



# Ejercicios de Shell Scripting - Ordenados por Dificultad

Basándome en el trabajo práctico que compartiste, he creado una serie de ejercicios similares organizados de menor a mayor dificultad. Esta progresión te ayudará a reforzar los conceptos fundamentales antes de abordar problemas más complejos.

## Nivel 1: Fundamentos Básicos

### Ejercicio 1: Presentación Personal

Crear un script llamado `info_personal.sh` que:

- Solicite al usuario su nombre, edad y ciudad de origen
- Muestre la información con formato: "Hola [nombre], tienes [edad] años y vives en [ciudad]"
- Agregue la fecha y hora actual del sistema
- Muestre el directorio home del usuario

**Conceptos practicados:** `echo`, `read`, variables, sustitución de comandos

### Ejercicio 2: Sistema de Información

Crear un script que muestre:

- El nombre del usuario actual
- El hostname de la máquina
- El sistema operativo (usando `uname`)
- El espacio libre en disco con `df -h`
- Los procesos del usuario actual

**Conceptos practicados:** Comandos del sistema, sustitución de comandos, variables especiales

### Ejercicio 3: Calculadora Simple

Crear un script que:

- Solicite dos números al usuario
- Muestre: suma, resta, multiplicación, división y módulo
- Indique cuál de los dos números es mayor
- Calcule el promedio de ambos números

**Conceptos practicados:** `expr`, operaciones aritméticas, comparaciones

### Nivel 2: Parámetros y Validaciones

#### Ejercicio 4: Calculadora con Parámetros

Modificar el ejercicio anterior para que:

- Reciba los dos números como parámetros (\$1 y \$2)
- Valide que se hayan enviado exactamente 2 parámetros
- Si faltan parámetros, muestre un mensaje de uso correcto
- Retorne código de error 1 si los parámetros son incorrectos

**Conceptos practicados:**  `$#`, `$1`, `$2`, `exit`, validación de parámetros

#### Ejercicio 5: Información de Archivo

Crear un script que reciba un nombre de archivo/directorio como parámetro y:

- Verifique si existe
- Si existe, indique si es archivo regular, directorio, enlace simbólico u otro
- Muestre los permisos del archivo
- Indique su tamaño en bytes
- Si no existe, pregunte al usuario si desea crearlo (archivo o directorio según elija)

**Conceptos practicados:** `test`, evaluación de archivos, `-f`, `-d`, `-L`, condicionales

#### Ejercicio 6: Conversor de Temperaturas

Crear un script que reciba como parámetros:

- Una temperatura (número)
- Una unidad de origen (C, F, K)
- Una unidad de destino (C, F, K)

Y realice la conversión mostrando el resultado.

**Conceptos practicados:** `case`, operaciones aritméticas, validación de parámetros

### Nivel 3: Estructuras de Control

#### Ejercicio 7: Tabla de Multiplicar

Crear un script que:

- Reciba un número como parámetro
- Si no recibe parámetro, solicite el número al usuario
- Muestre la tabla de multiplicar de ese número del 1 al 12
- Opcionalmente, permita especificar hasta qué número multiplicar

**Conceptos practicados:** `for`, bucles, validación, parámetros opcionales

#### Ejercicio 8: Menú de Operaciones con Archivos

Crear un script que muestre un menú con opciones:

1. Crear archivo
2. Listar archivos del directorio actual
3. Buscar archivo por nombre
4. Eliminar archivo
5. Cambiar permisos de archivo
6. Salir

El menú debe permanecer activo hasta que se elija "Salir".

**Conceptos practicados:** `select` o `case` con `while`, operaciones con archivos

#### Ejercicio 9: Contador de Tipos de Archivo

Crear un script que reciba un directorio como parámetro y cuente:

- Cantidad de archivos regulares
- Cantidad de directorios
- Cantidad de enlaces simbólicos
- Cantidad de archivos ocultos
- Total de elementos

**Conceptos practicados:** Bucles, evaluación de archivos, contadores

## Nivel 4: Arrays y Procesamiento de Datos

### Ejercicio 10: Estadísticas de Array

Dado un array de números: `nums=(45 23 67 12 89 34 56 90 15 78)`

Crear funciones que calculen:

- Suma total
- Promedio
- Valor máximo
- Valor mínimo
- Cantidad de números pares e impares

**Conceptos practicados:** Arrays, bucles, funciones, operaciones matemáticas

### Ejercicio 11: Fusión de Arrays

Dados dos arrays:

```
dias=("Lunes" "Martes" "Miércoles" "Jueves" "Viernes")
temperaturas=(22 25 23 28 26)
```

Crear un script que imprima: "El [día] la temperatura fue de [temp]°C"

**Conceptos practicados:** Arrays, índices, bucles sincronizados

### Ejercicio 12: Procesamiento de Logs

Crear un script que:

