

## Linux 3

```
~ /workspace/test/test/test: ./vars.sh hola como estas 123 777

~/workspace$ chmod +x mkcd.sh
~/workspace$ ./mkcd.sh test
~/workspace$ . ./mkcd.sh test
~/workspace/test$ code mcd.sh
~/workspace/test$ source mcd.sh
~/workspace/test$ mcd test
~/.../test/test$ safe_mcd() {
    if [ -z "$1" ]; then
        echo "Error: Debes proporcionar un nombre de directorio."
        return 1
    fi
    mkdir -p "$1" && cd "$1" || echo "No se pudo cambiar al directorio"
    . "$1"
}
~/.../test/test$ code safe_mcd()
bash: syntax error near unexpected token `('
~/.../test/test$ code safe_mcd.sh
~/.../test/test$ source safe_mcd.sh
~/.../test/test$ safe_mcd test
~/.../test/test$ code vars.sh
~/.../test/test$ chmod +x vars.sh
~/.../test/test$ ./vars.sh hola como estas 123 777
Nombre del script: ./vars.sh
Primer argumento: hola
Segundo argumento: como
Número de argumentos: 5
Todos los argumentos (usando $@): hola como estas 123 777
Todos los argumentos (usando $*): hola como estas 123 777
ls: cannot access '/nonexistentfile': No such file or directory
Código de salida del último comando: 2
PID del script: 810
PID del último comando en segundo plano: 812
Opciones de shell actuales: hB
Último argumento del último comando ejecutado: 3
```

```
~ /workspace/test/test/test: set -o

~/.../test/test$ echo {1..10}
1 2 3 4 5 6 7 8 9 10
~/.../test/test$ echo {1..10..2}
1 3 5 7 9
~/.../test/test$ echo {a..z..5}
a f k p u z
~/.../test/test$ mkdir carpeta{1..10}_ver{A..B}
~/.../test/test$ set +B
~/.../test/test$ echo {1..10}
{1..10}
~/.../test/test$ set -B
~/.../test/test$ echo $SHELLOPTS
braceexpand:emacs:hashall:histexpand:history:interactive-comments:monitor
~/.../test/test$ set -o
allexport          off
braceexpand        on
emacs              on
errexit            off
errtrace           off
functrace          off
hashall            on
histexpand         on
history            on
ignoreeof          off
interactive-comments on
keyword            off
monitor            on
noclobber          off
noexec             off
noglob             off
nolog              off
notify             off
nounset            off
```

## Ejercicios

1.

a. Crear archivos numerados

```
~ /workspace: for i in {1..10}; do touch \"archivo_$i\"; done

~/workspace$ for i in {1..10}; do
  touch \"archivo_$i\"
done
```

### b. Crear directorios por letras

```
~/workspace$ for letra in {A..E}; do  
    mkdir "proyecto_${letra}"  
done
```

### c. Crear archivos combinados

```
~/workspace$ for num in {1..3}; do  
    for letra in {a..c}; do  
        touch "${num}${letra}.txt" "${num}${letra}.pdf"  
    done  
done
```

### d. Estructura de proyectos

```
~/workspace$ mkdir -p projects/{dev,prod}/{config1,config2,config3,logs}  
~/workspace$ tree projects  
projects  
├── dev  
│   ├── config1  
│   ├── config2  
│   ├── config3  
│   └── logs  
└── prod  
    ├── config1  
    ├── config2  
    ├── config3  
    └── logs  
  
10 directories, 0 files
```

## 2. Generación de secuencias:

### a. Imprimir números pares:

```
~/workspace$ for i in {2..20..2}; do  
    echo $i  
done  
2  
4  
6  
8  
10  
12  
14  
16  
18  
20
```

**b. Imprimir letras del alfabeto:**

```
~/workspace$ for letra in {b..z..2}; do  
    echo $letra  
done  
b  
d  
f  
h  
j  
l  
n  
p  
r  
t  
v  
x  
z
```

**c. Generar fechas:**

```
~/workspace$ for dia in {01..10}; do  
    echo "2024-01-$dia"  
done  
2024-01-01  
2024-01-02  
2024-01-03  
2024-01-04  
2024-01-05  
2024-01-06  
2024-01-07  
2024-01-08  
2024-01-09  
2024-01-10
```

**d. Combinaciones aleatorias:**

```
~/workspace$ shuf -e {a..z}{a..z}{a..z} -n 4
sar
cxz
hdn
qmr
```

- $\{a..z\} \{a..z\} \{a..z\}$  = genera todas las combinaciones posibles de tres letras, desde aaa hasta zzz.
- `shuf -e` = toma una lista de elementos y los mezcla aleatoriamente.
- `-n 4` = selecciona 4 elementos al azar.

