1. Variables

variables.sh

Declaración de variables

valor1=10

valor2=20

Mostrar valores de las variables

echo "El valor de valor1 es: $valor1"

echo "El valor de valor2 es: $valor2"

```

## 2. Parámetros

**nombre.sh**

Leer nombre y apellidos por teclado

echo "Por favor, ingresa tu nombre:"

read nombre

echo "Por favor, ingresa tus apellidos:"

read apellidos

Mostrar nombre completo

echo "Nombre completo: $nombre $apellidos"

```

suma.sh

Leer dos valores por teclado

```
echo "Por favor, ingresa el primer valor:"
read valor1
echo "Por favor, ingresa el segundo valor:"
read valor2

Calcular y mostrar la suma
suma=$(($valor1 + $valor2))
echo "La suma de $valor1 y $valor2 es: $suma"
```

3. E/S por consola

operaciones.sh

Leer tres valores por teclado

```
echo "Por favor, ingresa el primer valor:"
read valor1
echo "Por favor, ingresa el segundo valor:"
read valor2
echo "Por favor, ingresa el tercer valor:"
read valor3
```

Calcular y mostrar suma y multiplicación

suma=$((valor1 + valor2 + valor3))

multiplicacion=$((valor1 * valor2 * valor3))

echo "La suma de $valor1, $valor2 y $valor3 es: $suma"

echo "La multiplicación de $valor1, $valor2 y $valor3 es: $multiplicacion"

```

4. Redirección de la E/S

**leeretc.sh**

Leer contenido del directorio /etc y guardarlo en contenidoetc.txt

ls /etc > "contenidoetc.txt"

echo "El contenido del directorio /etc se ha guardado en contenidoetc.txt"

```

showfile.sh

Mostrar contenido de /etc/passwd y /etc/shadow

echo "Contenido del archivo /etc/passwd:"

cat /etc/passwd

echo "-----------------------------------"

echo "Contenido del archivo /etc/shadow:"

cat /etc/shadow

```
```

5. Filtrado de textos: grep

usuario.sh

Mostrar líneas de /etc/passwd que contienen el nombre de usuario actual

usuario=$(whoami)

echo "Líneas en /etc/passwd con el nombre de usuario '$usuario':"

grep "$usuario" /etc/passwd

```

***usuario2.sh***

Mostrar líneas de /etc/passwd que contienen un nombre de usuario especificado

echo "Introduce el nombre de usuario:"

read nombre_usuario

echo "Líneas en /etc/passwd con el nombre de usuario '$nombre_usuario':"

grep "$nombre_usuario" /etc/passwd

```

6. Filtrado de textos: head/tail

startfile.sh

Mostrar las primeras 3 líneas de /etc/group

```
echo "Primeras 3 líneas del archivo /etc/group:"
head -n 3 /etc/group
```

endfile.sh

Mostrar las últimas 3 líneas de /etc/group

```
echo "Últimas 3 líneas del archivo /etc/group:"
tail -n 3 /etc/group
```

startfile2.sh

Mostrar las primeras x líneas de /etc/group

```
echo "Introduce el número de líneas a mostrar:"
read numero_lineas
echo "Primeras $numero_lineas líneas del archivo /etc/group:"
head -n "$numero_lineas" /etc/group
```

7. Filtrado de textos: cut/sort

sortusers.sh

Mostrar usuarios del sistema ordenados alfabéticamente

```
echo "Usuarios del sistema ordenados alfabéticamente:"
```

```
cut -d: -f1 /etc/passwd | sort
```

sortgroups.sh

Mostrar grupos del sistema ordenados alfabéticamente de mayor a menor

```
echo "Grupos del sistema ordenados alfabéticamente (de mayor a menor):"
cut -d: -f1 /etc/group | sort -r
```