- Reciba como parámetro una ruta de archivo de log
- Cuente cuántas líneas contienen "ERROR", "WARNING" e "INFO"
- Genere un reporte con estas estadísticas
- Guarde el reporte en `log_report.txt`

**Conceptos practicados:** `grep`, procesamiento de archivos, contadores, redirección

## Nivel 5: Funciones y Operaciones Complejas

### Ejercicio 13: Gestor de Tareas (TODO List)

Implementar un script con las siguientes funciones:

- `agregar_tarea <descripcion>`: Agrega una tarea al array
- `listar_tareas`: Muestra todas las tareas con índice
- `completar_tarea <indice>`: Marca como completada (agregar "✓")
- `eliminar_tarea <indice>`: Elimina una tarea
- `guardar_tareas`: Guarda en archivo `tareas.txt`
- `cargar_tareas`: Carga desde archivo `tareas.txt`

**Conceptos practicados:** Arrays, funciones, persistencia de datos, manipulación de strings

### Ejercicio 14: Renombrado Masivo con Prefijo/Sufijo

Crear un script que reciba:

- Un directorio
- Una opción: `-p` `TEXT0` (prefijo) o `-s` `TEXT0` (sufijo)
- Un tipo de archivo opcional: `-t` `EXTENSION`

Y renombre todos los archivos (o solo los de la extensión especificada) agregando el prefijo o sufijo.

**Conceptos practicados:** Procesamiento de parámetros, manipulación de nombres, bucles

### Ejercicio 15: Búsqueda de Archivos Duplicados

Crear un script que:

- Reciba un directorio como parámetro
- Identifique archivos con el mismo tamaño
- Use `md5sum` o `sha256sum` para verificar si son realmente duplicados
- Genere un reporte listando los duplicados

**Conceptos practicados:** Arrays asociativos, comandos de hash, comparaciones

## Nivel 6: Procesamiento Avanzado de Texto

### Ejercicio 16: Analizador de CSV

Crear un script que reciba un archivo CSV y:

- Cuente el número de filas y columnas
- Muestre los encabezados (primera línea)

- Permita buscar filas que contengan un texto específico
- Permita extraer una columna específica por número o nombre

**Conceptos practicados:** `cut`, `awk`, procesamiento de texto, arrays

## Ejercicio 17: Generador de Contraseñas

Crear un script que genere contraseñas aleatorias con opciones:

- `-l` LONGITUD: longitud de la contraseña (default: 12)
- `-n` CANTIDAD: cantidad de contraseñas a generar (default: 1)
- `-e`: incluir caracteres especiales
- `-s`: guardar en archivo `passwords.txt`

**Conceptos practicados:** `$RANDOM`, procesamiento de parámetros, strings

## Ejercicio 18: Transformador de Texto

Crear un script que reciba un archivo de texto y permita:

1. Convertir todo a mayúsculas
2. Convertir todo a minúsculas
3. Capitalizar cada palabra
4. Eliminar números
5. Eliminar caracteres especiales
6. Invertir cada línea

Usar estructura de menú para elegir la operación.

**Conceptos practicados:** `tr`, `sed`, transformación de texto, menús

## Nivel 7: Monitoreo y Automatización

### Ejercicio 19: Monitor de Procesos

Crear un script que:

- Reciba el nombre de un proceso como parámetro
- Verifique cada 5 segundos si el proceso está corriendo
- Si el proceso no está activo, envíe una alerta y registre en log
- Permita detener el monitoreo con `Ctrl+C`
- Guarde estadísticas de tiempo activo/inactivo

**Conceptos practicados:** `while`, `sleep`, `ps`, señales, logging

## Ejercicio 20: Backup Inteligente

Crear un script que:

- Reciba un directorio origen y uno destino
- Copie solo archivos modificados en las últimas 24 horas
- Cree un archivo comprimido con fecha: `backup_YYYY-MM-DD.tar.gz`
- Mantenga solo los últimos 7 backups (elimine los más antiguos)
- Registre en log cada operación realizada

**Conceptos practicados:** `find`, `tar`, fechas, compresión, gestión de archivos

## Ejercicio 21: Analizador de Uso de Disco

Crear un script que:

- Escanee el directorio home del usuario
- Identifique los 10 directorios que más espacio ocupan
- Identifique los 10 archivos más grandes
- Encuentre archivos duplicados (mismo tamaño y hash)
- Genere un reporte HTML con los resultados

**Conceptos practicados:** `du`, `find`, `sorting`, generación de HTML

## Nivel 8: Estructuras de Datos Avanzadas

### Ejercicio 22: Implementación de Cola (Queue)

Implementar una estructura de tipo COLA (FIFO) con funciones:

- `enqueue <elemento>`: Agrega al final
- `dequeue`: Extrae del inicio
- `peek`: Muestra el primer elemento sin extraerlo
- `is_empty`: Verifica si está vacía
- `size`: Retorna el tamaño
- `clear`: Limpia la cola

Demostrar su uso con un simulador de atención al cliente.