## Ejemplo genesis

```
~/workspace$ code genesis.sh
~/workspace$ chmod +x genesis.sh
~/workspace$ ./genesis.sh
~/workspace$ #!/bin/bash
~/workspace$ generar_texto() {
    cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 80 | head -n 65
}
~/workspace$ generar_imagen() {
    local archivo="$1"
    convert -size 400x400 \
    plasma:fractal \
    -modulate 100,150,100 \
    -swirl $((RANDOM % 360)) \
    -paint $((RANDOM % 5 + 1)) \
    "$archivo"
}
~/workspace$ for i in {1..5}; do
    echo "$(generar_texto)" > "./archivo$i.txt"
done
~/workspace$ for i in {1..5}; do
    cp archivo$i.txt archivo$i.html
    wkhtmltopdf -q archivo$i.html archivo$i.pdf
    rm archivo$i.html
done
~/workspace$ for i in {1..5}; do
    generar_imagen "imagen${i}.png"
done
```

## Ejemplo docninja

```
~ /workspace: for archivo in *; do extension="\${archivo##*.*}"; ...  
~ /workspace$ code docninja.sh  
~ /workspace$ chmod +x docninja.sh  
~ /workspace$ ./docninja.sh  
Archivo add_headers.sh: extensión no reconocida  
Archivo archivo_1: extensión no reconocida  
Archivo archivo_10: extensión no reconocida  
Archivo archivo_2: extensión no reconocida  
Archivo archivo_3: extensión no reconocida  
Archivo archivo_4: extensión no reconocida  
Archivo archivo_5: extensión no reconocida  
Archivo archivo_6: extensión no reconocida  
Archivo archivo_7: extensión no reconocida  
Archivo archivo_8: extensión no reconocida  
Archivo archivo_9: extensión no reconocida  
Archivo docninja.sh: extensión no reconocida  
Archivo documentos: extensión no reconocida  
Archivo genesis.sh: extensión no reconocida  
Archivo hacker-lab: extensión no reconocida  
Archivo imagenes: extensión no reconocida  
Archivo img: extensión no reconocida  
Archivo linux1_shell.md: extensión no reconocida  
Archivo linux2_commands.md: extensión no reconocida  
Archivo linux3_scripting.md: extensión no reconocida  
Archivo mkcd.sh: extensión no reconocida  
Archivo projects: extensión no reconocida  
Archivo Proyecto: extensión no reconocida  
Archivo proyecto_A: extensión no reconocida  
Archivo proyecto_B: extensión no reconocida  
Archivo proyecto_C: extensión no reconocida  
Archivo proyecto_D: extensión no reconocida  
Archivo proyecto_E: extensión no reconocida  
Archivo replit.nix: extensión no reconocida  
Archivo test: extensión no reconocida  
Archivo textos: extensión no reconocida
```

```
~ /workspace: for archivo in *; do extension="\${archivo##*.}\\"; ... Q X
~/workspace$ #!/bin/bash
# Directorios de destino
directorio_textos="textos"
directorio_pdf="documentos"
directorio_imagenes="imagenes"
~/workspace$ # Crear directorios si no existen
mkdir -p "$directorio_textos" "$directorio_pdf" "$directorio_imagenes"
~/workspace$ # Mover archivos
for archivo in *; do
    extension="\${archivo##*.}"
    case "$extension" in
        txt)
            mv "$archivo" "$directorio_textos"
            ;;
        pdf)
            mv "$archivo" "$directorio_pdf"
            ;;
        png)
            mv "$archivo" "$directorio_imagenes"
            ;;
        *)
            echo "Archivo $archivo: extensión no reconocida"
            ;;
    esac
done
Archivo add_headers.sh: extensión no reconocida
Archivo archivo_1: extensión no reconocida
Archivo archivo_10: extensión no reconocida
Archivo archivo_2: extensión no reconocida
Archivo archivo_3: extensión no reconocida
Archivo archivo_4: extensión no reconocida
Archivo archivo_5: extensión no reconocida
Archivo archivo_6: extensión no reconocida
Archivo archivo_7: extensión no reconocida
```

