

```
#!/bin/bash
# selector_saludo_aleatorio.sh
# Muestra un saludo según la hora con un muñeco aleatorio de cowsay (versión sencilla)
## (1)
```

```
hora=$(date +%H)          # Hora actual 00-23
```

```
# Elegimos saludo según la hora
if [ "$hora" -lt 12 ]; then
    saludo="Buenos días"
elif [ "$hora" -lt 20 ]; then
    saludo="Buenas tardes"
else
    saludo="Buenas noches"
fi
```

```
# Lista de muñecos en una misma línea, shuf saca uno al azar
muneco=$(echo "duck tux koala whale" | tr ' ' '\n' | shuf -n1)
```

```
# Mostramos el mensaje
cowsay -f "$muneco" "$saludo"
```

-----

```
#!/bin/bash
# saludos: dice buenos días / tardes / noches según la hora
## (2)
```

```
hora=$(date +%H) # 00-23
```

```
if [ "$hora" -lt 12 ]; then
    echo "Buenos días"
elif [ "$hora" -lt 20 ]; then
    echo "Buenas tardes"
else
    echo "Buenas noches"
fi
```

-----

```
#!/bin/bash
# quiniela 14 líneas con 1, X o 2
## (3)
```

```
for i in {1..14}; do
    signo=$(shuf -e 1 X 2 -n1) # elijo 1, X o 2 al azar
    echo "$i.- $signo"
done
```

-----

```
#!/bin/bash
# lista_archivos.sh – directorio en curso
# / directorio
# * ejecutable
# - archivo normal
# si hay argumento, entramos en ese directorio [ -n "$1" ] && cd "$1"
## (4)
```

```
[ -n "$1" ] && cd "$1" || cd . # si hay argumento, entra en ese dir, sino, usa el actual
```

```
for a in *; do
    if [ -d "$a" ]; then
        echo "$a/"
    elif [ -x "$a" ]; then
        echo "$a*"
    else
        echo "$a-"
    fi
done
```

```
fi
done
```

```
-----

#!/bin/bash
# cuenta_tipos - cuenta y lista:
# d (directorios)
# e (ejecutables)
# dv (dispositivos)
# t (archivos de texto)
# Uso: cuenta_tipos [dir] → si omities dir, usa "."
## (5)

dir=${1:-.}      # "." si no pasas argumento
[ -d "$dir" ] || exit 1  # si no existe el dir, error 1

echo "Directorio analizado: $dir"
echo

echo "d (directorios):"
ls -1p "$dir" | grep '/'$'      # lista
echo "Total d : $(ls -1p "$dir" | grep -c '/'$)"
echo

echo "e (ejecutables):"
ls -1p "$dir" | grep '\*$'      # lista
echo "Total e : $(ls -1p "$dir" | grep -c '\*$')"
echo

echo "dv (dispositivos):"
ls -1 "$dir" | grep -E '^[bc]'  # lista
echo "Total dv : $(ls -1 "$dir" | grep -c -E '^[bc]')"
echo

echo "t (archivos de texto):"
for f in "$dir"/*; do          # lista
    file -b "$f" | grep -qi text && echo "$(basename "$f")"
done
echo "Total t : $(for f in "$dir"/*; do file -b "$f" | grep -qi text && echo; done | wc -l)"

exit 0
```

```
-----

#!/bin/bash
# crear.sh [nombre] [tamañoKB]
# - Si NO pongo nombre, usará fichero_vacio
# - Si NO pongo tamaño, usará 1024 (KB)
## (6)

# 1) nombre
nombre=$1
if [ -z "$nombre" ]; then      # -z → la variable está VACÍA
    nombre=fichero_vacio
fi

# 2) tamaño
tam=$2
if [ -z "$tam" ]; then
    tam=1024                    # 1 MB
fi

# 3) crear el archivo del tamaño pedido
truncate -s "$tam"K "$nombre" # K = kilobytes
```

```
echo "Creado: $nombre ($tam KB)"
```

```
-----  
#!/bin/bash  
## (7)
```

```
n=${1:-fichero_vacio}  
k=${2:-1024}  
for i in "" 1 2 3 4 5 6 7 8 9; do  
    f="$n$i"  
    [ ! -e "$f" ] && { truncate -s "${k}K" "$f"; exit 0; }  
done  
exit 1
```

```
-----  
#!/bin/bash  
# Uso: mayormenorsuma.sh n1 n2 n3 ...  
## (8)
```

```
M=$1 # menor (de momento)  
m=$1 # mayor (de momento)  
s=0  # suma
```

```
for n in "$@"; do      # recorre todos los números dados  
    (( s+=n ))          # suma total  
    (( n < m )) && m=$n  # actualiza menor  
    (( n > M )) && M=$n  # actualiza mayor  
done
```

```
echo "Menor : $m"  
echo "Mayor : $M"  
echo "Suma  : $s"
```

```
-----  
#!/bin/bash  
## (9)
```

```
read -p "Introduce tu año de nacimiento: " anio    # mensaje de entrada
```

```
case $((anio%12)) in          # resto 0-11  
    0) animal="Mono" ;;  
    1) animal="Gallo" ;;  
    2) animal="Perro" ;;  
    3) animal="Cerdo" ;;  
    4) animal="Rata" ;;  
    5) animal="Buey" ;;  
    6) animal="Tigre" ;;  
    7) animal="Conejo" ;;  
    8) animal="Dragón" ;;  
    9) animal="Serpiente" ;;  
   10) animal="Caballo" ;;  
   11) animal="Cabra" ;;  
esac
```

```
echo "Si naciste en $anio te corresponde $animal según el horóscopo chino."
```

```
-----  
#!/bin/bash  
## (10)
```

```
while ;; do  
    read -p "Nombre: " n
```

```

read -p "Dirección: " d
read -p "Teléfono: " t
echo "$n | $d | $t" >> agenda.txt
echo "Añadido."

read -p "Enter para continuar / FIN para salir: " x
[ "$x" = FIN ] && break      # si escribes FIN, sal del bucle
done

