

Uso básico del editor de texto Nano

Miguel Ortuño
Escuela de Ingeniería de Fuenlabrada
Universidad Rey Juan Carlos

Septiembre de 2024



© 2024 Miguel Angel Ortuño Pérez.
Algunos derechos reservados. Este documento se distribuye bajo la
licencia *Atribución-CompartirIgual 4.0 Internacional* de Creative
Commons, disponible en
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

El editor de texto Nano

Nano es un editor de texto para usar desde el terminal, libre y gratuito. Lo usarás para editar ficheros en tu cuenta del laboratorio

- Está disponible para Linux, macOS y Windows. Pero para trabajar en tu ordenador seguramente no merece la pena instalarlo. De momento puedes usar el editor de texto preestablecido en tu equipo
- Es sencillo y suficiente para tareas básicas. Algunas cosas las denomina de forma un poco distinta a los programas modernos, pero es fácil de usar

Uso de Nano

Para empezar a editar un fichero, teclea en el terminal

```
nano nombre_del_fichero.txt
```

o

```
nano nombre_del_fichero.pas
```

- Si el fichero existía, lo abre. Si no, crea uno nuevo
- En las dos últimas líneas del terminal, Nano nos irá mostrando los atajos de teclado que necesitaremos normalmente
- El acento circunflejo (^) significa Ctrl
- M- significa *tecla meta*
 - En Windows y Linux, pulsar Alt, mantener pulsado y pulsar la letra que corresponda
 - En macOS, pulsar Esc, soltar y pulsa la letra que corresponda
- Los atajos aparecen escritos en mayúsculas, pero es indiferente usar mayúsculas o minúsculas

Funcionalidad principal

- ^O Write out
Guarda el fichero. Equivalente al Save o Guardar de otros programas

Seleccionar texto

- 1 Llevamos el cursor al principio del texto que queremos seleccionar
- 2 Pulsamos *shift* y lo mantenemos pulsado
- 3 Llevamos el cursor al final del texto a seleccionar. La posición actual del cursor quedará excluida de la selección

- ^K Cut text

Cortar texto. Por omisión, una línea. Si hemos seleccionado algo, copia la selección

- ^U Paste text

Pega el texto cortado o copiado

- Copiar

Para no perder el texto original, esto es, copiar, podemos hacer dos cosas:

- 1 Cortar, pegar lo que acabamos de cortar, ir donde queramos pegar, pegar
- 2 M-6 Esto es,
Alt 6 (Linux, Windows)
Esc 6 (macOS)

- M-U (Alt u) Undo
Deshace el último cambio
- M-E (Alt e) Redo
Rehace lo último que hayamos deshecho
- ^W Where is
Busca una cadena de texto. Equivalente al Search o Buscar de otros programas
- ^X Exit
Sale del programa. Si hay alguna modificación que no hemos guardado, nos preguntará si queremos guardar las modificaciones (yes/no) o si ya no queremos salir (cancel)

En este vídeo puedes ver una demostración

<https://youtu.be/U3WBWZHvf7Q>

Observa que usamos dos formas distintas de copiar y pegar (o de cortar y pegar)

- Normalmente trabajaremos dentro del mismo fichero y podremos usar las funciones del propio Nano

| | | |
|-----|----|--|
| M-6 | ^U | (Copiar y pegar. Alt 6 en Windows y Linux, Esc 6 en macOS) |
| ^K | ^U | (Cortar y pegar) |

- Cuando queramos copiar y pegar o cortar y pegar entre dos ficheros, será necesario hacer algo diferente. Aquí recomendamos copiar y pegar usando el entorno gráfico de nuestro ordenador local (botón secundario del ratón)¹

¹El inconveniente es que esto será ligeramente distinto en Windows, Linux o macOS

Copiar y pegar con el ratón no siempre es posible. Supongamos que quieres copiar y pegar las directivas, sin ratón:

- 1 Descarga este holamundo en la carpeta donde necesites el fichero. Para ello, haz clic con el botón derecho en el siguiente enlace, en la opción *guardar enlace como*
<http://ortuno.es/holamundo.pas>
- 2 Abre el fichero con nano. Quédate con las directivas, modifica lo que necesites.
- 3 Guarda el fichero (^O), y cuando nano te muestre el nombre del fichero a escribir, cámbialo por el nombre adecuado.

Ver línea y columna en Nano

El compilador te dirá en qué línea y columna están los errores, así que te será útil que Nano te muestre la línea y la columna donde está el cursor

- Edita con el propio Nano un fichero llamado *.nanorc* en tu *home*
`nano ~/.nanorc`
- Dentro escribe el siguiente texto
`set constantshow`

Observa que el nombre de este fichero empieza por punto. Esto hace que sea un fichero *oculto*. Podrás editarlo como cualquier otro, pero

- Al hacer un listado normal con `ls`, este fichero no se verá
- Cuando quieras ver todos los ficheros del directorio actual, incluyendo los ocultos, deberás añadir la opción `-a`
`ls -a`

Color basado en la sintaxis

Para que Nano use distintos colores para los distintos elementos de un programa Pascal (esto es, de los ficheros .pas)

- 1 Descarga el fichero `http://ortuno.es/pascal.nanorc` en el directorio `~/fpi/`

Esto puedes hacerlo de dos formas

- Editar un fichero `~/fpi/pascal.nanorc`, copiar el contenido del fichero desde el navegador, pegar en Nano
- `cd ~/fpi`
`wget http://ortuno.es/pascal.nanorc`

- 2 Añade en el fichero de configuración de Nano (`~/nanorc`) la siguiente línea

```
include ~/fpi/pascal.nanorc
```

Como siempre, estas instrucciones debes seguirlas al pie de la letra. Una mayúscula o un espacio mal puestos provocarán que nada funcione

Indentado en Nano

- Activar y desactivar la visualización de caracteres invisibles
M-P

Esto es:

- Windows:
alt p
- macOS:
esc p
- Nano representa los tabuladores con un ancho por omisión de 8 caracteres. Si preferimos, otro valor (normalmente 4), añadimos la siguiente línea en `~/.nanorc`
`set tabsize 4`
- Para que Nano reemplace todos los tabuladores por espacios, añadimos la siguiente línea en `~/.nanorc`
`set tabstospaces`

Correspondencia de paréntesis

Es muy normal tener expresiones con muchos paréntesis, lo que resulta propenso a errores. Los editores suelen tener una opción que, señalando un paréntesis abierto, nos indica el paréntesis cerrado correspondiente. Y viceversa. En Nano:

- Llevamos el cursor a un paréntesis
- Pulsamos M-]

Esto es,

- Windows:
alt altgr +
- macOS:
esc altgr +

Para que la correspondencia de paréntesis funcione, es necesario editar

- O bien editar el fichero `~/ .nanorc`
- O bien el fichero `/etc/nanorc` (si somos los administradores de la máquina)

Y añadir / descomentar las líneas

```
set brackets "'>]}"  
set matchbrackets "(<[>]}"
```

(En el laboratorio esto ya está hecho, si instalas Nano en casa, deberás hacerlo tú)

Personalización automática

- El siguiente script realiza automáticamente todas las personalizaciones que hemos descrito
http://ortuno.es/custom_nano.py
- En el laboratorio puedes ejecutar en un terminal
`~mortuno/custom_nano.py`²

²Recuerda que la virgulilla normalmente no se puede copiar y pegar desde un pdf