8. Operaciones aritmeticológicas

calc.sh

Sencilla calculadora para suma, resta y división

```
echo "Introduce la operación a realizar: suma, resta, división"
read operacion
echo "Introduce un valor"
read operando1
echo "Introduce otro valor"
read operando2

case $operacion in
  suma)
    resultado=$(($operando1 + $operando2))
    echo "El resultado de la suma es: $resultado"
    ;;
```

```
  resta)
    resultado=$(($operando1 - $operando2))
    echo "El resultado de la resta es: $resultado"
    ;;
  división)
    if [ $operando2 -eq 0 ]; then
      echo "No se puede dividir por cero."
      exit 1
    fi
    resultado=$(awk "BEGIN {printf \"%.2f\", $operando1 / $operando2}")
    echo "El resultado de la división es: $resultado"
    ;;
  *)
    echo "Operación no válida. Por favor, introduce 'suma', 'resta' o 'división'."
    exit 1
    ;;
esac
```

9. Estructuras de control – if

comptextos.sh

Comparar dos cadenas de texto alfanuméricas

```
echo "Introduce la primera cadena de texto:"
read texto1
echo "Introduce la segunda cadena de texto:"
```

```
read texto2
if [ "$texto1" = "$texto2" ]; then
   echo "Las cadenas de texto son iguales."
else
   echo "Las cadenas de texto son diferentes."
fi
```

compnums.sh

Comparar dos números

```
echo "Introduce el primer número:"
read num1
echo "Introduce el segundo número:"
read num2
if [ "$num1" -eq "$num2" ]; then
   echo "Los números son iguales."
else
   echo "Los números son diferentes."
fi
```

10. Estructuras de control – case

nettools.sh

Mostrar un menú con herramientas de monitorización y ejecutar la seleccionada

```
mostrar_menu() {
    echo "Seleccione una herramienta de monitorización:"
    echo "1. top"
    echo "2. ps"
    echo "3. df"
    echo "4. who"
    echo "5. Salir"
}

ejecutar_herramienta() {
    case $opcion in
        1)
            top
            ;;
        2)
            ps aux
            ;;
        3)
            df -h
            ;;
        4)
            who
            ;;
        5)
            echo "Saliendo del programa."
            exit 0
            ;;
        *)
```

```
        echo "Opción no válida."
        ;;
    esac
}

while true; do
    mostrar_menu
    read -p "Ingrese el número de la herramienta a ejecutar (1-5): " opcion

    if ! [[ "$opcion" =~ ^[1-5]$ ]]; then
        echo "Por favor, ingrese un número válido (1-5)."
        continue
    fi

    ejecutar_herramienta $opcion
done
```

11. Estructuras de control – for

dominios.sh

Listar la información DNS de dos dominios

```
echo "Introduce el primer dominio:"
read dominio1
echo "Introduce el segundo dominio:"
read dominio2
```

```
info_dominio1=$(/usr/bin/host "$dominio1")
info_dominio2=$(/usr/bin/host "$dominio2")

echo "Información DNS del dominio $dominio1:"
echo "$info_dominio1"
echo "-----------------------------------------"
echo "Información DNS del dominio $dominio2:"
echo "$info_dominio2"
```

12. Estructuras de control – while

lectura.sh

```
 Leer datos de teclado hasta que se ingrese '0'
entrada=""
while [ "$entrada" != "0" ]; do
   echo "Introduce un dato (ingresa '0' para salir):"
   read entrada
done
echo "Has ingresado '0'. Saliendo del script."
```