```

```

-----

#!/bin/bash
## (11)

f=agenda.txt      # archivo donde se guarda todo
touch "$f"        # lo crea si aún no existe

while true; do
    echo
    echo "1) Añadir  2) Borrar  3) Buscar  4) Listar  5) Salir"
    read -p "Elige una opción (1-5): " op

    if [ "$op" = "1" ]; then
        read -p "Nombre: " n
        read -p "Dirección: " d
        read -p "Teléfono: " t
        echo "$n|$d|$t" >> "$f"
        echo "Contacto añadido."

    elif [ "$op" = "2" ]; then
        read -p "Nombre a borrar: " n
        grep -v "$n|" "$f" > tmp && mv tmp "$f"
        echo "Si existía, se eliminó."

    elif [ "$op" = "3" ]; then
        read -p "Texto a buscar: " t
        grep "$t" "$f"

    elif [ "$op" = "4" ]; then
        cat "$f"

    elif [ "$op" = "5" ]; then
        break      # salir del menú

    else
        echo "Opción no válida."
    fi
done

```

```

-----

#!/bin/bash
# Uso: buscapalabras.sh fichero depalabras fichero debusqueda
## (12)

[ $# -ne 2 ] && { echo "Uso: $0 fichero depalabras fichero debusqueda"; exit 1; }

fichero depalabras=$1      # lista de palabras
fichero debusqueda=$2      # texto donde buscar

while read palabra; do
    [ -z "$palabra" ] && continue      # salta líneas vacías
    lineas=$(grep -i "$palabra" "$fichero debusqueda" | wc -l)
    echo "$palabra $lineas"
done < "$fichero depalabras"

```

```

-----

#!/bin/bash
# Uso: ./informa.sh -m|-a DNI
# -m → sueldo mensual
# -a → sueldo anual (12 sueldos + 2 pagas extra)
## (13)

file=bdreparto
[ $# -ne 2 ] && { echo "Uso: $0 -m|-a DNI"; exit 1; }

op=$1      # -m o -a
dni=$2

# 1) Buscar la línea del empleado
line=$(grep "$dni" "$file") || { echo "DNI no encontrado"; exit 1; }

# 2) Extraer campos (separador ':')
nombre=$(echo "$line" | cut -d: -f2)
apellido=$(echo "$line" | cut -d: -f3)
sueldo=$(echo "$line" | cut -d: -f5) # 5º campo = sueldo
puesto=$(echo "$line" | cut -d: -f6) # 6º campo = puesto

if [ "$op" = "-m" ]; then          # — opción mensual
    echo "$nombre $apellido cobra $sueldo € al mes"
elif [ "$op" = "-a" ]; then        # — opción anual
    if [ "$puesto" = "Productor" ]; then extra=1500
    elif [ "$puesto" = "Director" ]; then extra=900
    elif [ "$puesto" = "Actor principal" ]; then extra=854
    elif [ "$puesto" = "Actor" ]; then extra=700
    elif [ "$puesto" = "Guionista" ]; then extra=500
    else extra=0
    fi
    anual=$(( sueldo * 12 + extra * 2 ))
    echo "$nombre $apellido cobra $anual € al año"
else
    echo "Opción no válida (usa -m o -a)"
fi

```

```

-----

#!/bin/bash
## (14)

dest="$HOME/copia"
mkdir -p "$dest"

for f in *; do
    if [ -f "$f" ]; then          # si es archivo normal
        cp -n "$f" "$dest"      # -n = no sobrescribir si ya existe
        echo "Copiado: $f"
    elif [ -d "$f" ]; then        # si es directorio
        echo "Directorio ignorado: $f"
    fi
done

```