

# Guía Completa de Vim y Neovim

Edison Achalma

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

## Resumen

Esta guía exhaustiva presenta una introducción progresiva y práctica al editor de texto Vim y su evolución moderna, Neovim. Se abordan desde los fundamentos filosóficos y modales del editor hasta técnicas avanzadas de edición, navegación eficiente, composición de comandos, uso de registros, macros, múltiples ventanas y buffers, búsqueda/reemplazo con expresiones regulares, plegado, marcas y configuración personalizada tanto en Vimscript como en Lua. Se detallan las diferencias clave entre ambas versiones, con énfasis en las ventajas contemporáneas de Neovim; soporte nativo para LSP, Tree-sitter, arquitectura asíncrona y configuración en Lua. La obra incluye instrucciones detalladas de instalación en múltiples sistemas operativos, una selección comentada de plugins esenciales, mapeos prácticos, trucos avanzados y una hoja de ruta de aprendizaje por niveles. Dirigida tanto a principiantes que buscan superar la curva inicial de aprendizaje como a usuarios intermedios-avanzados que desean optimizar su flujo de trabajo en entornos Linux/Unix, esta guía busca servir como referencia completa y actualizada para dominar uno de los editores de texto más potentes y ubicuos del ecosistema del software libre.

*Palabras Claves:* Vim, Neovim, Comandos Vim

## Tabla de contenidos

<b>Introduction</b>	<b>6</b>
<b>1 Instalación</b>	<b>6</b>
1.1 Instalar Vim . . . . .	6
1.2 Instalar Neovim . . . . .	7
1.3 Verificar Instalación . . . . .	7
1.4 Configurar Alias . . . . .	7

---

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](mailto:elmer.achalma.09@unsch.edu.pe)

<b>2 Filosofía y Conceptos Básicos</b>	<b>8</b>
2.1 Filosofía Modal . . . . .	8
2.2 Composición de Comandos . . . . .	8
2.3 Filosofía “Modal” . . . . .	8
2.4 El Camino del Vim . . . . .	8
<b>3 Modos de Vim</b>	<b>8</b>
3.1 Modo Normal (Default) . . . . .	8
3.2 Modo Insert . . . . .	9
3.3 Modo Visual . . . . .	9
3.4 Modo Command (Command-line) . . . . .	9
3.5 Modo Replace . . . . .	10
3.6 Modo Select . . . . .	10
<b>4 Navegación Básica</b>	<b>10</b>
4.1 Movimiento de Cursor (Modo Normal) . . . . .	10
4.2 Básico (hjkl) . . . . .	10
4.3 Por Palabras . . . . .	10
4.4 Por Línea . . . . .	11
4.5 Por Pantalla . . . . .	11
4.6 Por Archivo . . . . .	11
4.7 Búsqueda de Caracteres en Línea . . . . .	11
<b>5 Edición de Texto</b>	<b>12</b>
5.1 Insertar . . . . .	12
5.2 Eliminar . . . . .	12
5.3 Cambiar (Change) . . . . .	12
5.4 Copiar y Pegar (Yank/Put) . . . . .	13
5.5 Reemplazar . . . . .	13
5.6 Deshacer y Rehacer . . . . .	13
5.7 Repetir . . . . .	13
5.8 Unir Líneas . . . . .	14
5.9 Indentación . . . . .	14
<b>6 Comandos de Búsqueda</b>	<b>14</b>
6.1 Búsqueda Básica . . . . .	14
6.2 Opciones de Búsqueda . . . . .	14
6.3 Expresiones Regulares . . . . .	15
<b>7 Modo Visual</b>	<b>15</b>
7.1 Tipos de Selección Visual . . . . .	15
7.2 Operaciones en Visual . . . . .	15
7.3 Visual Block (Columnar) . . . . .	16
7.4 Objetos de Texto . . . . .	16
<b>8 Registros y Macros</b>	<b>17</b>
8.1 Registros . . . . .	17
8.2 Tipos de Registros . . . . .	17
8.3 Usar Registros . . . . .	17

8.4 Ver Registros . . . . .	17
8.5 Macros . . . . .	17
8.6 Grabar Macro . . . . .	17
8.7 Ejecutar Macro . . . . .	18
8.8 Editar Macros . . . . .	18
<b>9 Ventanas y Pestañas</b>	<b>18</b>
9.1 Ventanas (Splits) . . . . .	18
9.2 Crear Ventanas . . . . .	18
9.3 Navegar Entre Ventanas . . . . .	19
9.4 Mover Ventanas . . . . .	19
9.5 Redimensionar Ventanas . . . . .	19
9.6 Cerrar Ventanas . . . . .	20
9.7 Pestañas (Tabs) . . . . .	20
9.8 Gestión de Pestañas . . . . .	20
<b>10 Buffers</b>	<b>20</b>
10.1 ¿Qué son los Buffers? . . . . .	20
10.2 Comandos de Buffer . . . . .	20
10.3 Navegación Rápida . . . . .	20
<b>11 Búsqueda y Reemplazo</b>	<b>21</b>
11.1 Comando Substitute . . . . .	21
11.2 Rangos . . . . .	21
11.3 Flags . . . . .	21
11.4 Ejemplos Prácticos . . . . .	21
11.5 Caracteres Especiales en Reemplazo . . . . .	22
<b>12 Plegado (Folding)</b>	<b>23</b>
12.1 ¿Qué es Folding? . . . . .	23
12.2 Métodos de Plegado . . . . .	23
12.3 Comandos de Plegado . . . . .	23
12.4 Nivel de Plegado . . . . .	23
<b>13 Marcas y Saltos</b>	<b>23</b>
13.1 Marcas (Marks) . . . . .	23
13.2 Establecer Marcas . . . . .	24
13.3 Ir a Marcas . . . . .	24
13.4 Ver Marcas . . . . .	24
13.5 Lista de Saltos (Jump List) . . . . .	24
13.6 Lista de Cambios (Change List) . . . . .	24
<b>14 Comandos Ex</b>	<b>24</b>
14.1 Comandos de Archivo . . . . .	24
14.2 Comandos de Shell . . . . .	25
14.3 Comandos de Ayuda . . . . .	25

<b>15 Configuración (.vimrc/init.vim)</b>	<b>25</b>
15.1 Ubicaciones . . . . .	25
15.2 Configuración Básica . . . . .	26
15.3 Opciones Útiles . . . . .	29
<b>16 Plugins</b>	<b>30</b>
16.1 Gestores de Plugins . . . . .	30
16.2 vim-plug (Recomendado) . . . . .	30
16.3 Packer (Neovim + Lua) . . . . .	30
16.4 Plugins Esenciales . . . . .	30
16.5 1. Navegación de Archivos . . . . .	30
16.6 2. Edición . . . . .	31
16.7 3. Git . . . . .	31
16.8 4. Temas . . . . .	32
16.9 5. Statusline . . . . .	32
<b>17 Neovim - Características Especiales</b>	<b>32</b>
17.1 Diferencias con Vim . . . . .	32
17.2 Configuración en Lua . . . . .	33
17.3 Tree-sitter . . . . .	34
17.4 Telescope (Fuzzy Finder) . . . . .	34
<b>18 LSP y Autocompletado</b>	<b>35</b>
18.1 LSP en Neovim . . . . .	35
18.2 Mapeos LSP . . . . .	35
18.3 nvim-cmp (Autocompletado) . . . . .	35
<b>19 Vimscript y Lua</b>	<b>36</b>
19.1 Vimscript Básico . . . . .	36
19.2 Lua Básico . . . . .	37
<b>20 Trucos Avanzados</b>	<b>38</b>
20.1 1. Edición de Múltiples Archivos . . . . .	38
20.2 2. Edición Avanzada con Expresiones . . . . .	38
20.3 3. Cambiar en Múltiples Líneas . . . . .	38
20.4 4. Formatear Código . . . . .	39
20.5 5. Diff Files . . . . .	39
20.6 6. Sesiones . . . . .	39
20.7 7. Spell Checking . . . . .	39
20.8 8. Abrir URL . . . . .	39
20.9 9. Ejecutar Línea como Comando . . . . .	40
20.10 10. Formateo de Tabla . . . . .	40
<b>21 Solución de Problemas</b>	<b>40</b>
21.1 Problemas Comunes . . . . .	40
21.2 1. Vim/Neovim Lento . . . . .	40
21.3 2. Clipboard No Funciona . . . . .	40
21.4 3. Colores Incorrectos . . . . .	40
21.5 4. LSP No Funciona (Neovim) . . . . .	41