13. Funciones

funcionescalc.sh

Calculadora con funciones para suma, resta, multiplicación y división

```
suma() {
   resultado=$(($1 + $2))
   echo "El resultado de la suma es: $resultado"
}

resta() {
   resultado=$(($1 - $2))
   echo "El resultado de la resta es: $resultado"
}

multiplicacion() {
   resultado=$(($1 * $2))
   echo "El resultado de la multiplicación es: $resultado"
}

division() {
   if [ $2 -eq 0 ];
```

scripts2.sh

```
# Script 1: Ingresar nombre, apellidos y DNI

# Este script pide al usuario que ingrese su nombre, apellidos y DNI, y luego los muestra.

read -p "Ingrese su nombre: " nombre
```

```
read -p "Ingrese sus apellidos: " apellidos

read -p "Ingrese su DNI: " dni

echo "Nombre: $nombre"

echo "Apellidos: $apellidos"

echo "DNI: $dni"


# Script 2: Mostrar los números del 1 al 100

# Este script muestra los números del 1 al 100.

for i in {1..100}

do

  echo $i

done


# Script 3: Comprobar si un número es mayor, menor o igual a 15

# Este script pide al usuario que ingrese un número y luego indica si es mayor, menor o igual a 15.

read -p "Ingrese un número: " numero

if [ $numero -gt 15 ]; then

  echo "El número es mayor a 15"

elif [ $numero -lt 15 ]; then

  echo "El número es menor a 15"

else

  echo "El número es igual a 15"

fi


# Script 4: Comprobar si una palabra es un directorio en el home del usuario

# Este script pide al usuario que ingrese un nombre de directorio y verifica si existe en el home del usuario.
```

```
read -p "Ingrese un nombre de directorio: " dir
if [ -d "$HOME/$dir" ]; then
  echo "El directorio existe en el home del usuario"
else
  echo "El directorio no existe en el home del usuario"
fi

# Script 5: Comprobar si la palabra ingresada es "puzzle"
# Este script pide al usuario que ingrese una palabra y verifica si es "puzzle".
read -p "Ingrese una palabra: " palabra
if [ "$palabra" == "puzzle" ]; then
  echo "Has entrado la palabra puzzle"
else
  echo "La palabra entrada no es puzzle"
fi

# Script 6: Restar dos números ingresados por teclado
# Este script pide al usuario que ingrese dos números y muestra el resultado de su resta.
read -p "Ingrese el primer número: " num1
read -p "Ingrese el segundo número: " num2
resultado=$((num1 - num2))
echo "El resultado de la resta es: $resultado"

# Script 7: Buscar una palabra en un fichero de texto
# Este script pide al usuario que ingrese una palabra y un nombre de fichero, y busca la palabra en el fichero.
read -p "Ingrese la palabra a buscar: " palabra
```

```
read -p "Ingrese el nombre del fichero: " fichero
if grep -q "$palabra" "$fichero"; then
  echo "La palabra '$palabra' se encontró en el fichero."
else
  echo "La palabra '$palabra' no se encontró en el fichero."
fi

# Script 8: Listar 4 directorios usando un for
# Este script lista cuatro directorios específicos.
dirs=( "/home" "/etc" "/var" "/usr" )
for dir in "${dirs[@]}"
do
  echo $dir
done

# Script 9: Menú con opciones de suma, resta y multiplicación
# Este script muestra un menú con opciones para sumar, restar, multiplicar dos números o salir.
while true; do
  echo "1. Suma"
  echo "2. Resta"
  echo "3. Multiplicación"
  echo "4. Salir"
  read -p "Elija una opción: " opcion

  case $opcion in
    1)
      read -p "Ingrese el primer número: " num1
```

```
    read -p "Ingrese el segundo número: " num2
    echo "El resultado es: $((num1 + num2))"
    ;;
  2)
    read -p "Ingrese el primer número: " num1
    read -p "Ingrese el segundo número: " num2
    echo "El resultado es: $((num1 - num2))"
    ;;
  3)
    read -p "Ingrese el primer número: " num1
    read -p "Ingrese el segundo número: " num2
    echo "El resultado es: $((num1 * num2))"
    ;;
  4)
    echo "Saliendo..."
    break
    ;;
  *)
    echo "Opción no válida"
    ;;
 esac
done

# Script 10: Menú para crear directorio, listar directorio actual, editar /etc/passwd
# Este script muestra un menú con opciones para crear un directorio, listar el directorio actual, editar el fichero /etc/passwd o salir.
while true; do
 echo "1. Crear un directorio"
```

```
 echo "2. Listar el directorio actual"
 echo "3. Editar el fichero /etc/passwd"
 echo "4. Salir"
 read -p "Elija una opción: " opcion

 case $opcion in
  1)
   read -p "Ingrese el nombre del directorio a crear: " dirname
   mkdir -p "$dirname"
   echo "Directorio '$dirname' creado."
   ;;
  2)
   ls -l
   ;;
  3)
   sudo nano /etc/passwd
   ;;
  4)
   echo "Saliendo..."
   break
   ;;
  *)
   echo "Opción no válida"
   ;;
 esac
done
```

