

Linux + IA 2026/05/13

1. Solo para **MINIOS**:

1. Antes de empezar modificaremos el manejo de la hora

```
1. sudo hwclock --systohc --utc
```

Comandos esenciales de Nano

El símbolo ^ representa la tecla **Ctrl**. M- representa **Alt**.

Abrir, guardar y salir

- nano <archivo> : Abre o crea un archivo.
- sudo nano <archivo> : Abre archivos del sistema (requiere permisos).
- nano +<número> <archivo> : Abre en una línea específica.
- ^O : Guarda el archivo (sin salir).
- ^X : Salir del editor. Pregunta si hay cambios sin guardar.

Comandos esenciales de vim

- Instalando:

```
◦ sudo apt update && sudo apt install vim
```

Abrir, guardar y salir

- Guardar y salir (dentro de vi, en modo normal)
- :w Guardar (sin salir)
- :q Salir (si no hay cambios sin guardar)
- :q! Salir forzado (descartar cambios)
- :wq Guardar y salir

Comandos editor de Wave

- Abrir un archivo:

```
◦ wsh edit NombreArchivo
```

Ollama

- Ruta de los archivos de los MLL

- `ls /usr/share/ollama/.ollama/models/blobs/`

Permisos

- CHMOD
 - Creamos el script

```
nano script.sh
```

- `script.sh`

```
echo "HOLA DESDE EL SCRIPT"
```

- permitir ejecutar un script:

```
chmod u+x script.sh
./script.sh
```

- restringir escritura a terceros: `chmod o-w archivo_confidencial.txt`
- proteger archivos sensibles: `chmod 700 script_privado.sh`

- CHOWN

- ```
sudo cp documento.txt /home/jared/
ls -l /home/jared/documento.txt
-rw-r--r-- 1 root root ...
Solución: cambiar dueño a tu usuario
sudo chown jared:jared /home/jared/documento.txt
Ahora: jared:jared
```

- UMASK

- ```
echo "mi diario publico" > diario.txt
# Ver sus permisos
ls -l diario.txt
# -rw-r--r-- 1 jared jared 20 may 13 10:00 diario.txt
# ¡Cualquiera en el sistema puede leer tu diario!
umask 0077
echo "mi diario secreto" > diario_privado.txt
```

Wave Integrando ollama

1. Editar el archivo ~/.config/waveterm/waveai.json

1. **nano** ~/.config/waveterm/waveai.json

2. Agragar

1. {
 "ollama-granite4": {
 "display:name": "Ollama - granite4 tiny-h",
 "display:order": 2,
 "display:icon": "microchip",
 "display:description": "Local Granite4 tiny-h via Ollama",
 "ai:apitype": "openai-chat",
 "ai:model": "granite4:tiny-h",
 "ai:thinkinglevel": "medium",
 "ai:endpoint": "http://localhost:11434/v1/chat/completions",
 "ai:apitoken": "ollama"
 }
}

3. Uso practico:

1. Buscar parametros de un comando:

```
wsh ai -m "parametros principales comando ls" -s
```

2. Mandar analizar una salida a IA:

1. **free -h** |wsh ai - -m "analiza el uso de memoria" -s

3. Verificar errores:

1. Causar un error

```
sudo timedatectl set-local-rtc 0
```

2. Analizar por la IA los errores

```
sudo journalctl -p 3 --since "1 hour ago" |wsh ai - -m "analiza los errores y responde en español" -s
```

From:

<https://dw.openit.dev/> - DokuWikiOpenIT

Permanent link:

<https://dw.openit.dev/lia20260513?rev=1778723014>

Last update: **2026/05/14 01:43**