21.6 5. Plugins No Se Instalan . . . . .	41
21.7 Debugging . . . . .	41
<b>22 Referencia Rápida</b>	<b>42</b>
22.1 Atajos Esenciales . . . . .	42
22.2 Comandos por Modo . . . . .	43
<b>23 Conclusión</b>	<b>43</b>
23.1 El Viaje de Aprendizaje . . . . .	43
23.2 Recursos . . . . .	44
23.3 Tips Finales . . . . .	44
<b>24 Publicaciones Similares</b>	<b>45</b>

## Guía Completa de Vim y Neovim

### ¿Qué es Vim?

Vim (Vi IMproved) es un editor de texto modal altamente configurable, creado por Bram Moolenaar como una versión mejorada del editor Unix vi. Es conocido por:

- **Eficiencia:** Una vez dominado, permite editar texto extremadamente rápido
- **Ubicuidad:** Disponible en prácticamente todos los sistemas Unix/Linux
- **Potencia:** Capacidades avanzadas de edición y automatización
- **Extensibilidad:** Sistema robusto de plugins y configuración

### ¿Qué es Neovim?

Neovim es una refactorización moderna de Vim que mantiene compatibilidad mientras añade:

- **Arquitectura moderna:** Cliente-servidor, API asíncrona
- **LSP integrado:** Language Server Protocol nativo
- **Lua:** Lenguaje de configuración más rápido que Vimscript
- **UI remota:** Soporte para interfaces gráficas externas
- **Rendimiento:** Más rápido y eficiente
- **Desarrollo activo:** Comunidad muy activa

### Diferencias Principales

Característica	Vim	Neovim
Lenguaje config	Vimscript	Vimscript + Lua
LSP	Plugin	Integrado
Async	Plugin	Nativo
Tree-sitter	No	Sí
UI externa	Limitada	Completa
Desarrollo	Conservador	Agresivo

## 1 Instalación

### 1.1 Instalar Vim

#### Ubuntu/Debian:

```
sudo apt update
sudo apt install vim
```

#### Fedora:

```
sudo dnf install vim-enhanced
```

#### macOS:

```
brew install vim
```

**Windows:** Descargar desde: <https://www.vim.org/download.php>

## 1.2 Instalar Neovim

**Ubuntu/Debian (PPA):**

```
sudo add-apt-repository ppa:neovim-ppa/stable
sudo apt update
sudo apt install neovim
```

**Fedora:**

```
sudo dnf install neovim
```

**macOS:**

```
brew install neovim
```

**Arch Linux:**

```
sudo pacman -S neovim
```

**Desde AppImage (cualquier Linux):**

```
curl -LO https://github.com/neovim/neovim/releases/latest/download/nvim.appimage
chmod u+x nvim.appimage
./nvim.appimage
```

**Windows:**

```
choco install neovim
# O
scoop install neovim
```

## 1.3 Verificar Instalación

```
# Vim
vim --version

# Neovim
nvim --version
```

## 1.4 Configurar Alias

Para usar `vim` para lanzar Neovim:

```
# En ~/.bashrc o ~/.zshrc
alias vim='nvim'
alias vi='nvim'
```

## 2 Filosofía y Conceptos Básicos

### 2.1 Filosofía Modal

Vim usa **modos** diferentes para distintas tareas:

- **Normal:** Navegar y ejecutar comandos (modo por defecto)
- **Insert:** Insertar texto
- **Visual:** Seleccionar texto
- **Command:** Ejecutar comandos Ex

Esta separación permite:

- **Eficiencia:** Cada tecla hace algo útil en modo Normal
- **Composición:** Combinar operadores con movimientos
- **Repetición:** El comando . repite la última acción

### 2.2 Composición de Comandos

Vim usa una gramática:

[operador] [movimiento]  
[operador] [número] [movimiento]

#### Ejemplos:

- d2w = Eliminar (delete) 2 palabras (words)
- c\$ = Cambiar (change) hasta fin de línea
- y3j = Copiar (yank) 3 líneas hacia abajo

### 2.3 Filosofía “Modal”

- **Separación de tareas:** No mezclar edición y navegación
- **Comando vs texto:** Modo Normal para comandos, Insert para texto
- **Eficiencia sobre simplicidad:** Curva de aprendizaje pero altamente eficiente

### 2.4 El Camino del Vim

1. **Semana 1-2:** Básicos (hjkl, i, a, o, Esc, :wq)
2. **Mes 1:** Movimientos (w, b, f, t, /, n)
3. **Meses 2-3:** Operadores (d, c, y con movimientos)
4. **Meses 4-6:** Visual, macros, registros
5. **Después:** Plugins, configuración avanzada, Vimscript/Lua

## 3 Modos de Vim

### 3.1 Modo Normal (Default)

El modo por defecto. Todas las teclas son comandos. Para activar el modo comando en Vim, debes presionar la tecla “Esc”. Esto te llevará al modo comando desde cualquier otro modo en el que te encuentres, como el modo insertar o el modo de reemplazo. Una vez que estés en el modo comando, puedes utilizar una variedad de comandos y combinaciones de teclas para navegar, editar y guardar tus archivos. Para salir de Vim, puedes ingresar el comando “:q” seguido de Enter. Si has realizado cambios y deseas guardarlos antes de salir, utiliza el comando “:wq” para escribir y guardar los cambios y salir de Vim.

**Entrar:**

- Esc desde cualquier modo
- Ctrl+[ (alternativa a Esc)
- Ctrl+c (similar a Esc)

### 3.2 Modo Insert

Para insertar texto.

**Entrar desde Normal:**

- i - Insertar antes del cursor
- I - Insertar al inicio de la línea
- a - Insertar después del cursor (append)
- A - Insertar al final de la línea
- o - Abrir línea nueva abajo
- O - Abrir línea nueva arriba
- s - Sustituir carácter (eliminar + insert)
- S - Sustituir línea completa
- C - Cambiar hasta fin de línea (c\$)

**Salir:**

- Esc o Ctrl+[ - Volver a Normal

### 3.3 Modo Visual

Para seleccionar texto.

**Entrar desde Normal:**

- v - Visual por carácter
- V - Visual por línea
- Ctrl+v - Visual en bloque

**Operaciones:**

Una vez seleccionado:

- d - Eliminar
- c - Cambiar
- y - Copiar
- > - Indentar
- < - Desindentar
- ~ - Cambiar mayúsculas/minúsculas

### 3.4 Modo Command (Command-line)

Para ejecutar comandos Ex.

**Entrar desde Normal:**

- : - Comando Ex
- / - Búsqueda hacia adelante
- ? - Búsqueda hacia atrás

**Comandos comunes:**

- :w - Guardar (write)
- :q - Salir (quit)
- :wq o :x - Guardar y salir
- :q! - Salir sin guardar
- :e archivo - Abrir archivo (edit)
- :help tema - Ayuda

### 3.5 Modo Replace

Sobrescribir texto existente.