scripts3.sh

```
# Script 1: Comprobar si un usuario y un grupo existen en el sistema

# Este script pide al usuario que ingrese un nombre de usuario y un nombre de grupo, y verifica si existen en el sistema.

read -p "Ingrese el nombre del usuario: " usuario

read -p "Ingrese el nombre del grupo: " grupo


# Comprobar si el usuario existe

if grep  "$usuario" /etc/passwd; then

   echo "El usuario $usuario existe."

else

   echo "El usuario $usuario no existe."

fi


# Comprobar si el grupo existe

if grep  "$grupo" /etc/group; then

   echo "El grupo $grupo existe."

else

   echo "El grupo $grupo no existe."

fi


# Script 2: Comprobar si el usuario actual es Pepe

# Este script verifica si el usuario actual es "Pepe" y muestra un mensaje en consecuencia.

if [ "$USER" == "Pepe" ]; then

   echo "Hola, Pepe"

else

   echo "Adiós"
```

```
fi

# Este script verifica si el usuario actual es "root" y muestra un mensaje en consecuencia.
if ["$USER" = "root" ]; then
echo "Usuario conectado"
else
"Chau"
fi

# Script 3: Comprobar si estamos en el día 17
# Este script verifica si el día actual es 17 y muestra un mensaje en consecuencia.
if [ $(date +%d) -eq 17 ]; then
   echo "Hoy es el día 17."
else
   echo "Hoy no es el día 17."
fi

# Script 4: Verificar si la segunda columna de un archivo contiene la palabra "pepe"
# Este script verifica si la segunda columna de un archivo contiene la palabra "pepe".
archivo="archivo.txt"

cut -d' ' -f2 pepe.txt > a.txt
if grep -q pepe a.txt; then
  echo "pepe existe en la 2 columna"
else
  echo "pepe no existe"
fi
```

```
# Script 5: Obtener las líneas del archivo /etc/passwd que empiezan con 'r'

# Este script muestra las líneas del archivo /etc/passwd que empiezan con la letra 'r'.

grep ^r /etc/passwd


# Script 6: Obtener personas que vienen de Rusia

# Este script muestra las personas que vienen de Rusia de un archivo llamado personas.txt.

archivo="personas.txt"


while IFS=: read -r nombre pais

do

  if [ "$pais" == "Rusia" ]; then

    echo "$nombre viene de Rusia"

  fi

done < "$archivo"


# Script 7: Determinar si se ha ingresado una letra mayúscula, una letra minúscula o un número

# Este script determina si el carácter ingresado es una letra mayúscula, una letra minúscula o un número.

read -p "Ingrese un carácter: " char


if [[ "$char" =~ [A-Z] ]]; then

  echo "Has ingresado una letra mayúscula."

elif [[ "$char" =~ [a-z] ]]; then

  echo "Has ingresado una letra minúscula."

elif [[ "$char" =~ [0-9] ]]; then

  echo "Has ingresado un número."
```

```
else
    echo "No has ingresado una letra ni un número."
fi

# Script 8: Ingresar números o letras hasta presionar el número 0 para salir
# Este script permite ingresar números o letras hasta que se presiona el número 0 para salir.
while true; do
    read -p "Ingrese un número o una letra (0 para salir): " input
    if [ "$input" == "0" ]; then
        echo "Saliendo..."
        break
    else
        echo "Ingresaste: $input"
    fi
done

# Script 9: Saludar a cuatro personas que ha ingresado un usuario
# Este script permite ingresar los nombres de cuatro personas y luego las saluda.
personas=()

for i in {1..4}
do
    read -p "Ingrese el nombre de la persona $i: " nombre
    personas+=("$nombre")
done

for nombre in "${personas[@]}"
```

```
do
  echo "Hola, $nombre!"
done


# Script 10: Menú con opciones de buscar un usuario, multiplicar dos números, obtener nombres de los grupos en el sistema y salir
# Este script muestra un menú con opciones para buscar un usuario, multiplicar dos números, obtener los nombres de los grupos en el sistema o salir.
while true; do
  echo "1. Buscar un usuario"
  echo "2. Multiplicar dos números"
  echo "3. Obtener nombres de los grupos en el sistema"
  echo "4. Salir"
  read -p "Elija una opción: " opcion

  case $opcion in
    1)
      read -p "Ingrese el nombre del usuario: " usuario
      if id "$usuario" &>/dev/null; then
        echo "El usuario $usuario existe."
      else
        echo "El usuario $usuario no existe."
      fi
      ;;
    2)
      read -p "Ingrese el primer número: " num1
      read -p "Ingrese el segundo número: " num2
      echo "El resultado de la multiplicación es: $((num1 * num2))"
```

```
        ;;
    3)
        echo "Los nombres de los grupos en el sistema son:"
        cut -d: -f1 /etc/group
        ;;
    4)
        echo "Saliendo..."
        break
        ;;
    *)
        echo "Opción no válida"
        ;;
  esac
done
```

