Ejercicios de shell scripting

disk.sh **Solución**

```
#!/bin/bash

df -h | grep home | tr -s ' '| cut -d' ' -f5-
```

mem.sh **Solución**

```
#!/bin/bash

free -h | grep Mem | tr -s ' ' | cut -d' ' -f4 >> free.log
```

mac.sh **Solución**

```
#!/bin/bash

ip link show enp2s0 | grep link | tr -s ' ' |  cut -d' ' -f3
```

listar_usuario_grupo.sh **Solución**

```
#!/bin/bash
for i in $(cat /etc/passwd | grep bash | grep -v root | cut -d':' -f1)
do
  groups $i
done
```

lastlog_ip.sh **Solución**

```
#!/bin/bash

for i in $(last -w | tr -s ' '  | cut -d' ' -f3 | sort |  grep '\.' | uniq);
do
  echo $(last -w| grep -c $i) $i
done | sort -rn

#last -w | tr -s ' '  | cut -d' ' -f3 | grep '\.' | sort | uniq -c | sort -r
```

lastlog.sh **Solución**

```
#!/bin/bash

for i in $(last -w | sort | cut -d' ' -f1 | uniq | egrep -v reboot | egrep -v
do
  echo $(last -w| grep -c $i) $i  $(groups $i | cut -d':' -f2)
done | sort -rn

#last -w | tr -s ' '  | cut -d' ' -f1 | sort | uniq  -c | sort -r
```

tabla_multiplicar_read.sh **Solución**

```
#!/bin/bash

echo "Introduce un número para mostrar su tabla de multiplicar: "
read numero

for ((i=1;i<11;i++))
do
    resultado=$((numero * i))
    echo "$numero x $i = $resultado"
done
```

tabla_multiplicar.sh **Solución**

```
#!/bin/bash

mostrar_ayuda() {

    echo "Uso: ./tabla_multiplicar.sh [número]
    Este script muestra la tabla de multiplicar del número que se le pase como
    Opciones:
      --help   Muestra este mensaje de ayuda.

    Ejemplos:
    ./tabla_multiplicar.sh 4   Muestra la tabla de multiplicar del 4."
}


if [[ $# -eq 0 ]]
then
    # Sin argumentos: mostrar ayuda
    mostrar_ayuda
elif [[ $1 == "--help" ]]
  then
    mostrar_ayuda
else
    numero=$1
    for ((i=1;i<11;i++))
    do
      resultado=$((numero * i))
      echo "$numero x $i = $resultado"
    done
fi
```

imag.sh **Solución**

```
#!/bin/bash


for i in imag_*
do
  mv $i $(echo $i | cut -d'.' -f1).png
done
```

rnd.sh **Solución**

```
#!/bin/bash

n=${1:-5}

for (( i = 0 ; i < n ; i++ ))
  do
  x=$((RANDOM % 10 + 1))' '
  out=$x' '
  for (( j=0 ; j<x ; j++ ))
    do
      out=${out}A
    done
  echo $out
done
```

adivina.sh **Solución**

```
#!/bin/bash

# Generar un número aleatorio entre 1 y 20
numero_secreto=$((RANDOM % 20 + 1))
echo $numero_secreto
intentos=0

echo "¡Adivina el número entre 1 y 20!"

acertado=0
while [[ $acertado == 0 ]]
do
  intentos=$((intentos+1))
  read -p "Introduce tu número: " respuesta

  # Comparar el número ingresado con el número secreto
  if [[ $respuesta -lt $numero_secreto ]]; then
    echo "El número es más grande."
  elif [[ $respuesta -gt $numero_secreto ]]; then
    echo "El número es más pequeño."
  else
    echo "¡Correcto! El número era $numero_secreto."
    echo "Lo has adivinado en $intentos intento(s)."
    acertado=1
  fi
done
```