**Entrar desde Normal:**

- R - Modo replace (sobrescribir)
- r - Reemplazar un solo carácter

### 3.6 Modo Select

Similar a selección en editores normales.

**Entrar:**

- gh - Select mode por carácter
- gH - Select mode por línea
- g Ctrl+h - Select mode en bloque

## 4 Navegación Básica

### 4.1 Movimiento de Cursor (Modo Normal)

#### 4.2 Básico (hjkl)

Tecla	Movimiento
h	← Izquierda
j	↓ Abajo
k	↑ Arriba
l	→ Derecha

#### 4.3 Por Palabras

Tecla	Movimiento
w	Inicio de siguiente palabra
W	Inicio de siguiente PALABRA (ignora puntuación)
e	Final de palabra actual/siguiente
E	Final de PALABRA
b	Inicio de palabra anterior
B	Inicio de PALABRA anterior
ge	Final de palabra anterior

**Nota:** “palabra” = letras/números, “PALABRA” = cualquier carácter no-espacio

#### 4.4 Por Línea

Tecla	Movimiento
0	Inicio de línea (columna 0)
^	Primer carácter no-blanco
\$	Final de línea
g_	Último carácter no-blanco
g0	Inicio de línea visual (con wrap)
g\$	Final de línea visual

#### 4.5 Por Pantalla

Tecla	Movimiento
H	Top de pantalla (High)
M	Medio de pantalla (Middle)
L	Bottom de pantalla (Low)
zt	Scroll: línea actual arriba
zz	Scroll: línea actual al centro
zb	Scroll: línea actual abajo
Ctrl+e	Scroll abajo una línea
Ctrl+y	Scroll arriba una línea
Ctrl+d	Media página abajo
Ctrl+u	Media página arriba
Ctrl+f	Página completa abajo (forward)
Ctrl+b	Página completa arriba (backward)

#### 4.6 Por Archivo

Tecla	Movimiento
gg	Inicio del archivo
G	Final del archivo
:<n> o <n>G	Ir a línea n
%	Ir al paréntesis/llave/corchoete coincidente

#### 4.7 Búsqueda de Caracteres en Línea

Tecla	Movimiento
f{char}	Find: ir a siguiente {char}
F{char}	Find: ir a anterior {char}
t{char}	Till: ir antes de siguiente {char}
T{char}	Till: ir después de anterior {char}
;	Repetir último f/F/t/T
,	Repetir último f/F/t/T en reversa

---

**Ejemplos:**

Esta es una línea de ejemplo

Con cursor en 'E':

- **fw** - Ir a 'u' (primera 'u')
- **3fl** - Ir a tercera 'l'
- **t .** - Ir antes del punto

## 5 Edición de Texto

### 5.1 Insertar

---

Comando	Acción
i	Insertar antes del cursor
I	Insertar al inicio de línea
a	Insertar después del cursor (append)
A	Insertar al final de línea
o	Abrir línea nueva abajo
O	Abrir línea nueva arriba
gi	Insertar en última posición de inserción

---

### 5.2 Eliminar

---

Comando	Acción
x	Eliminar carácter bajo cursor
X	Eliminar carácter antes del cursor
dd	Eliminar línea
D	Eliminar hasta fin de línea d\$
d{movimiento}	Eliminar hasta movimiento
dw	Eliminar palabra
db	Eliminar palabra hacia atrás
d\$ o D	Eliminar hasta fin de línea
d0	Eliminar hasta inicio de línea
dG	Eliminar hasta fin de archivo
dgg	Eliminar hasta inicio de archivo

---

### 5.3 Cambiar (Change)

Igual que eliminar pero entra en modo Insert.

---

Comando	Acción
c{movimiento}	Cambiar texto

---

Comando	Acción
cc o S	Cambiar línea completa
C	Cambiar hasta fin de línea
cw	Cambiar palabra
ciw	Cambiar palabra interna (inner word)
ci"	Cambiar dentro de comillas
ci(	Cambiar dentro de paréntesis
cit	Cambiar dentro de tag HTML

## 5.4 Copiar y Pegar (Yank/Put)

Comando	Acción
y{movimiento}	Copiar (yank)
yy o Y	Copiar línea
yw	Copiar palabra
y\$	Copiar hasta fin de línea
p	Pegar después del cursor
P	Pegar antes del cursor
gp	Pegar y mover cursor después
gP	Pegar antes y mover cursor

## 5.5 Reemplazar

Comando	Acción
r{char}	Reemplazar carácter bajo cursor
R	Entrar en modo Replace
~	Cambiar mayúscula/minúscula
g~{movimiento}	Cambiar caso
gu{movimiento}	A minúsculas
gU{movimiento}	A MAYÚSCULAS

## 5.6 Deshacer y Rehacer

Comando	Acción
u	Deshacer (undo)
Ctrl+r	Rehacer (redo)
U	Deshacer todos los cambios en línea

## 5.7 Repetir

Comando	Acción
.	Repetir último cambio

Comando	Acción
@:	Repetir último comando Ex
@@	Repetir último macro

## 5.8 Unir Líneas

Comando	Acción
J	Unir línea siguiente con actual
gJ	Unir sin añadir espacio

## 5.9 Indentación

Comando	Acción
>>	Indentar línea
<<	Desindentar línea
==	Auto-indentar línea
>%	Indentar bloque (con cursor en paréntesis)
gg=G	Auto-indentar archivo completo

# 6 Comandos de Búsqueda

## 6.1 Búsqueda Básica

Comando	Acción
/patrón	Buscar hacia adelante
?patrón	Buscar hacia atrás
n	Siguiente ocurrencia
N	Ocurrencia anterior
*	Buscar palabra bajo cursor (adelante)
#	Buscar palabra bajo cursor (atrás)
g*	Buscar parcial hacia adelante
g#	Buscar parcial hacia atrás

## 6.2 Opciones de Búsqueda

```
:set ignorecase      " Ignorar mayúsculas/minúsculas
:set smartcase       " Case-sensitive si hay mayúsculas
:set incsearch        " Búsqueda incremental
:set hlsearch         " Resaltar búsquedas
:nohlsearch          " Quitar resaltado (o :noh)
```

### 6.3 Expresiones Regulares

**Metacaracteres básicos:**

- . - Cualquier carácter
- \* - 0 o más del anterior
- ^ - Inicio de línea
- \$ - Fin de línea
- [] - Conjunto de caracteres
- \< - Inicio de palabra
- \> - Fin de palabra

**Ejemplos:**

```
/\<vim\>          " Palabra exacta "vim"
/^#               " Líneas que empiezan con #
/error.*$         " "error" hasta fin de línea
/[0-9]+\+        " Uno o más dígitos
/\d\{3}-\d\{4}   " XXX-XXXX (teléfono)
```

## 7 Modo Visual

### 7.1 Tipos de Selección Visual

Comando	Tipo
v	Visual por carácter
V	Visual por línea
Ctrl+v	Visual en bloque (columnar)
gv	Reseleccionar última selección

### 7.2 Operaciones en Visual

Una vez en modo visual:

Comando	Acción
d	Eliminar selección
c	Cambiar selección
y	Copiar selección
>	Indentar
<	Desindentar
=	Auto-indentar
~	Cambiar mayúsculas/minúsculas
u	A minúsculas
U	A MAYÚSCULAS
o	Ir al otro extremo de selección
0	Ir a otra esquina (bloque)

### 7.3 Visual Block (Columnar)

**Casos de uso:**

1. Insertar en múltiples líneas
2. Eliminar columnas
3. Cambiar bloques

**Ejemplo - Insertar en múltiples líneas:**

1. Ctrl+v (visual block)
2. Seleccionar líneas (j/k)
3. I (Insert al inicio)
4. Escribir texto
5. Esc (se aplica a todas las líneas)

### 7.4 Objetos de Texto

Operan sobre “objetos” como palabras, párrafos, etc.