EJERCICIOS REPASO 2024

1. Hacer un script donde entres por teclado el nombre, apellidos y dni de un usuario y lo muestre por pantalla.

```
# Solicitar el nombre
        echo "Introduce tu nombre:"
        read nombre

# Solicitar los apellidos
        echo "Introduce tus apellidos:"
        read apellidos

# Solicitar el DNI
        echo "Introduce tu DNI:"
        read dni
```

Mostrar la información ingresada

```
echo "Nombre: $nombre"

echo "Apellidos: $apellidos"

echo "DNI: $dni"
```

2. Haz un script que muestre todos los números del 1 al 100

```
for i in $(seq 1 100);do
  echo $i
done
```

DE 100 a 1

```
for i in $(seq 100 -1 1);do
  echo $i
Done
```

3. Hacer un script que me pida un número por teclado y me diga si es mayor, menor o igual a 15.

```
echo "Introduce un número:"
```

```
read num

if [ $num -gt 15 ];then
    echo "$num es mayor que 15"
else
    echo "$num es menor que 15"
fi
```

4. Hacer un programa que compruebe que la palabra entrada por teclado es un directorio de home de usuario, y en caso contrario que me lo diga.

```
# Solicitar el nombre del directorio
    echo "Introduce el nombre de un directorio en tu home:"
    read dir_name

# Obtener el path del directorio home del usuario actual
    home_dir="$HOME"

# Comprobar si el directorio existe en el directorio home
    if [ -d "$home_dir/$dir_name" ]; then
      echo "$dir_name es un directorio en tu home."
```

```
else
   echo "$dir_name no es un directorio en tu home."
Fi
```

5. Haz un script en el cual tú entras una palabra, si esa palabra es puzzle, que te diga que has entrado la palabra puzzle, en otro caso que te diga que la palabra entrada no es puzzle.

```
echo "Introduce una palabra"
read palabra

while [ $palabra != "puzzle" ];do
    echo "la palabra no es correcta, introduce otra palabra"
    read palabra
done
```

