

BASH AVANZADO Y TRANSFERENCIA

CONDICIONALES, BUCLES Y COPIAS SEGURAS

Equipo docente ClústerLab

6 de agosto de 2025

Usa `rsync -a` para sincronizar carpetas entre dispositivos.

```
1  # desde local hasta remoto (subir datos)  
2  $ rsync -a datos pi@10.0.0.21:~/datos/
```

Esto es útil para copiar datos de simulación o de análisis que se hagan de forma remota.

```
1  # desde remoto hasta local (bajar datos)  
2  $ rsync -a pi@10.0.0.21:~/datos/ datos/
```

```
1 $ rsync -av --delete origen/ pi@10.0.0.21:~/destino/  
2 $ rsync -av --delete origen/ rpi21:~/destino/
```

- **-delete** borra en destino lo que ya no existe en origen.
- Ideal para copias de seguridad periódicas.

PING A TODO

En serie:

```
1  for host in {1..254}; do
2      ping -c1 -W1 "10.0.0.$host"
3  done
4  wait
```

En paralelo:

```
1  for host in {1..254}; do
2      ping -c1 -W1 "10.0.0.$host" &
3  done
4  wait
```

Ejecutar varios **ping** simultáneos reduce el tiempo total pero genera más tráfico.

```
1 FRUTAS=(manzana pera mango)
2 for fruta in "${FRUTAS[@]}"; do
3     echo "Me gusta la $fruta"
4 done
```

- Las comillas dobles preservan espacios si los hay.
- Añade más elementos y ejecuta de nuevo el ciclo.

Evaluar condiciones y ejecutar bloques de código:

```
1  #!/usr/bin/env bash
2  FILE="$1"
3  if [ -f "$FILE" ]; then
4      echo "El archivo $FILE existe"
5  elif [ -d "$FILE" ]; then
6      echo "$FILE es un directorio"
7  else
8      echo "No existe $FILE"
9  fi
```

Nota

Las expresiones entre corchetes [] admiten comparaciones de cadenas (=, !=) y números (-lt, -eq, etc.).

Copia un archivo a la Pi:

```
1 $ scp informe.pdf pi@10.0.0.21:~/
```

Sincroniza un directorio local con el remoto:

```
1 $ rsync -avz proyectos/ pi@10.0.0.21:~/proyectos/
```

Nota

La opción `-z` de **rsync** habilita compresión; `-a` conserva permisos y tiempos.

Calcula y compara hashes para asegurar integridad:

```
1 $ sha256sum archivo.txt
2 e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855  archivo.txt
```

- Ejecuta el comando antes y después de transferir.
- Si los hashes coinciden, la copia es correcta.

1. Implementa y ejecuta `ping_all.sh` en tu subred.
2. Prueba la variante en paralelo y compara el tiempo de ejecución.
3. Copia un archivo de al menos 10 MB a tu Pi usando `scp` y mide el tiempo.
4. Repite la copia con `rsync` y compara tiempos.
5. Verifica los hashes SHA-256 para confirmar integridad.

- Aprendiste a escribir condicionales y bucles en Bash.
- Creaste un script para mapear y medir latencias en la red.
- Transferiste y sincronizaste archivos de forma eficiente y segura.

¡Terminó la configuración básica de tu entorno remoto!