**Sintaxis:** {operador}{a/i}{objeto}

Objeto	Descripción
w	Palabra (word)
W	PALABRA
s	Sentencia
p	Párrafo
[ o ]	Bloque []
( o )	Bloque ()
{ o }	Bloque {}
< o >	Bloque <>
t	Tag HTML
"	Comillas dobles
'	Comillas simples
`	Backticks

**Modificadores:**

- a - “a” (around) - incluye delimitadores
- i - “inner” - solo contenido

**Ejemplos:**

```
ciw    " Change inner word
da"    " Delete a quote (incluye comillas)
yi(    " Yank inner parentheses
vit    " Visual select inner tag
ci{    " Change inner braces
```

## 8 Registros y Macros

### 8.1 Registros

Vim tiene múltiples “clipboards” llamados registros.

### 8.2 Tipos de Registros

Registro	Descripción
""	Sin nombre (defecto)
"0	Último yank
"1-" <sup>9</sup>	Historial de deletes
"a-" <sup>z</sup>	Nombrados (minúsculas)
"A-" <sup>Z</sup>	Nombrados (append)
"_	Agujero negro (no guarda)
"+	Clipboard del sistema
"*	Clipboard de selección (X11)
"%	Nombre del archivo actual
"#	Nombre del archivo alterno
"/	Último patrón de búsqueda
Último comando Ex	

### 8.3 Usar Registros

**Sintaxis:** "{registro}{comando}

```
"ayy      " Copiar línea al registro a
"ap      " Pegar desde registro a
"Ayy      " Añadir línea al registro a
"_dd      " Eliminar sin guardar
"+y      " Copiar al clipboard del sistema
"+p      " Pegar desde clipboard
```

### 8.4 Ver Registros

```
:registers      " Ver todos los registros
:reg a b c      " Ver registros específicos
```

### 8.5 Macros

Grabar y repetir secuencias de comandos.

### 8.6 Grabar Macro

```
q{registro}      " Empezar a grabar en registro
... comandos ...
q                  " Terminar grabación
```

## 8.7 Ejecutar Macro

```
@{registro}      " Ejecutar macro
@@                  " Repetir último macro
{n}@{registro}  " Ejecutar n veces
```

### Ejemplo - Formatear lista:

```
# Lista:
item 1
item 2
item 3

# Grabar macro en registro a:
qa          " Empezar grabación
I- <Esc>    " Insertar "-" al inicio
j           " Bajar una línea
q           " Terminar grabación

# Ejecutar:
2@a        " Aplicar a siguientes 2 líneas

# Resultado:
- item 1
- item 2
- item 3
```

## 8.8 Editar Macros

```
# Ver contenido del registro
:echo @a

# Editar
:let @a = 'nuevo contenido'

# O pegar en buffer, editar, y copiar de vuelta
"ap          " Pegar macro
... editar ...
"ayy         " Copiar de vuelta
```

## 9 Ventanas y Pestañas

### 9.1 Ventanas (Splits)

### 9.2 Crear Ventanas

Comando	Acción
:split o :sp	Dividir horizontalmente
:vsplit o :vsp	Dividir verticalmente
:new	Nueva ventana horizontal
:vnew	Nueva ventana vertical
Ctrl+w s	Split horizontal
Ctrl+w v	Split vertical
:sp archivo	Abrir archivo en split

### 9.3 Navegar Entre Ventanas

Comando	Acción
Ctrl+w h	Ir a ventana izquierda
Ctrl+w j	Ir a ventana abajo
Ctrl+w k	Ir a ventana arriba
Ctrl+w l	Ir a ventana derecha
Ctrl+w w	Ciclar entre ventanas
Ctrl+w p	Ir a ventana anterior

### 9.4 Mover Ventanas

Comando	Acción
Ctrl+w H	Mover ventana a la izquierda
Ctrl+w J	Mover ventana abajo
Ctrl+w K	Mover ventana arriba
Ctrl+w L	Mover ventana a la derecha
Ctrl+w r	Rotar ventanas
Ctrl+w x	Intercambiar con siguiente

### 9.5 Redimensionar Ventanas

Comando	Acción
Ctrl+w =	Igualar tamaños
Ctrl+w +	Aumentar altura
Ctrl+w -	Disminuir altura
Ctrl+w >	Aumentar ancho
Ctrl+w <	Disminuir ancho
Ctrl+w _	Maximizar altura
Ctrl+w	Maximizar ancho
:resize {n}	Establecer altura
:vertical resize {n}	Establecer ancho

## 9.6 Cerrar Ventanas

Comando	Acción
:q o Ctrl+w q	Cerrar ventana actual
:only o Ctrl+w o	Cerrar todas menos actual

## 9.7 Pestañas (Tabs)

### 9.8 Gestión de Pestañas

Comando	Acción
:tabnew	Nueva pestaña
:tabe archivo	Abrir archivo en pestaña
:tabc	Cerrar pestaña actual
:tabo	Cerrar todas las pestañas menos actual
gt	Siguiente pestaña
gT	Pestaña anterior
{n}gt	Ir a pestaña n
:tabs	Listar pestañas

## 10 Buffers

### 10.1 ¿Qué son los Buffers?

Un buffer es un archivo cargado en memoria. Puedes tener múltiples buffers abiertos.

### 10.2 Comandos de Buffer

Comando	Acción
:ls o :buffers	Listar buffers
:b {n}	Ir a buffer n
:b {nombre}	Ir a buffer por nombre (Tab complete)
:bn o :bnext	Siguiente buffer
:bp o :bprevious	Buffer anterior
:bf	Primer buffer
:bl	Último buffer
:bd	Eliminar buffer (cerrar archivo)
:e archivo	Editar archivo (crear buffer)
:badd archivo	Añadir archivo a buffer list

### 10.3 Navegación Rápida

```
Ctrl+^      " Alternar entre buffer actual y anterior
Ctrl+o      " Saltar a posición anterior
Ctrl+i      " Saltar a posición siguiente
```

## 11 Búsqueda y Reemplazo

### 11.1 Comando Substitute

**Sintaxis:**

```
: [rango]s/patrón/reemplazo/[flags]
```

### 11.2 Rangos

Rango	Descripción
%	Todo el archivo
.	Línea actual
\$	Última línea
'<, '>	Selección visual
{n}, {m}	Líneas n a m
. ,+5	Línea actual + 5 líneas
g/patrón/	Líneas que coinciden con patrón

### 11.3 Flags

Flag	Descripción
g	Global (todas las ocurrencias en línea)
c	Confirmar cada reemplazo
i	Case-insensitive
I	Case-sensitive
n	Reportar número de coincidencias

### 11.4 Ejemplos Prácticos

```
# Reemplazar en todo el archivo
:%s/foo/bar/g

# Reemplazar en líneas 10-20
:10,20s/old/new/g

# Reemplazar con confirmación
:%s/foo/bar/gc

# Reemplazar solo palabras completas
:%s/\<foo\>/bar/g

# Eliminar espacios al final de línea
:%s/\s\+$//g
```

```
# Añadir ; al final de líneas con texto
:%s/\(\.\+\)/\1;/g

# Reemplazar en selección visual
:'<,'>s/old/new/g

# Case-insensitive
:%s/error/ERROR/gi

# Eliminar líneas vacías
:g/^$/d

# Eliminar líneas que contengan patrón
:g/DEBUG/d

# Mantener solo líneas que coincidan
:v/KEEP/d

# Duplicar líneas que contengan patrón
:g/TODO/t.

# Mover líneas que contengan patrón
:g/ERROR/m$
```