## Actividad 1

```
~ /workspace: tree Proyecto Q X
~/workspace$ mkdir -p Proyecto/{src/{module_{1..3},tests},docs/{guides,
,references},build/{debug,release}}
touch Proyecto/src/module_{1..3}/main_{A,B}.sh
~/workspace$ tree Proyecto
Proyecto
├── build
│   ├── debug
│   └── release
├── docs
│   ├── guides
│   └── references
└── src
    ├── module_1
    │   ├── main_A.sh
    │   └── main_B.sh
    ├── module_2
    │   ├── main_A.sh
    │   └── main_B.sh
    ├── module_3
    │   ├── main_A.sh
    │   └── main_B.sh
    └── tests

11 directories, 6 files
```

## Actividad 2: Manipulación de Archivos Generados

```
~/workspace: cat Proyecto/src/module_1/main_A.sh

~/workspace$ code add_headers.sh
~/workspace$ chmod +x add_headers.sh
~/workspace$ ./add_headers.sh
Añadiendo encabezado a: Proyecto/src/module_1/main_A.sh
Añadiendo encabezado a: Proyecto/src/module_1/main_B.sh
Añadiendo encabezado a: Proyecto/src/module_2/main_A.sh
Añadiendo encabezado a: Proyecto/src/module_2/main_B.sh
Añadiendo encabezado a: Proyecto/src/module_3/main_A.sh
Añadiendo encabezado a: Proyecto/src/module_3/main_B.sh
Encabezados agregados exitosamente.
~/workspace$ cat Proyecto/src/module_1/main_A.sh
# =====
# Archivo: main_A.sh
# Fecha de creación: 2025-10-08 04:51:17
# Descripción: Script principal del módulo.
# =====

# =====
# Archivo: main_A.sh
# Fecha de creación: 2025-10-08 04:50:08
# Descripción: Script principal del módulo.
# =====

# =====
# Archivo: main_A.sh
# Fecha de creación: 2025-10-08 04:43:45
# Descripción: Script principal del módulo.
# =====
```

## Actividad 3: Análisis de Variables Especiales



```
~ /workspace: ./analizar_variables.sh hola mundo bash
~/workspace$ code analizar_variables.sh
~/workspace$ chmod +x analizar_variables.sh
~/workspace$ ./analizar_variables.sh hola mundo bash
===== Análisis de Variables Especiales =====

1 Modos de operación actuales del shell:
allexport      off
braceexpand   on
emacs         off
errexit       off
errtrace      off
functrace     off
hashall       on
histexpand    off
history       off
ignoreeof     off
interactive-comments  on
keyword       off
monitor       off
noclobber     off
noexec        off
noglob        off
nolog         off
notify        off
nounset       off
onecmd        off
physical      off
pipefail      off
posix         off
privileged    off
verbose       off
vi            off
xtrace        off
```

```
nounset       off
onecmd        off
physical      off
pipefail      off
posix         off
privileged    off
verbose       off
vi            off
xtrace        off

Contenido de $-:
hB

2 Ejecutando un comando en segundo plano...
PID del proceso en segundo plano: 2403

3 Argumentos recibidos por el script ($@):
hola mundo bash

===== Fin del análisis =====
```

## Actividad 4

```
~/workspace: bash -x condicional.sh start

~/workspace$ code condicional.sh
~/workspace$ chmod +x condicional.sh
~/workspace$ ./condicional.sh
No se pasaron argumentos.
~/workspace$ ./condicional.sh start
Iniciando proceso en segundo plano...
Proceso iniciado con PID: 2712
~/workspace$ bash -x condicional.sh start
+ [[ hxB == *x* ]]
+ echo 'Modo de depuración activado'
Modo de depuración activado
+ [[ 1 -eq 0 ]]
+ [[ 1 -eq 1 ]]
+ [[ start == \s\t\a\r\t ]]
+ echo 'Iniciando proceso en segundo plano...'
Iniciando proceso en segundo plano...
+ echo 'Proceso iniciado con PID: 2725'
Proceso iniciado con PID: 2725
+ sleep 20
```

## Actividad 5

```
~/workspace: ./backup.sh

~/workspace$ code backup.sh
~/workspace$ chmod +x backup.sh
~/workspace$ ./backup.sh
Backup creado con éxito: backup_20251008.tar.gz
Enviando correo electrónico con el backup y el log adjunto...
Correo enviado con éxito (simulado).
~/workspace$ █
```

## Actividad tarea

~/workspace: ./tareas.sh

```
~/workspace$ code tareas.sh
~/workspace$ chmod +x tareas.sh
~/workspace$ ./tareas.sh
```

Generate Ctrl I

==== Administrador de Tareas =====

1. Agregar tarea
2. Listar tareas
3. Eliminar tarea
4. Salir

=====

Selecciona una opción (1-4): 1  
Ingresa la descripción de la tarea:  
Matematica aplicada  
Tarea agregada correctamente.

