

Estudiar libro de referencia “El shell bash” antes o durante la resolución de estos ejercicios.

No usar IA, sólo el libro de referencia.

Ejercicios de scripting: variados

Ejercicio 1 – Validador de backups con códigos de salida

- Ejecuta un comando de copia (`rsync` o `cp`) y redirige la salida estándar a `backup.log` y los errores a `backup.err`.
 - Si el código de salida es 0, imprime “**Backup OK**” y añade la fecha a `status.log`.
 - Si falla, imprime “**Backup FAILED**” y termina con `exit 1`.
-

Ejercicio 2 – Filtro inteligente de procesos

- Cuenta los procesos del usuario actual con `ps aux | grep "$USER"`.
 - Excluye del conteo el propio `grep` y el script.
 - Si hay más de 50 procesos, escribe un aviso en `stderr` y devuelve `exit 2`.
 - Si hay entre 0 y 50, guarda el número en `procesos-YYYY-MM-DD.log`.
-

Ejercicio 3 – Sanitizador de logs

- Dado un fichero de log, crea una versión “anonimizada”:
 - Sustituye emails por `<EMAIL>` y teléfonos por `<PHONE>` con `sed`.
 - Si tiene más de N líneas (parámetro), conserva solo las últimas 500 con `tail`.

- Guarda el resultado en `sanitized.log`.
 - Si el fichero no existe, muestra error y termina con `exit 3`.
-

Ejercicio 4 – Informe con `tee` y alertas

- Genera un informe del sistema con: `date, hostname, df -h, free -m`.
 - Usa `tee` para mostrarlo en pantalla y guardarlo en `informe.txt`.
 - Si alguna partición supera el 90% de uso, añade al final “ALERTA DISCO”.
-

Ejercicio 5 – Pipeline robusto con `set -o pipefail`

- Crea un diccionario de palabras de un texto con:
`cat archivo | grep -oE '\w+' | sort | uniq`.
 - Activa `set -euo pipefail`.
 - Si cualquier comando falla, captura el error y muestra un mensaje en `stderr`.
-

Ejercicio 6 – Procesar CSV de usuarios

- Lee `usuarios.csv` con `while IFS=, read -r user email role; do ... done < usuarios.csv`.
 - Guarda líneas con `rol=admin` en `admins.txt` y las demás en `usuarios.txt`.
 - Si alguna línea no tiene 3 campos, escribe esa línea en `malformadas.txt`.
-

Ejercicio 7 – Compresión de logs antiguos

- Encuentra todos los `.log` modificados hace más de 7 días.
 - Comprímelos en `logs-YYYYMMDD.tar.gz`.
 - Si no hay logs, imprime “**Nada que comprimir**” y termina con éxito.
 - Si `tar` falla, termina con `exit 4`.
-

Ejercicio 8 – Menú generador de plantillas

- Crea un menú con `case`:
 1. Generar `README.md` con un here-doc.
 2. Generar un `.gitignore` estándar.
 3. Generar un `deploy.sh` con cabecera `#!/usr/bin/env bash`.
 - Si el archivo ya existe, pregunta al usuario si quiere sobrescribirlo.
-

Ejercicio 9 – Comparar inventarios con process substitution

- Compara dos listados (`lista1.txt` y `lista2.txt`) con:
`diff <(sort lista1.txt) <(sort lista2.txt) > diff.txt`.
 - Si `diff.txt` está vacío, imprime “**Sin cambios**”.
 - Si tiene contenido, imprime “**Cambios detectados**” y muestra las diferencias.
-

Ejercicio 10 – Descarga con reintentos y resumen

- Lee URLs de `urls.txt` y descárgalas con `curl` o `wget`.

- Guarda `stdout` en `ok/<host>.html` y errores en `err/<host>.log`.
- Implementa reintentos exponenciales (1s, 2s, 4s) si el código HTTP no es 200.
- Al final, genera un `report.csv` con el número de descargas correctas y fallidas.