## 11.5 Caracteres Especiales en Reemplazo

Carácter	Significado
&	Texto coincidente completo
\0	Texto coincidente completo
\1-\9	Grupo capturado 1-9
\L	Siguiente en minúsculas
\U	Siguiente en mayúsculas
\l	Siguiente carácter en minúscula
\u	Siguiente carácter en mayúscula
\e	Fin de \U o \L

### Ejemplos:

```
# Convertir "lastName" a "last_name"
:%s/(\([a-z]\)\)([A-Z]\))/\1_\L\2/g

# Convertir a mayúsculas primera letra
:%s/\<\(\w\)\)/\u\1/g

# Invertir orden (swap)
:%s/\(.*\), \(.*\)}/\2 \1/g
```

## 12 Plegado (Folding)

### 12.1 ¿Qué es Folding?

Folding permite ocultar secciones de texto para enfocarse en otras partes.

### 12.2 Métodos de Plegado

```
:set foldmethod=manual      " Manual
:set foldmethod=indent     " Por indentación
:set foldmethod=syntax     " Por sintaxis
:set foldmethod=marker     " Por marcadores
:set foldmethod=expr       " Por expresión
```

### 12.3 Comandos de Plegado

Comando	Acción
zf{movimiento}	Crear fold (manual)
zf3j	Fold 3 líneas abajo
zfap	Fold párrafo
za	Toggle fold
zA	Toggle fold recursivamente
zo	Abrir fold
z0	Abrir fold recursivamente
zc	Cerrar fold
zC	Cerrar fold recursivamente
zd	Eliminar fold
zD	Eliminar folds recursivamente
zR	Abrir todos los folds
zM	Cerrar todos los folds
zj	Mover al siguiente fold
zk	Mover al fold anterior

### 12.4 Nivel de Plegado

```
:set foldlevel=0      " Cerrar todos
:set foldlevel=99     " Abrir todos
zm                  " Aumentar foldlevel (cerrar más)
zr                  " Disminuir foldlevel (abrir más)
```

## 13 Marcas y Saltos

### 13.1 Marcas (Marks)

Guardar posiciones en el archivo.

### 13.2 Establecer Marcas

```
m{a-z}      " Marca local al buffer
m{A-Z}      " Marca global (entre archivos)
```

### 13.3 Ir a Marcas

Comando	Acción
'{marca}	Ir a línea de marca
`{marca}	Ir a posición exacta
''	Ir a posición antes del último salto
--	Ir a posición exacta antes del salto
'..	Ir a última modificación
'.	Ir a posición exacta de última modificación
'^	Ir a posición donde se salió de Insert

### 13.4 Ver Marcas

```
:marks          " Ver todas las marcas
:marks abc    " Ver marcas específicas
```

### 13.5 Lista de Saltos (Jump List)

Vim mantiene historial de saltos.

Comando	Acción
Ctrl+o	Salto atrás
Ctrl+i o Tab	Salto adelante
:jumps	Ver lista de saltos

### 13.6 Lista de Cambios (Change List)

Comando	Acción
g;	Ir a cambio anterior
g,	Ir a siguiente cambio
:changes	Ver lista de cambios

## 14 Comandos Ex

### 14.1 Comandos de Archivo

```
:w           " Guardar
:w archivo   " Guardar como
:w!          " Forzar guardar
:wa          " Guardar todos
:q           " Salir
:q!          " Forzar salir
:wq o :x     " Guardar y salir
:qa          " Salir de todos
:qa!         " Forzar salir de todos
:e archivo    " Editar archivo
:e!          " Recargar archivo
:saveas archivo " Guardar como (cambiar nombre)
```

## 14.2 Comandos de Shell

```
:!comando    " Ejecutar comando shell
:r !comando  " Insertar output de comando
:w !comando  " Pasar buffer a comando
:%!comando   " Filtrar buffer por comando
```

### Ejemplos:

```
:!ls          " Listar archivos
:r !date      " Insertar fecha
:%!sort       " Ordenar líneas
:%!python -m json.tool  " Formatear JSON
```

## 14.3 Comandos de Ayuda

```
:help         " Ayuda general
:help comando " Ayuda de comando específico
:help 'opción' " Ayuda de opción
:helpgrep patrón " Buscar en ayuda
Ctrl+]        " Seguir enlace (en ayuda)
Ctrl+t        " Volver (en ayuda)
```

## 15 Configuración (.vimrc/init.vim)

### 15.1 Ubicaciones

#### Vim:

- Linux/Mac: `~/.vimrc`
- Windows: `~/_vimrc`

#### Neovim:

- Linux/Mac: `~/.config/nvim/init.vim` o `~/.config/nvim/init.lua`
- Windows: `~/AppData/Local/nvim/init.vim`

## 15.2 Configuración Básica

```
" =====
" CONFIGURACIÓN BÁSICA
" =====

" Deshabilitar compatibilidad con vi
set nocompatible

" Habilitar tipo de archivo y plugins
filetype plugin indent on

" Habilitar sintaxis
syntax enable

" Codificación
set encoding=utf-8
set fileencoding=utf-8

" =====
" INTERFAZ
" =====

" Números de línea
set number          " Mostrar números
set relativenumber " Números relativos

" Cursor
set cursorline      " Resaltar línea actual
set scrolloff=8     " Mantener 8 líneas visibles arriba/abajo

" Búsqueda
set incsearch        " Búsqueda incremental
set hlsearch         " Resaltar búsquedas
set ignorecase       " Ignorar mayúsculas en búsqueda
set smartcase        " Case-sensitive si hay mayúsculas

" Interfaz
set showcmd          " Mostrar comando en progreso
set showmode         " Mostrar modo actual
set wildmenu         " Autocompletado de comandos mejorado
set laststatus=2     " Siempre mostrar statusline
set ruler            " Mostrar posición del cursor

" Split behavior
set splitbelow        " Split horizontal abajo
set splithright       " Split vertical a la derecha
```

```
" =====
" EDICIÓN
" =====

" Indentación
set tabstop=4      " Ancho de tab
set shiftwidth=4    " Ancho de indentación
set expandtab       " Usar espacios en lugar de tabs
set smartindent     " Indentación inteligente
set autoindent      " Mantener indentación

" Wrap
set wrap            " Ajustar líneas largas
set linebreak        " Romper en palabras completas

" Clipboard
set clipboard=unnamedplus " Usar clipboard del sistema

" Backup y swap
set nobackup        " No crear backups
set noswapfile       " No crear swap files
" O guardarlos en directorio específico:
" set backupdir=~/vim/backup
" set directory=~/vim/swap

" Undo persistente
set undofile
set undodir=~/vim/undo

" Mouse
set mouse=a         " Habilitar mouse

" Tiempo de espera
set timeoutlen=500  " Tiempo para secuencias de teclas

" =====
" MAPEOS
" =====

" Líder key
let mapleader = " "      " Usar espacio como líder
let maplocalleader = "," " Líder local

" Guardar con Ctrl+S
nnoremap <C-s> :w<CR>
inoremap <C-s> <Esc>:w<CR>
```

```
" Salir de Insert con jj
inoremap jj <Esc>

" Navegar entre ventanas más fácil
nnoremap <C-h> <C-w>h
nnoremap <C-j> <C-w>j
nnoremap <C-k> <C-w>k
nnoremap <C-l> <C-w>l

" Mover líneas
nnoremap <A-j> :m .+1<CR>==
nnoremap <A-k> :m .-2<CR>==
vnoremap <A-j> :m '>+1<CR>gv=gv
vnoremap <A-k> :m '<-2<CR>gv=gv

" Mantener selección al indentar
vnoremap < >gv
vnoremap > >gv

" Quitar highlight de búsqueda
nnoremap <leader><space> :nohlsearch<CR>

" Split rápido
nnoremap <leader>v :vsplit<CR>
nnoremap <leader>h :split<CR>

" Buffer navigation
nnoremap <leader>bn :bnext<CR>
nnoremap <leader>bp :bprevious<CR>
nnoremap <leader>bd :bdelete<CR>

" =====
" COLORES
" =====

" Tema (requiere instalación)
colorscheme desert " Tema por defecto

" True color
if has('termguicolors')
    set termguicolors
endif

" =====
" AUTOCOMANDOS
" =====
```

```

augroup vimrc
    autocmd!
    " Volver a última posición al abrir archivo
    autocmd BufReadPost * if line("'"") > 1 && line("'"") <= line("$") | exe "norma

    " Eliminar espacios al final al guardar
    autocmd BufWritePre * :%s/\s\+$//e

    " Resaltar yank
    autocmd TextYankPost * silent! lua vim.highlight.on_yank()
augroup END

" =====
" FUNCIONES PERSONALIZADAS
" =====

" Toggle números relativos
function! NumberToggle()
    if(&relativenumber == 1)
        set norelativenumber
    else
        set relativenumber
    endif
endfunction
nnoremap <leader>n :call NumberToggle()

