Editores de Texto en Linux

Guía completa para administradores de sistemas Linux

nano

vi/vim

Configuración

¿Por qué usar editores de texto en Linux?

Configuración de sistemas y servicios

Edición de archivos críticos del sistema como /etc/, /usr/local/, etc.

Edición de scripts y archivos de configuración

Shell scripts, Python, Perl, archivos de configuración de aplicaciones.

Entornos sin interfaz gráfica

Servidores remotos, SSH, entornos de producción sin GUI.

Herramienta estándar de administración

Universalmente disponible en todas las distribuciones de Linux y Unix.

```
root@server:~

[root@server ~]# nano /etc/hosts
# Editando archivo de configuración del sistema...

[root@server ~]# vim script.sh
# Modificando script de automatización...

[root@server ~]# :wq
# Guardando cambios y saliendo...
```

Indispensable para sysadmins
Domine estas herramientas para Linux

Editores de texto habituales en Linux

nano
Terminal

Amigable y fácil de usar, ideal para principiantes

vi/vim
Estándar

Editor universal de Unix, muy potente y versátil

emacs
Extensible

Potente editor con múltiples funciones y personalizaciones

ed
Líneas

Editor de líneas básico, muy ligero y rápido

gedit
GUI

Entorno gráfico de GNOME, intuitivo y amigable

kate
GUI

Entorno gráfico de KDE, potente y versátil

Disponibilidad universal

Todos los editores de terminal están disponibles en todas las distribuciones de Linux

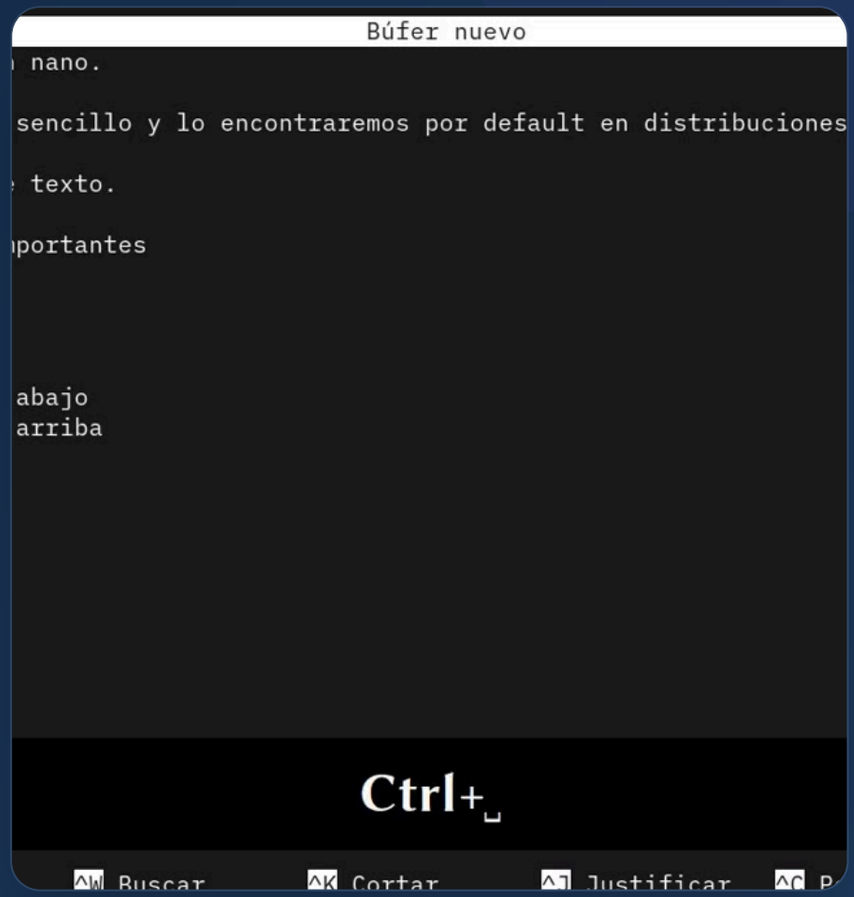
El editor Nano

Editor amigable y fácil de usar

Nano es un editor de texto minimalista y accesible, ideal para principiantes y usuarios que buscan una interfaz sencilla sin menús complejos.