notas.sh **Solución**

```bash
#!/bin/bash

archivo="calificaciones.txt"
todos_aprobados=0
un_suspenso=0
dos_suspensos=0
tres_o_mas_suspensos=0

num_alumnos=$(wc -l $archivo | cut -d' ' -f1)

for ((i=1; i<=num_alumnos; i++))
    do
    linea=$(sed -n "${i}p" "$archivo")
    aprobado_1=0
    aprobado_2=0
    aprobado_3=0
    aprobado_4=0
    aprobado_5=0
    nota_1=$(echo $linea |cut -d':' -f2|cut -d',' -f1)
    if [ $nota_1 -ge 5 ]; then aprobado_1=1 ; fi

    nota_2=$(echo $linea |cut -d':' -f2|cut -d',' -f2)
    if [ $nota_2 -ge 5 ]; then aprobado_2=1 ; fi

    nota_3=$(echo $linea |cut -d':' -f2|cut -d',' -f3)
    if [ $nota_3 -ge 5 ]; then aprobado_3=1 ; fi

    nota_4=$(echo $linea |cut -d':' -f2|cut -d',' -f4)
    if [ $nota_4 -ge 5 ]; then aprobado_4=1 ; fi

    nota_5=$(echo $linea |cut -d':' -f2|cut -d',' -f5)
    if [ $nota_5 -ge 5 ]; then aprobado_5=1 ; fi
    sum_aprobados=$((aprobado_1 + aprobado_2 + aprobado_3 + aprobado_4 + aproba

    if [ $sum_aprobados -eq 5 ];  then  todos_aprobados=$((todos_aprobados+1))
    if [ $sum_aprobados -eq 4 ];  then  un_suspenso=$((un_suspenso+1)) ;    fi
    if [ $sum_aprobados -eq 3 ];  then  dos_suspensos=$((dos_suspensos+1)) ;
    if [ $sum_aprobados -lt 3 ];  then  tres_o_mas_suspensos=$((tres_o_mas_susp
    echo -e "$linea : \033[32m$sum_aprobados \033[0mmod. aprobados "

done

porcentaje_aprobados=$(echo "scale=2; ($todos_aprobados / $num_alumnos) * 100"
porcentaje_suspensos=$(echo "scale=2; (($num_alumnos-$todos_aprobados) / $num_a

echo -e "\033[34m-------------- Resultados ---------------- \033[0m"
echo "Número total de alumnos matriculados: $total_alumnos"
echo "Número de alumnos que han aprobado todos los módulos: $todos_aprobados"
echo "Número de alumnos que han suspendido solo un módulo: $un_suspenso"
echo "Número de alumnos que han suspendido dos módulos: $dos_suspensos"
echo "Número de alumnos que han suspendido tres módulos o más: $tres_o_mas_susp
echo "Porcentaje de alumnos aprobados: $porcentaje_aprobados%"
echo "Porcentaje de alumnos suspensos: $porcentaje_suspensos%"
```

usuarios.sh **Solución**

```
#!/bin/bash

function show_help() {
    echo "Uso del script:"
    echo "./usuarios.sh -addgroup <grupo>         : Crea un grupo si no existe"
    echo "./usuarios.sh -delgroup <grupo>         : Elimina un grupo si existe"
    echo "./usuarios.sh -adduser <usuario> [grupo]: Crea un usuario en el grupo
    echo "./usuarios.sh -deluser <usuario>        : Elimina un usuario si exist
    echo "./usuarios.sh -lista                    : Lista todos los grupos y su
}

if [[ $# -gt 0 ]]
then
  if [[ $1 == "-addgroup" ]]
  then
    if [[ $(grep -c $2 /etc/group) -gt 0 ]]
    then
      echo "No se crea el grupo, el grupo $2 existe."
    else
      echo "Creamos el grupo $2"
      groupadd $2
    fi
  fi
  if [[ $1 == "-delgroup" ]]
  then
    groupdel $2
  fi

  if [[ $1 == "-adduser" ]]
  then
    usuario=$2
    grupo=${3:-$usuario}
    echo "creamos el $usuario en $grupo"
    if [[ $(grep -c $usuario /etc/passwd) -gt 0 ]]
    then
      echo "No se crea el usuario, el usuario $usuario ya existe."
    else
      if [[ $(grep -c $grupo /etc/group) -eq 0 ]]
      then
        groupadd $grupo
        echo "Creamos el grupo $grupo"
      fi
      useradd -g $grupo -d /home/$usuario -m -s /bin/bash -p $( mkpasswd -m sha
      echo "Usuario $usuario creado en el grupo $grupo con contraseña por defec
    fi
  fi
  if [[ $1 == "-deluser" ]]
  then
    if [[ $(grep -c $usuario /etc/passwd) -gt 0 ]]
    then
      userdel -r $2
      echo "Usuario $2 eliminado."
    else
      echo "No se borra el usuario, el usuario $2 no existe."
    fi
  fi
  if [[ "$1" == "-lista" ]]
```

```
  then
    #solo listamos los grupos que tienen usuarios
    for g in $(cut -d':' -f1 /etc/group)
    do
      if [[ $(members $g ) != "" ]]
      then
        echo ----- $g------
        members $g
      fi
    done
  fi

else
  show_help
fi
```