```

### 15.3 Opciones Útiles

```

" Persistencia
set hidden          " Permitir buffers ocultos con cambios

" Rendimiento
set lazyredraw      " No redibujar durante macros
set updatetime=300   " Tiempo de espera para swap/CursorHold

" Signos
set signcolumn=yes  " Siempre mostrar columna de signos

" Formato
set textwidth=80     " Ancho de texto
set colorcolumn=80    " Línea visual en columna 80

" Completion
set completeopt=menuone,noselect,noinsert

" Diff
set diffopt+=vertical " Diff vertical

```

## 16 Plugins

### 16.1 Gestores de Plugins

#### 16.2 vim-plug (Recomendado)

**Instalación:**

```
# Vim
curl -fLo ~/.vim/autoload/plug.vim --create-dirs \
      https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim

# Neovim
sh -c 'curl -fLo "${XDG_DATA_HOME:-$HOME/.local/share}"/nvim/site/autoload/plug.vim \
      https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim'
```

**Uso en .vimrc:**

```
call plug#begin('~/vim/plugged')

" Plugins aquí
Plug 'tpope/vim-surround'
Plug 'junegunn/fzf.vim'
Plug 'preservim/nerdtree'

call plug#end()
```

**Comandos:**

```
:PlugInstall    " Instalar plugins
:PlugUpdate     " Actualizar plugins
:PlugClean      " Eliminar plugins no usados
:PlugStatus     " Ver estado
```

### 16.3 Packer (Neovim + Lua)

```
-- ~/.config/nvim/lua/plugins.lua
return require('packer').startup(function(use)
    use 'wbthomason/packer.nvim'
    use 'tpope/vim-surround'
    use 'nvim-telescope/telescope.nvim'
end)
```

### 16.4 Plugins Esenciales

#### 16.5 1. Navegación de Archivos

**NERDTree:**

```
Plug 'preservim/nerdtree'

" Mapeo
nnoremap <C-n> :NERDTreeToggle<CR>
```

**fzf.vim:**

```
Plug 'junegunn/fzf', { 'do': { -> fzf#install() } }
Plug 'junegunn/fzf.vim'

" Mapeos
nnoremap <C-p> :Files<CR>
nnoremap <leader>b :Buffers<CR>
nnoremap <leader>g :Rg<CR>
```

**16.6 2. Edición****vim-surround:**

```
Plug 'tpope/vim-surround'

" Comandos:
" ysiw" - Rodear palabra con "
" cs'" - Cambiar " por '
" ds" - Eliminar "
" yss) - Rodear línea con ()
```

**vim-commentary:**

```
Plug 'tpope/vim-commentary'

" Comandos:
" gcc - Comentar/descomentar línea
" gc{movimiento} - Comentar movimiento
" gcap - Comentar párrafo
```

**vim-repeat:**

```
Plug 'tpope/vim-repeat'
" Permite usar . con plugins
```

**16.7 3. Git****vim-fugitive:**

```
Plug 'tpope/vim-fugitive'

" Comandos:
:Git      " Git status
:Git add %
:Git commit
:Git push
:Gvdiffsplit " Ver diff
```

**gitsigns (Neovim):**

```
Plug 'lewis6991/gitsigns.nvim'
```

**16.8 4. Temas**

```
Plug 'gruvbox-community/gruvbox'
Plug 'morhetz/gruvbox'
Plug 'dracula/vim'
Plug 'joshdick/onedark.vim'
Plug 'arcticicestudio/nord-vim'

colorscheme gruvbox
```

**16.9 5. Statusline****vim-airline:**

```
Plug 'vim-airline/vim-airline'
Plug 'vim-airline/vim-airline-themes'

let g:airline_theme='gruvbox'
let g:airline_powerline_fonts = 1
```

**lualine (Neovim):**

```
use {
    'nvim-lualine/lualine.nvim',
    requires = { 'kyazdani42/nvim-web-devicons' }
}
```

**17 Neovim - Características Especiales****17.1 Diferencias con Vim****Arquitectura:**

- API asíncrona
- Cliente-servidor

- Soporte para UIs remotas
- Jobs y canales nativos

### Características:

- LSP integrado
- Tree-sitter (parsing AST)
- Lua como lenguaje de primera clase
- Mejor rendimiento
- API moderna

## 17.2 Configuración en Lua

### init.lua:

```
-- ~/.config/nvim/init.lua

-- Opciones básicas
vim.opt.number = true
vim.opt.relativenumber = true
vim.opt.expandtab = true
vim.opt.tabstop = 4
vim.opt.shiftwidth = 4
vim.opt.smartindent = true
vim.opt.wrap = false
vim.opt.swapfile = false
vim.opt.backup = false
vim.opt.undofile = true
vim.opt.hlsearch = false
vim.opt.incsearch = true
vim.opt.termguicolors = true
vim.opt.scrolloff = 8
vim.opt.signcolumn = "yes"
vim.opt.updatetime = 50
vim.opt.colorcolumn = "80"

-- Leader key
vim.g.mapleader = " "
vim.g.maplocalleader = ","

-- Mapeos
vim.keymap.set("n", "<leader>pv", vim.cmd.Ex)
vim.keymap.set("n", "<C-s>", ":w<CR>")
vim.keymap.set("i", "jj", "<Esc>")

-- Mover líneas
vim.keymap.set("v", "J", ":m '>+1<CR>gv=gv")
vim.keymap.set("v", "K", ":m '<-2<CR>gv=gv")
```

```
-- Mantener cursor centrado
vim.keymap.set("n", "<C-d>", "<C-d>zz")
vim.keymap.set("n", "<C-u>", "<C-u>zz")
vim.keymap.set("n", "n", "nzzzv")
vim.keymap.set("n", "N", "Nzzzv")

-- Navegar ventanas
vim.keymap.set("n", "<C-h>", "<C-w>h")
vim.keymap.set("n", "<C-j>", "<C-w>j")
vim.keymap.set("n", "<C-k>", "<C-w>k")
vim.keymap.set("n", "<C-l>", "<C-w>l")

-- Resaltar yank
vim.api.nvim_create_autocmd("TextYankPost", {
    group = vim.api.nvim_create_augroup("highlight_yank", { clear = true }),
    callback = function()
        vim.highlight.on_yank()
    end,
})
```