Comandos principales

Ctrl+G Ayuda	**Ctrl+X** Salir
Ctrl+O Guardar	**Ctrl+R** Leer archivo
Ctrl+W Buscar	**Ctrl+Y** Pág anterior
Ctrl+V Pág siguiente	**Ctrl+K** Cortar línea
Ctrl+U Pegar	**Ctrl+L** Refrescar



Los comandos se muestran en la parte inferior de la pantalla

El editor universal vi y derivados

Editor estándar de Unix

Creado en 1976 por Bill Joy, vi es el editor de texto clásico y universal de sistemas Unix y Linux.

- Orientado a pantalla
- Multipropósito
- Rápido y eficiente
- Universalmente disponible

Derivados principales

vim
Vi Improved

Versión mejorada con más características y plugins

elvis
Clone de vi

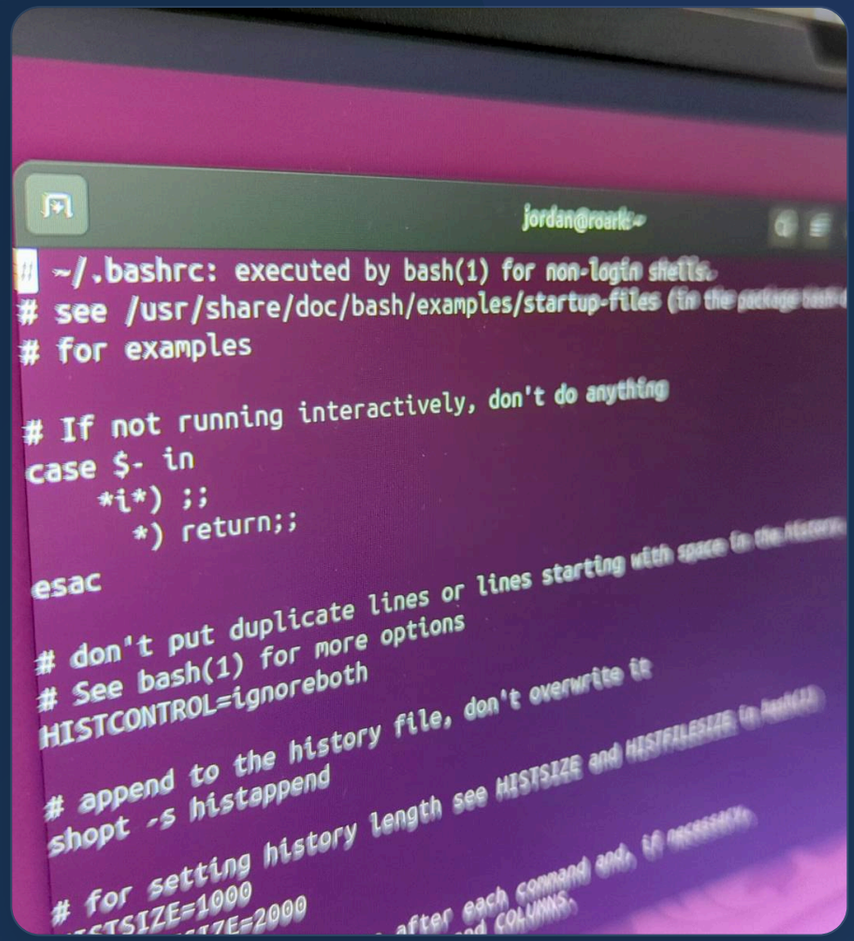
Implementación libre compatible con vi

nvi
New vi

Versión BSD con soporte multi-ventana

vim-tiny
Versión compacta

Versión ligera para sistemas pequeños



vim es el derivado más utilizado y extendido

Modos de vi

Modo Comando

Inicial

Navegación y ejecución de comandos

Teclas: h, j, k, l, dd, yy, G

Presionar i, a, o...

Modo Edición

Insert

Edición de texto

Teclas: i, a, I, A, o, O

Presionar Esc

Modo Ex

Última Línea

Comandos con dos puntos

Teclas: :, /, ?