6. Hacer un script que permita obtener la resta de dos números entrados por teclado y me saque el resultado por pantalla

```
# Solicitar el primer número
    echo "Introduce el primer número:"
    read num1

# Solicitar el segundo número
    echo "Introduce el segundo número:"
    read num2
```

```
# Calcular la resta
        resultado=$((num1 - num2))

# Mostrar el resultado
        echo "La resta de $num1 y $num2 es: $resultado"
```

7. Hacer un script que busque una palabra entrada por teclado (por ejemplo money en un fichero de texto que nos inventemos nosotros. Intentar ver si podéis hacer que nos diga que no la ha encontrado en el caso de que exista.

```
        echo "Escribe la palabra a buscar:"
        read palabra

        if grep -q "$palabra" "palabras.txt"; then
           echo "La palabra '$palabra' existe en el fichero."
        else
           echo "La palabra '$palabra' no existe en el fichero."
        Fi
```

8. Usando un for, hacer un programa que me liste 4 directorios

```
# Contador de directorios encontrados
count=0

# Bucle for para iterar sobre los elementos en el directorio actual
for item in *; do
# Comprobar si el elemento es un directorio
```

```
    if [ -d "$item" ]; then
        echo "$item"
        count=$((count + 1))

# Salir del bucle después de listar 4 directorios
        if [ "$count" -ge 4 ]; then
            break
        fi
    fi
done
# Comprobar si no se encontraron suficientes directorios
if [ "$count" -lt 4 ]; then
    echo "Se encontraron menos de 4 directorios."
Fi
```

9. Hacer un menú con las opciones de suma, resta y multiplicación. También tiene que tener una opción de salida.

```
while true; do
    # Mostrar el menú
    echo "Selecciona una opción:"
    echo "1) Suma"
```

```
echo "2) Resta"
echo "3) Multiplicación"
echo "4) Salir"

# Leer la opción del usuario
read -p "Opción: " opcion

# Salir del bucle si la opción es 4
if [ "$opcion" -eq 4 ]; then
   echo "Saliendo..."
   break
fi

# Pedir los números al usuario
read -p "Introduce el primer número: " num1
read -p "Introduce el segundo número: " num2

# Realizar la operación seleccionada
case $opcion in
   1)
      resultado=$((num1 + num2))
      echo "El resultado de la suma es: $resultado"
      ;;
   2)
      resultado=$((num1 - num2))
      echo "El resultado de la resta es: $resultado"
      ;;
   3)
```

```
        resultado=$((num1 * num2))
        echo "El resultado de la multiplicación es: $resultado"
        ;;
    *)
        echo "Opción no válida. Inténtalo de nuevo."
        ;;
  esac
Done
```

10.Hacer un menú donde la primera opción sea crear un directorio, la segunda opción sea listas el directorio actual, la tercera opción sea editar el fichero /etc/passwd, y la 4 opción sea salir.

```
while true; do
```

```
# Mostrar el menú
echo "Selecciona una opción:"
echo "1) Crear un directorio"
echo "2) Listar el directorio actual"
echo "3) Editar el fichero /etc/passwd"
echo "4) Salir"

# Leer la opción del usuario
read -p "Opción: " opcion

case $opcion in
  1)
    # Crear un directorio
    read -p "Introduce el nombre del directorio a crear: " dir_name
    mkdir -p "$dir_name"
    echo "Directorio '$dir_name' creado."
    ;;
  2)
    # Listar el directorio actual
    echo "Contenido del directorio actual:"
    ls -la
    ;;
  3)
    # Editar el fichero /etc/passwd
    echo "Abriendo el fichero /etc/passwd en el editor predeterminado..."
    nano /etc/passwd
    ;;
  4)
```

```
        # Salir
        echo "Saliendo..."
        break
        ;;
    *)
        # Opción no válida
        echo "Opción no válida. Inténtalo de nuevo."
        ;;
  esac

  echo # Línea en blanco para separar las iteraciones del menú
done
```