==== Administrador de Tareas =====

1. Agregar tarea
2. Listar tareas
3. Eliminar tarea
4. Salir

=====

Selecciona una opción (1-4): 1  
Ingresa la descripción de la tarea:  
algoritmos  
Tarea agregada correctamente.

==== Administrador de Tareas =====

1. Agregar tarea
2. Listar tareas
3. Eliminar tarea
4. Salir

=====

Selecciona una opción (1-4): 2  
Lista de tareas:

~/workspace: ./tareas.sh

Q  X

==== Administrador de Tareas ====

1. Agregar tarea
2. Listar tareas
3. Eliminar tarea
4. Salir

=====

Selecciona una opción (1-4): 2

Lista de tareas:

1. Matematica aplicada
2. algoritmos

==== Administrador de Tareas ====

1. Agregar tarea
2. Listar tareas
3. Eliminar tarea
4. Salir

=====

Selecciona una opción (1-4): 3

Lista de tareas:

1. Matematica aplicada
2. algoritmos

Ingresa el número de la tarea que deseas eliminar:

algoritmos

Debes ingresar un número válido.

==== Administrador de Tareas ====

1. Agregar tarea
2. Listar tareas
3. Eliminar tarea
4. Salir

=====

Selecciona una opción (1-4):

>\_ tareas.sh

```
1  #!/bin/bash
2
3  # Archivo donde se guardarán las tareas
4  ARCHIVO="tareas.txt"
5
6  # Crear archivo si no existe
7  touch "$ARCHIVO"
8
9  # === Función para agregar una tarea ===
10 agregar_tarea() {
11     echo "Ingresa la descripción de la tarea:"
12     read -r tarea
13     if [[ -z "$tarea" ]]; then
14         echo " No puedes agregar una tarea vacía."
15     else
16         echo "$tarea" >> "$ARCHIVO"
17         echo " Tarea agregada correctamente."
18     fi
19 }
20
21 # === Función para listar tareas ===
22 listar_tareas() {
23     if [[ ! -s "$ARCHIVO" ]]; then
24         echo " No hay tareas registradas."
25     else
```

>\_ tareas.sh

```
26     echo " Lista de tareas:"
27     nl -w2 -s". " "$ARCHIVO"
28 fi
29 }
30
31 # === Función para eliminar una tarea ===
32 eliminar_tarea() {
33     listar_tareas
34     echo ""
35     echo "Ingresa el número de la tarea que deseas
eliminar:"
36     read -r num
37
38     if ! [[ "$num" =~ ^[0-9]+$ ]]; then
39         echo " Debes ingresar un número válido."
40         return
41     fi
42
43     total=$(wc -l < "$ARCHIVO")
44     if (( num < 1 || num > total )); then
45         echo " Número fuera de rango."
46         return
47     fi
48
49     # Eliminar línea usando sed y guardar en archivo
```

>\_ tareas.sh

```
50     sed "${num}d" "$ARCHIVO" > temp.txt && mv temp.txt
    "$ARCHIVO"
51     echo " Tarea número $num eliminada correctamente."
52 }
53
54 # === Menú principal ===
55 while true; do
56     echo ""
57     echo "==== Administrador de Tareas ====="
58     echo "1. Agregar tarea"
59     echo "2. Listar tareas"
60     echo "3. Eliminar tarea"
61     echo "4. Salir"
62     echo "===== "
63     echo -n "Selecciona una opción (1-4): "
64     read -r opcion
65
66     case $opcion in
67         1) agregar_tarea ;;
68         2) listar_tareas ;;
69         3) eliminar_tarea ;;
70         4) echo "Saliendo... ¡Hasta pronto!"; break ;;
71         *) echo "Opción inválida. Intenta nuevamente." ;;
72     esac
73 done
```