Presionar Esc

Entrar a edición

Desde modo comando, presiona **i** (insertar), **a** (agregar), **o** (línea)

Salir de edición

Presiona **Esc** para volver al modo comando desde cualquier modo

Modo ex

Desde modo comando, presiona **:** para acceder a comandos de última línea

Cambio entre modos y comandos de edición

Entrar en modo edición

Comando	Descripción
`i`	Insertar antes del cursor
`a`	Insertar después del cursor
`I`	Insertar al inicio de línea
`A`	Insertar al final de línea
`o`	Abrir línea debajo del cursor
`O`	Abrir línea encima del cursor

Salir del modo edición

Esc — **Tecla Escape**

Vuelve al modo comando desde cualquier modo

Si no estás seguro del modo en el que estás, presiona Esc para asegurarte de estar en modo comando

Ejemplo práctico

1. Presiona `i` para entrar en modo edición
2. Escribe el texto que necesitas
3. Presiona **`Esc`** para salir del modo edición
4. Ahora puedes usar comandos de modo comando

Comandos de desplazamiento en vi

Básico

`h`	Moverse un carácter a la izquierda
`j`	Moverse una línea hacia abajo
`k`	Moverse una línea hacia arriba
`l`	Moverse un carácter a la derecha

Por palabras

`w`	Siguiente palabra
`b`	Palabra anterior
`e`	Final de palabra

Por líneas

`0`	Inicio de línea
`$`	Final de línea

Por archivo

`G`	Ir a última línea
`1G`	Ir a primera línea
`nG`	Ir a línea n

Por pantalla

`Ctrl+F`	Página adelante
`Ctrl+B`	Página atrás
`H`	Parte superior
`M`	Centro de pantalla
`L`	Parte inferior

Comandos de búsqueda, reemplazo y copia (modo ex)

Búsqueda

`/texto`	Buscar hacia adelante
`?texto`	Buscar hacia atrás
`n`	Siguiente coincidencia
`N`	Coincidencia anterior

Reemplazo

`:s/antiguo/nuevo`	Primera ocurrencia en línea
`:s/antiguo/nuevo/g`	Todas en línea
`:%s/antiguo/nuevo/g`	Todo el archivo
`:n,m d`	Eliminar líneas n-m
`:n,m co m`	Copiar líneas n-m después de m

Copiar y Pegar

`yy`	Copiar línea (yank)
`Y`	Copiar línea
`p`	Pegar después
`P`	Pegar antes

Ejemplo de uso

- **`:%s/texto/nuevo_texto/g`** reemplaza todas las ocurrencias
- **`:5,10 d`** elimina las líneas 5 a 10
- **`yy`** luego **`p`** para duplicar líneas

Configuración de .vimrc

Comandos de configuración

`set nu`	Mostrar números de línea
`set nonu`	Ocultar números de línea
`set ic`	Ignorar mayúsculas en búsqueda
`set noic`	Distinguir mayúsculas en búsqueda
`syntax on`	Activar resaltado de sintaxis
`set tabstop=4`	Tamaño de tabulación
`set expandtab`	Usar espacios en lugar de tabs
`set number`	Mostrar números de línea

Ubicación del archivo

~/.vimrc

Directorio home del usuario

El archivo **`.vimrc`** contiene la configuración personalizada de vim. Se carga automáticamente al iniciar el editor.

Ejemplo de archivo

```
" Activar números de línea
set number
" Ignorar mayúsculas en búsqueda
set ignorecase
" Resaltado de sintaxis
syntax on
" Configurar tabulación
```

Preguntas de evaluación

1. ¿Cuál es el comando para guardar cambios en nano?
2. ¿Qué tecla se usa para salir del modo edición en vi?
3. ¿En qué modo se ejecutan comandos que comienzan con dos puntos en vi?
4. ¿Cuál es el comando para copiar la línea actual en vi?
5. ¿Qué comando de nano se usa para buscar texto?
6. ¿Cuál es el comando para ir a la última línea del archivo en vi?
7. ¿Qué comando de modo ex se usa para reemplazar texto en todo el archivo?
8. ¿Dónde se encuentra el archivo de configuración .vimrc?
9. ¿Qué comando de nano se usa para cortar una línea?
10. ¿Cuál es el comando para mostrar números de línea en vi?