### 17.3 Tree-sitter

Parser incremental para mejor resaltado de sintaxis.

```
use {
    'nvim-treesitter/nvim-treesitter',
    run = ':TSUpdate'
}

require'nvim-treesitter.configs'.setup {
    ensure_installed = { "python", "javascript", "lua", "vim" },
    highlight = {
        enable = true,
    },
}
```

### 17.4 Telescope (Fuzzy Finder)

```
use {
    'nvim-telescope/telescope.nvim',
    requires = { 'nvim-lua/plenary.nvim' }
}

-- Mapeos
vim.keymap.set('n', '<leader>ff', ':Telescope find_files<CR>')
vim.keymap.set('n', '<leader>fg', ':Telescope live_grep<CR>')
vim.keymap.set('n', '<leader>fb', ':Telescope buffers<CR>')
```

## 18 LSP y Autocompletado

### 18.1 LSP en Neovim

**Instalar nvim-lspconfig:**

```
use 'neovim/nvim-lspconfig'

-- Configurar servidor
local lspconfig = require('lspconfig')

-- Python
lspconfig.pyright.setup{}

-- JavaScript/TypeScript
lspconfig.tsserver.setup{}

-- Lua
lspconfig.lua_ls.setup{
    settings = {
        Lua = {
            diagnostics = {
                globals = { 'vim' }
            }
        }
    }
}
```

### 18.2 Mapeos LSP

```
-- Después de attach LSP
local opts = { noremap=true, silent=true }
vim.keymap.set('n', 'gD', vim.lsp.buf.declaration, opts)
vim.keymap.set('n', 'gd', vim.lsp.buf.definition, opts)
vim.keymap.set('n', 'K', vim.lsp.buf.hover, opts)
vim.keymap.set('n', 'gi', vim.lsp.buf.implementation, opts)
vim.keymap.set('n', '<C-k>', vim.lsp.buf.signature_help, opts)
vim.keymap.set('n', '<leader>rn', vim.lsp.buf.rename, opts)
vim.keymap.set('n', '<leader>ca', vim.lsp.buf.code_action, opts)
vim.keymap.set('n', 'gr', vim.lsp.buf.references, opts)
```

### 18.3 nvim-cmp (Autocompletado)

```
use 'hrsh7th/nvim-cmp'
use 'hrsh7th/cmp-nvim-lsp'
use 'hrsh7th/cmp-buffer'
use 'hrsh7th/cmp-path'
```

```

use 'L3MON4D3/LuaSnip'

local cmp = require'cmp'

cmp.setup({
    snippet = {
        expand = function(args)
            require('luasnip').lsp_expand(args.body)
        end,
    },
    mapping = {
        ['<C-p>'] = cmp.mapping.select_prev_item(),
        ['<C-n>'] = cmp.mapping.select_next_item(),
        ['<C-d>'] = cmp.mapping.scroll_docs(-4),
        ['<C-f>'] = cmp.mapping.scroll_docs(4),
        ['<C-Space>'] = cmp.mapping.complete(),
        ['<C-e>'] = cmp.mapping.close(),
        ['<CR>'] = cmp.mapping.confirm({ select = true }),
    },
    sources = {
        { name = 'nvim_lsp' },
        { name = 'buffer' },
        { name = 'path' },
    }
})

```

## 19 Vimscript y Lua

### 19.1 Vimscript Básico

#### Variables:

```

let variable = "valor"
let g:global_var = 1
let l:local_var = 2
let s:script_var = 3

```

#### Condicionales:

```

if condición
    " código
elseif otra_condición
    " código
else
    " código
endif

```

#### Loops:

```
for item in lista
    " código
endfor

while condición
    " código
endwhile
```

**Funciones:**

```
function! MiFuncion(arg1, arg2)
    " código
    return resultado
endfunction

" Llamar
call MiFuncion(1, 2)
```

**19.2 Lua Básico****Variables:**

```
local variable = "valor"
vim.g.global_var = 1
```

**Condicionales:**

```
if condición then
    -- código
elseif otra_condición then
    -- código
else
    -- código
end
```

**Loops:**

```
for i = 1, 10 do
    -- código
end

for key, value in pairs(tabla) do
    -- código
end
```

**Funciones:**

```
local function mi_funcion(arg1, arg2)
    -- código
    return resultado
end
```

## 20 Trucos Avanzados

### 20.1 1. Edición de Múltiples Archivos

**Argumentos:**

```
:args *.js          " Cargar todos los .js
:argdo %s/old/new/g | update " Reemplazar en todos
```

**Buffers:**

```
:bufdo %s/old/new/g | update
```

**Quickfix:**

```
:vimgrep /patrón/ **/*.py      " Buscar en archivos
:copen                         " Abrir quickfix list
:cn                            " Siguiente
:cp                            " Anterior
```

### 20.2 2. Edición Avanzada con Expresiones

**Incrementar números:**

```
Ctrl+a      " Incrementar número bajo cursor
Ctrl+x      " Decrementar

# En visual block:
g Ctrl+a    " Incrementar secuencialmente
```

**Evaluar expresiones:**

```
# En insert mode:
Ctrl+r =2+2<CR>  " Inserta resultado (4)
```

### 20.3 3. Cambiar en Múltiples Líneas

**Visual Block:**

1. Ctrl+v (visual block)
2. Seleccionar líneas
3. I (insertar al inicio) o A (al final)
4. Escribir
5. Esc (se aplica a todas)

## 20.4 4. Formatear Código

**Formatear párrafo:**

```
gqap      " Format paragraph
gwap      " Format and keep cursor position
```

**External formatter:**

```
:%!python -m json.tool      " JSON
:!xmllint --format -       " XML
:% !prettier --parser babel " JavaScript
```

## 20.5 5. Diff Files

```
:vert diffsplits archivo2
:diffthis      " Marcar ventana actual para diff
:diffoff       " Desactivar diff
]c            " Siguiente cambio
[c            " Cambio anterior
do            " Diff obtain (obtener cambio)
dp            " Diff put (poner cambio)
```

## 20.6 6. Sesiones

**Guardar sesión:**

```
:mksession ~/sesion.vim
```

**Cargar sesión:**

```
:source ~/sesion.vim
# O al iniciar:
vim -S ~/sesion.vim
```

## 20.7 7. Spell Checking

```
:set spell spelllang=es      " Activar corrector español
:set spell spelllang=en      " Inglés

]s      " Siguiente error
[s      " Error anterior
z=      " Sugerencias
zg     " Añadir palabra a diccionario
zw     " Marcar palabra como mal escrita
```

## 20.8 8. Abrir URL

```
gx      " Abrir URL bajo cursor en navegador
```

## 20.9 9. Ejecutar Línea como Comando

```
:.!sh    " Ejecutar línea actual como comando shell
```

## 20.10 10. Formateo de Tabla

```
# Seleccionar tabla en visual
:!column -t      " Alinear columnas
```

# 21 Solución de Problemas

## 21.1 Problemas Comunes

### 21.2 1. Vim/Neovim Lento

**Soluciones:**

```
" Deshabilitar plugins temporalmente
vim --noplugin

" Ver tiempo de inicio
vim --startuptime tiempo.log

" Lazy loading de plugins
Plug 'plugin/name', { 'on': 'Comando' }

" Reducir updatetime
set updatetime=300
```

