cuando tienes que realizar acciones repetitivas en varios lugares del archivo. Graba una macro usando `q[a-z]` seguido de los comandos que deseas grabar y luego reproduce la macro usando `@[a-z]`. ¡Automatiza tareas tediosas y ahorra tiempo!

20.4 Uso de scripts y automatización

Vim ofrece la posibilidad de utilizar scripts para personalizar y automatizar tareas. Puedes crear tus propios scripts en lenguajes como VimScript o Python y ejecutarlos dentro de Vim. Esto te permite realizar acciones complejas y personalizadas de manera más eficiente. Explora la documentación de Vim para aprender cómo escribir y ejecutar scripts. ¡Haz que Vim se adapte aún más a tus necesidades!

21 Recursos adicionales

En esta sección, te proporcionaremos algunos recursos adicionales para que puedas profundizar en tus conocimientos sobre Vim y aprovechar al máximo este poderoso editor de texto.

21.1 Referencias y documentación oficial de Vim

La documentación oficial de Vim es una fuente invaluable de información. Puedes acceder a ella a través del comando `:help` dentro de Vim. Este recurso abarca todos los aspectos del editor y proporciona explicaciones detalladas de los comandos, configuraciones y características. Consulta la documentación oficial para tener una referencia confiable y completa sobre Vim.

21.2 Comunidades y foros de usuarios de Vim

Existen diversas comunidades en línea donde puedes encontrar apoyo y compartir conocimientos con otros usuarios de Vim. Reddit cuenta con un subreddit dedicado a Vim ([/r/vim](https://www.reddit.com/r/vim/)), donde puedes hacer preguntas, compartir consejos y estar al tanto de las últimas novedades relacionadas con Vim. Además, hay foros como VimGolf y Vim Tips Wiki que ofrecen desafíos y consejos prácticos para mejorar tus habilidades en Vim. Únete a estas comunidades para aprender de otros entusiastas de Vim.

21.3 Libros y tutoriales recomendados sobre Vim

Si prefieres un enfoque más estructurado y detallado, hay varios libros y tutoriales disponibles para aprender Vim. Algunos títulos populares incluyen “Aprendiendo Vim en 21 días” de Tony Steidler-Dennison, “Vim Book” de Steve Oualline y “Practical Vim” de Drew Neil. Estos recursos te guiarán paso a paso a través de los conceptos y técnicas fundamentales de Vim. Además, hay numerosos tutoriales en línea, videos y cursos que puedes encontrar en plataformas educativas como Udemy y YouTube.

Explora estos recursos adicionales para continuar mejorando tus habilidades en Vim y descubrir nuevas técnicas y trucos. Recuerda que la práctica regular y la experimentación son clave para dominar Vim.

22 Publicaciones Similares

Si te interesó este artículo, te recomendamos que explores otros blogs y recursos relacionados que pueden ampliar tus conocimientos. Aquí te dejo algunas sugerencias:

1. [!\[\]\(e27c4336460e9e6729a19580c0456728_img.jpg\) Comandos De Informacion Windows](#)
2. [!\[\]\(1a140e8db538fd46d58af9f9540232fd_img.jpg\) Adb](#)
3. [!\[\]\(5a658b86f2c8900a276c586c1f8f9f2f_img.jpg\) Limpieza Y Optimizacion De Pc](#)
4. [!\[\]\(dde796100cc481a63a6f917e6942c754_img.jpg\) Usando Apk En Window 11](#)
5. [!\[\]\(63a8f188d537bd691c8d94f41db6869a_img.jpg\) Gestionar Versiones De Jdk En Kubuntu](#)
6. [!\[\]\(499fe69158060e68a02a9089268949e0_img.jpg\) Instalar Tor Browser](#)
7. [!\[\]\(c8aba30b21c2fae4d961d3c29bf22065_img.jpg\) Crear Enlaces Duros O Hard Link En Linux](#)
8. [!\[\]\(b5c199051809cfc34ae5dec3ec9d2866_img.jpg\) Comandos Vim](#)
9. [!\[\]\(e9cae8d4fa1ce243ac194a8add4b82b1_img.jpg\) Guia De Git Y Github](#)
10. [!\[\]\(cc62fe1f274eeb176e71927d4522f091_img.jpg\) 00 Primeros Pasos En Linux](#)
11. [!\[\]\(065bc57f97b4bb5a4ce195dc8a8a390b_img.jpg\) 01 Introduccion Linux](#)
12. [!\[\]\(0d030af95e670a94900800fc708efb5a_img.jpg\) 02 Distribuciones Linux](#)
13. [!\[\]\(712e5fc75eb586af16f0a1cb413fb475_img.jpg\) 03 Instalacion Linux](#)
14. [!\[\]\(b357cb95e3258bae822a3c9df892833f_img.jpg\) 04 Administracion Particiones Volumenes](#)
15. [!\[\]\(00255f5807d4769659fee838eef6824f_img.jpg\) Atajos De Teclado Y Comandos Para Usar Vim](#)
16. [!\[\]\(063d16f3ac1e746f86ce9724991399a3_img.jpg\) Instalando Specitify](#)
17. [!\[\]\(2aa5f8fa44241cb00d732be49faaf7bd_img.jpg\) Gestiona Tus Dotfiles Con Gnu Stow](#)

Esperamos que encuentres estas publicaciones igualmente interesantes y útiles. ¡Disfruta de la lectura!