

LINUX BÁSICO Y PRIMEROS COMANDOS

HANDS-ON EN RASPBERRY PI 5

Equipo docente ClústerLab

4 de agosto de 2025

- Usaremos **una** Raspberry Pi por equipo, con **pantalla, teclado y mouse**.
- Objetivos del día:
 - Navegar el sistema de archivos y la terminal.
 - Instalar utilidades básicas.
 - Dejar la Pi *lista* para el día 2 (SSH y trabajo remoto).

```
1  # Usuario por defecto (si aplica a tu imagen)
2  user: pi    pass: raspberry
```

Nota

No activaremos SSH hoy; lo veremos al final y se usará desde el día 2.

```
1  /                # raíz
2  /home/pi         # tu carpeta
3  /etc             # configuración del
    ↪ sistema
4  /var/log         # logs
5  /opt             # software opcional
6  /tmp             # temporales
7  /boot            # arranque (Raspberry)
```



Nota

Carpetas “del sistema” (/etc, /var) vs
“tu espacio” (/home/pi).

```
1  $ ls -l /
2  $ tree -L 1 /home/pi # requiere 'tree'
```

COMANDOS ESENCIALES (AMPLIADO)

```
1 $ ls -lah           # tamaños humanos, muestra ocultos
2 $ cd ..            # subir un nivel
3 $ mv a b           # renombrar o mover
4 $ cp a b           # copiar
5 $ cp -r dir1 dir2  # copiar recursivo
6 $ history          # historial de comandos
7 $ which python3    # ruta de un ejecutable
8 $ clear            # limpiar pantalla
```

Nota

Atajos: **Ctrl+C** (cancelar), **Ctrl+A/E** (inicio/fin), **Ctrl+R** (buscar historial), **Tab** (autocompletar).

```
1  $ comando > out.txt           # stdout a archivo (sobrescribe)
2  $ comando » out.txt          # agrega al final
3  $ comando 2> err.txt          # stderr a archivo
4  $ comando 2>&1 | tee log.txt    # mezcla y copia a pantalla+archivo
5  $ dmesg | grep -i usb | less   # filtrar y paginar
6  $ du -sh * | sort -h           # uso de espacio ordenado
```

BÚSQUEDAS: GLOBBING, find Y grep

```
1 $ ls *.txt                                # comodines
2 $ find . -name "*.py"                    # buscar por nombre
3 $ find /var/log -type f -mtime -1        # logs de hoy
4 $ grep -R "ERROR" /var/log/              # buscar texto recursivo
5 $ grep -n "main(" *.c                    # con número de línea
```



Advertencia

Cuidado con `find / -name ...` sin permisos: puede tardar mucho.

```
1 $ ls -l backup.sh
2 -rwxr-xr-- 1 pi pi 54 jul 8 12:30 backup.sh
3
4 $ chmod u+x script.sh      # ejecutar para dueño
5 $ chmod g+w datos/        # escritura para grupo
6 $ chgrp alumnos datos/    # cambiar grupo
7 $ umask                    # máscara por defecto
```



Advertencia

Evita `chmod 777`. Permisos mínimos necesarios.

```
1 $ whoami
2 $ id                # UID/GID y grupos
3 $ groups pi
4 $ sudo -v           # validar sudo
5 $ sudo adduser alumno
6 $ sudo usermod -aG sudo alumno
```

Nota

Trabaja con usuario normal; usa **sudo** sólo cuando haga falta.

```
1  $ echo $HOME $USER $SHELL
2  $ echo $PATH
3  $ export EDITOR=nano
4
5  # alias útiles (añádelos a ~/.bashrc)
6  alias ll='ls -lah'
7  alias gs='git status'
8
9  $ source ~/.bashrc  # recargar configuración
```

EDITORES RÁPIDOS: nano, micro, vim

```
1 $ sudo apt update
2 $ sudo apt install -y nano micro vim
3 $ nano notas.txt      # controles en pantalla
4 $ micro notas.txt     # similar a editor moderno
5 $ vim notas.txt       # potente (requiere práctica)
```



Nota

Para empezar, **nano** o **micro** son ideales.

```
1 $ sudo apt update
2 $ sudo apt upgrade -y
3 $ sudo apt install -y htop tree git curl wget zip unzip
4 $ sudo apt install -y build-essential cmake pkg-config
5 $ sudo apt install -y python3-pip python3-venv
6 $ sudo apt autoremove -y
```

Nota

build-essential instala compiladores y herramientas clave.

```
1  $ htop                # procesos interactivo
2  $ free -h            # memoria
3  $ df -h              # uso de disco
4  $ lsblk              # dispositivos de bloque
5  $ vcgencmd measure_temp # temperatura (Raspberry)
6  $ watch -n 1 "vcgencmd measure_temp"
```

```
1 $ hostname          # nombre de equipo
2 $ hostname -I       # IPs
3 $ ip a              # interfaces
4 $ ping -c 4 8.8.8.8 # conectividad
5 $ ping -c 4 google.com
6 $ sudo apt install -y net-tools
7 $ ifconfig          # (alternativa, en net-tools)
```