### 21.3 2. Clipboard No Funciona

**Verificar soporte:**

```
:echo has('clipboard')  " Debe retornar 1
```

**Solución Ubuntu:**

```
sudo apt install vim-gtk3
# O para Neovim:
sudo apt install xclip
```

### 21.4 3. Colores Incorrectos

```
" Habilitar true color
set termguicolors

" O verificar $TERM
echo $TERM " Debe ser xterm-256color o similar
```

## 21.5 4. LSP No Funciona (Neovim)

**Verificar:**

```
:checkhealth          " Diagnóstico completo
:LspInfo             " Info de LSP

" Instalar servidor manualmente
:LspInstall pyright
```

## 21.6 5. Plugins No Se Instalan

```
" vim-plug
:PlugInstall
:PlugUpdate

" Verificar errores
:messages
```

## 21.7 Debugging

**Modo verbose:**

```
:set verbose=9        " Nivel de verbosidad
:set verbosefile=~/vim_debug.log
```

**Ver opciones:**

```
:set all              " Ver todas las opciones
:set opción?         " Ver valor de opción específica
```

**Ver mapeos:**

```
:map                  " Ver todos los mapeos
:nmap                " Mapeos en modo Normal
:imap                " Mapeos en modo Insert
```

## 22 Referencia Rápida

### 22.1 Atajos Esenciales

#### Navegación:

h/j/k/l	Básico
w/b	Palabra adelante/atrás
0/\$	Inicio/fin de línea
gg/G	Inicio/fin de archivo
Ctrl+d/u	Media página
f/F{char}	Find carácter

#### Edición:

i/a	Insert/append
o/O	Nueva línea abajo/arriba
x/dd	Eliminar carácter/línea
yy/p	Copiar/pegar
u/Ctrl+r	Undo/redo
.	Repetir

#### Visual:

v/V/Ctrl+v	Visual char/line/block
y/d/c	Copiar/eliminar/cambiar
>/<	Indentar/desindentar

#### Búsqueda:

/texto	Buscar
n/N	Siguiente/anterior
*/#	Buscar palabra bajo cursor
:s/old/new/g	Reemplazar

#### Archivos:

:w	Guardar
:q	Salir
:e archivo	Abrir
:bn/:bp	Buffer siguiente/anterior

#### Ventanas:

:sp/:vsp	Split
Ctrl+w h/j/k/l	Navegar
Ctrl+w =	Igualar tamaños

## 22.2 Comandos por Modo

### Normal Mode:

- Navegación: h, j, k, l, w, b, 0, \$, gg, G
- Edición: i, a, o, x, dd, yy, p
- Visual: v, V, Ctrl+v
- Búsqueda: /, ?, \*, #, n, N
- Comandos: :

### Insert Mode:

- Salir: Esc, Ctrl+[
- Autocompletado: Ctrl+n/p, Ctrl+x Ctrl+o
- Insertar: Ctrl+r{registro}

### Visual Mode:

- Operadores: y, d, c, >, <, =
- Cambiar modo: v, V, Ctrl+v
- Objetos: aw, iw, ap, i(), a"

### Command Mode:

- Archivo: :w, :q, :e, :wq
- Búsqueda: /, ?
- Rango: :%s, :., :'^<, '^>

## 23 Conclusión

### 23.1 El Viaje de Aprendizaje

#### Nivel 1 - Básicos (Semana 1):

- Modos (Normal, Insert)
- Navegación básica (hjkl)
- Guardar y salir (:wq)
- Insert/append (i, a, o)

#### Nivel 2 - Movimientos (Mes 1):

- Movimientos de palabra (w, b, e)
- Línea (0, \$, ^)
- Búsqueda (/?, \*, #)
- Objetos de texto (iw, aw)

#### Nivel 3 - Operadores (Meses 2-3):

- Eliminar/cambiar/copiar (d, c, y)
- Composición (dw, ciw, yap)
- Visual mode (v, V, Ctrl+v)
- Registros básicos

**Nivel 4 - Avanzado (Meses 4-6):**

- Macros (q, @)
- Múltiples buffers/ventanas
- Búsqueda y reemplazo avanzado
- Configuración personalizada

**Nivel 5 - Maestro (6+ meses):**

- Plugins y ecosistema
- Vimscript/Lua
- LSP y autocompletado
- Workflow completamente personalizado

## 23.2 Recursos

**Tutoriales:**

- `vimtutor` (incluido con Vim)
- Vim Adventures (juego)
- Openvim.com
- vim.fandom.com

**Documentación:**

- `:help` (ayuda integrada)
- [vimhelp.org](http://vimhelp.org)
- [neovim.io/doc](https://neovim.io/doc)

**Comunidad:**

- r/vim, r/neovim (Reddit)
- vi.stackexchange.com
- GitHub Discussions

## 23.3 Tips Finales

1. **Practica diariamente:** 15-30 minutos al día
2. **Aprende incrementalmente:** No trates de aprender todo a la vez
3. **Usa vimtutor:** El tutorial interactivo es excelente
4. **Evita plugins al inicio:** Aprende Vim vanilla primero
5. **Configura gradualmente:** Añade opciones según las necesites
6. **Usa el help:** `:help {tema}` es tu mejor amigo
7. **Sé paciente:** La curva de aprendizaje vale la pena

## 24 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. [!\[\]\(119f8b04176017b7e15543e5f2e9d485\_img.jpg\) Comandos De Informacion Windows](#)
2. [!\[\]\(99f30c3f0602b7fcb6ffe93a8a09d05a\_img.jpg\) Adb](#)
3. [!\[\]\(a4654b1bfdea74ca11550ff7d966a303\_img.jpg\) Limpieza Y Optimizacion De Pc](#)
4. [!\[\]\(cc4fcfb0f6386d14bd2bfa60f10e6989\_img.jpg\) Usando Apk En Windown 11](#)
5. [!\[\]\(bf5701c37eda62fa8cd7d14fc12a6070\_img.jpg\) Gestionar Versiones De Jdk En Kubuntu](#)
6. [!\[\]\(d82375c765d550a493e659a6a0d4ebc5\_img.jpg\) Instalar Tor Browser](#)
7. [!\[\]\(d85c76490f26ba7ec9bfa7079fd66cc4\_img.jpg\) Crear Enlaces Duros O Hard Link En Linux](#)
8. [!\[\]\(ad04849580bcb3667ae535011db4afcf\_img.jpg\) Comandos Vim](#)
9. [!\[\]\(c2c851456d638fc0cbe4f287d70dcb8d\_img.jpg\) Guia De Git Y Github](#)
10. [!\[\]\(a3633f907b6131f73a1974b8a4a5571c\_img.jpg\) 00 Primeros Pasos En Linux](#)
11. [!\[\]\(2af16d91814c6341e6a7dd08ff8b510b\_img.jpg\) 01 Introduccion Linux](#)
12. [!\[\]\(34f3df322df1403f66172b51b287f811\_img.jpg\) 02 Distribuciones Linux](#)
13. [!\[\]\(b529fc1ce957fbe7617745a30a0ffc8f\_img.jpg\) 03 Instalacion Linux](#)
14. [!\[\]\(da7a77c49b8460b9605c58db1eb76f2a\_img.jpg\) 04 Administracion Particiones Volumenes](#)
15. [!\[\]\(92133e31b98a541f79f52472c3cfe83a\_img.jpg\) Atajos De Teclado Y Comandos Para Usar Vim](#)
16. [!\[\]\(d4fa043ac5b0f5a0e83a467ed274909f\_img.jpg\) Instalando Specitify](#)
17. [!\[\]\(5af6b80811bebdbe007efcf4da7c9710\_img.jpg\) Gestiona Tus Dotfiles Con Gnu Stow](#)

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