**Conceptos practicados:** Arrays, funciones, lógica de estructuras de datos

## Ejercicio 23: Sistema de Inventario

Crear un sistema de inventario con arrays asociativos que permita:

- Agregar productos (nombre, cantidad, precio)
- Modificar stock
- Buscar productos
- Listar productos con stock bajo (< 10 unidades)
- Calcular valor total del inventario
- Guardar/cargar desde archivo

**Conceptos practicados:** Arrays asociativos, funciones, persistencia, cálculos

## Ejercicio 24: Árbol de Directorios

Crear un script que:

- Reciba un directorio como parámetro
- Genere una representación visual del árbol de directorios (similar a `tree`)
- Muestre niveles de profundidad con indentación
- Opcionalmente limite la profundidad máxima
- Permita filtrar por extensión de archivo

**Conceptos practicados:** Recursión, funciones, formateo de salida

## Nivel 9: Integración y Aplicaciones Complejas

### Ejercicio 25: Sistema de Gestión de Usuarios

Crear un script que permita:

- Listar usuarios del sistema filtrados por grupo
- Mostrar información detallada de un usuario (UID, GID, shell, home)
- Verificar si un usuario tiene permisos de lectura/escritura en un directorio
- Generar reporte de usuarios con sus últimos accesos
- Exportar información a CSV

**Conceptos practicados:** `/etc/passwd`, `/etc/group`, procesamiento de archivos del sistema, CSV



## Ejercicio 26: Organizador de Fotos

Crear un script que:

- Reciba un directorio con imágenes
- Organice las fotos por fecha de creación en estructura: YYYY/MM/DD/
- Renombre archivos duplicados agregando sufijo numérico
- Genere miniaturas en subdirectorio `thumbnails/` (usando `convert` de ImageMagick)
- Cree un índice HTML con las imágenes

**Conceptos practicados:** Fechas, organización de archivos, comandos externos, HTML

## Ejercicio 27: Parser de Logs del Sistema

Crear un script que analice `/var/log/syslog` (o `/var/log/messages`) y:

- Extraiga los errores del último día
- Agrupe errores por tipo/servicio
- Cuente la frecuencia de cada error
- Identifique patrones sospechosos (múltiples fallos del mismo servicio)
- Genere un dashboard en HTML con gráficos ASCII

**Conceptos practicados:** Expresiones regulares, `awk`, `sed`, procesamiento complejo de texto

## Nivel 10: Proyectos Integradores

### Ejercicio 28: Sistema de Respaldo Incremental

Implementar un sistema de backup que:

- Realice backup completo inicial
- Backups incrementales basados en cambios
- Mantenga un registro de versiones
- Permita restaurar a una fecha específica
- Implemente rotación de backups (diario, semanal, mensual)
- Envíe notificaciones por email en caso de errores

**Conceptos practicados:** Gestión de archivos, fechas, lógica compleja, integración de comandos

## Ejercicio 29: Juego de Ahorcado

Implementar el juego del ahorcado con:

- Array de palabras aleatorias
- Dibujo ASCII del ahorcado que se actualiza
- Validación de letras ya ingresadas
- Contador de intentos fallidos
- Sistema de puntuación
- Opción de jugar múltiples rondas
- Tabla de mejores puntuaciones guardada en archivo

**Conceptos practicados:** Lógica de juego, arrays, strings, funciones, persistencia

## Ejercicio 30: Gestor de Proyectos Git

Crear un script que:

- Liste todos los repositorios Git en un directorio
- Muestre el estado de cada uno (branch actual, cambios pendientes, commits sin push)
- Permita hacer pull en todos los repositorios
- Identifique repositorios con cambios sin commitear
- Genere un reporte de actividad por repositorio
- Permita buscar en commits de todos los repositorios

**Conceptos practicados:** Integración con Git, procesamiento de múltiples directorios, reportes

## Recomendaciones de Estudio

1. **Nivel 1-3:** Enfócate en dominar la sintaxis básica y las estructuras de control
2. **Nivel 4-6:** Practica el manejo de datos y procesamiento de archivos
3. **Nivel 7-8:** Desarrolla habilidades de automatización y estructuras complejas
4. **Nivel 9-10:** Integra múltiples conceptos en aplicaciones reales

Cada ejercicio está diseñado para reforzar los conceptos del trabajo práctico original mientras introduce gradualmente nueva complejidad. Te sugiero completar al menos 2-3 ejercicios de cada nivel antes de avanzar al siguiente.