Nota

Nombrar tu equipo ayuda: `sudo hostnamectl set-hostname pi-01`

```
1  # scp: copia simple
2  $ scp archivo.txt pi@pi-01:~/    # a la Pi
3  $ scp pi@pi-01:~/datos.csv .    # desde la Pi
4
5  # rsync: sincronización eficiente
6  $ rsync -avh carpeta/ pi@pi-01:~/carpeta/
```



Nota

Usaremos esto cuando activemos SSH mañana.

```
1 $ sudo apt install -y tmux
2 $ tmux new -s trabajo      # crear sesión
3 # dentro: Ctrl+B, luego D -> detach
4 $ tmux ls                  # listar sesiones
5 $ tmux attach -t trabajo   # volver a entrar
```



Nota

Ideal para dejar procesos corriendo sin perderlos al cerrar la terminal.

```
1 $ python3 --version
2 $ python3 -m venv ~/venvs/base
3 $ source ~/venvs/base/bin/activate
4 (base) $ pip install --upgrade pip
5 (base) $ pip install jupyterlab numpy matplotlib
6 (base) $ jupyter lab --no-browser --ip=0.0.0.0
```

Nota

Con SSH haremos *port forwarding* para abrir Jupyter desde tu laptop.

```
1 $ git config --global user.name "Tu Nombre"
2 $ git config --global user.email "tu@correo"
3 $ mkdir demo && cd demo
4 $ git init
5 $ echo "hola" > README.md
6 $ git add README.md
7 $ git commit -m "Primer commit"
```

```
1 $ sudo raspi-config
2 # Cambiar localización, zona horaria, expandir filesystem,
3 # habilitar interfaces (I2C, SPI, cámara), etc.
```



Advertencia

Aplica cambios con cuidado; anota lo que modificas.

```
1 $ sudo apt install -y neofetch ranger ncdu nmap \
2   screenfetch mc jq ripgrep btop
3 $ neofetch      # resumen del sistema
4 $ ncdu         # uso de disco interactivo
5 $ ranger       # explorador en terminal
```



Nota

ncdu y **ranger** ayudan a manejar archivos y espacio.

```
1 $ passwd                                # cambia tu contraseña
2 $ sudo apt update && sudo apt upgrade -y
3 $ sudo apt install -y unattended-upgrades
4 $ sudo dpkg-reconfigure unattended-upgrades
```



Nota

Mantén tu sistema actualizado y evita usar **sudo** sin necesidad.

```
1 $ sudo systemctl enable --now ssh
2 $ systemctl status ssh
3 $ sudo ss -ltnp | grep ':22'    # puerto en escucha
```



Advertencia

Activa SSH sólo cuando controles la red del aula.

```
1  # En tu laptop
2  $ ssh-keygen -t ed25519 -C "tu@correo"
3  $ ssh-copy-id pi@pi-01          # o pi@IP.DE.TU.PI
4  $ ssh pi@pi-01
```

```
1  # Config opcional (~/.ssh/config en tu laptop)
2  Host pi-01
3      HostName 192.168.1.50
4      User pi
5      IdentityFile ~/.ssh/id_ed25519
```

```
1  # Port forwarding (Jupyter en la Pi -> tu laptop)
2  $ ssh -L 8888:localhost:8888 pi@pi-01
3
4  # Copia de archivos
5  $ scp archivo.txt pi@pi-01:~/      # subir
6  $ scp pi@pi-01:~/datos.csv .      # bajar
7
8  # Sincronización eficiente
9  $ rsync -avh carpeta/ pi@pi-01:~/carpeta/
```

```
1 $ ping -c 2 pi-01
2 $ ssh -vvv pi@pi-01      # modo verboso
3 $ sudo journalctl -u ssh -e # logs en la Pi
4 $ sudo ufw status        # firewall (si está activo)
```

Nota

Si falla por nombre, intenta por IP directa.

1. Crear `/proyectos/{scripts,datos,logs}`
2. Escribir `backup.sh` que:
 - reciba carpeta (\$1)
 - comprima con `tar` → `.tar.gz`
 - mueva a `/backups`
3. Añadir bit de ejecución y probar.

💡 Tip: usa `$(date +%F)` para sellar fecha.

- Procesos: `htop` , `ps aux`
- Disco: `df -h` , `du -sh *`
- Buscar: `grep` , `find`
- Red: `ip a` , `ping`
- Editar: `nano` , `vim`
- Actualizar:
`sudo apt update` ~~`&&`~~ `apt upgrade`
- Historial: `history` , `!42`
- Autocompletar: `[?]` (TAB)

- *The Linux Command Line*, W. Shotts Jr. (ebook gratuito)
- <https://linuxjourney.com> – interactivo
- Documentación oficial de Raspberry Pi OS

¡Buen trabajo!

Siguiente bloque: redes IP y claves SSH