

## **TAREA 1. Q2**

---

**Silvia Rosúa Vargas**

---

Módulo: **Administración de Sistemas Operativos**

---

Curso: **2024-2025**

---

Profesor/a: **José Manuel Nebot Bosch**

---

Fecha entrega: **Q1 – 23/09/2024 a 07/10/2024**

## ÍNDICE

TAREA 1 .....	3
Ejercicio 1 .....	3
Ejercicio 2 .....	5
Ejercicio 3 .....	6
Ejercicio 4 .....	7
Ejercicio 5 .....	8

## TAREA 1

### Ejercicio 1

Realiza el script “ejer1.sh” que permita un número indeterminado de parámetros de entrada, donde cada uno sea el nombre de un fichero. El script ha de comprobar:

1. Que se le ha pasado al menos un parámetro de entrada, y en caso contrario muestre por pantalla:

“Usa: ejer1 fichero1 fichero2 ...”

2. Que los ficheros indicados existen, y en caso contrario indique: “El fichero XXXXX no existe.”

Utilizo \$@ para que trate cada parámetro de forma individual, ya que con \$\* los trataría como una cadena individual.

3. Indique si un fichero es ejecutable o no, mostrando por pantalla:

“El fichero XXXXX es ejecutable.”

“El fichero YYYYYY no es ejecutable”



```
1 #!/bin/bash
2 #
3 #Autora: Silvia Rosúa
4 #
5 #Fecha: 29/09/2024
6 #
7 #Descripción: Realiza un un script que permita un número indeterminado de entradas, donde cada
  uno sea le nombre de un archivo.
8 #
9 #Comprobación 1: que se le ha pasado al menos un parámetro de entrada, y en caso contrario que
  muestre "usa:ejercicio 1 fichero 1 fichero 2"
10
11 if [ $# -eq 0 ] ; then
12     echo "Usa:ejercicio 1 fichero 1 fichero 2"
13     exit 1
14
15 fi
16
17 #Comprobación 2: que los ficheros indicados existen, y en caso contrario que indique "El
  fichero xxxx no existe"
18
19 for fichero in "$@"; do
20     #comprobar si el fichero existe
21     if [ -e "$fichero" ]; then
22         #comprobar si el fichero es ejecutable
23         if [ -x "$fichero" ]; then
24             echo "El fichero $fichero es ejecutable."
25         else
26             echo "El fichero $fichero no es ejecutable"
27         fi
28     else
29         echo "El fichero $fichero no existe"
30     fi
31 done
```

```
silvia@silvia:~$ chmod +x ejer1.sh
```

```
silvia@silvia:~$ ./ejer1.sh
Usa: ejer 1 fichero 1 fichero 2
silvia@silvia:~$ ./ejer1.sh fichero1 fichero2
El fichero fichero1 no existe
El fichero fichero2 no existe
silvia@silvia:~$ touch fichero1.txt fichero2.txt
silvia@silvia:~$ ./ejer1.sh fichero1 fichero2
El fichero fichero1 no existe
El fichero fichero2 no existe
silvia@silvia:~$ ./ejer1.sh fichero1.txt fichero2.txt
El fichero fichero1.txt no es ejecutable
El fichero fichero2.txt no es ejecutable
silvia@silvia:~$
```

## Ejercicio 2

Crear el script “ejer2.sh” que acepta como argumentos dos ficheros de texto y calcula cual es el fichero que tiene mayor número de líneas. La salida del script indicará si un fichero tiene más, menos o el mismo número de líneas junto con el número de éstas para cada fichero. En caso de no existir alguno de los ficheros, debe indicarlo. Si los ficheros no se pasan con el formato especificado también debe indicarlo.

```
1 #!/bin/bash
2 #
3 #Autora: Silvia Rosúa
4 #
5 #Fecha: 29/09/2024
6 #
7 #Descripción: Realiza un un script que acepta como argumentos dos ficheros de texto y
8 #calcula cual es el fichero que tiene mayor número de líneas.
9 #Comprobación 1: verificar si se han pasado exactamente dos argumentos"
10
11 if [ "$#" -ne 2 ]; then
12     echo "No se han pasado dos ficheros"
13     exit 1
14 fi
15
16 #Comprobación 2: verificar si los ficheros existen
17
18 if [ ! -f "$1" ]; then
19     echo "El fichero \"$1\" no existe"
20     exit 1
21 fi
22
23 if [ ! -f "$2" ]; then
24     echo "El fichero \"$2\" no existe"
25     exit 1
26 fi
27
28 #Comprobación 3: contar el número de líneas de cada fichero
29
30 lineas1=$(wc -l < "$1")
31 lineas2=$(wc -l < "$2")
32
33 #Comprobación 4: comparar el número de líneas y mostrar el resultado
34
35 if [ "$lineas1" -gt "$lineas2" ]; then
36     echo "$1" con "$lineas1" líneas es mayor que "$2" con "$lineas2"
37
38 elif [ "$lineas1" -lt "$lineas2" ]; then
39     echo "$1" con "$lineas1" líneas es menor que "$2" con "$lineas2"
40
41 else
42     echo "$1" con "$lineas1" líneas es igual que "$2" con "$lineas2"
43
44 fi
```

```
silvia@silvia:~$ gedit ejer2.sh
silvia@silvia:~$ chmod -x ejer2.sh
silvia@silvia:~$ ./ejer2.sh fichero1.txt fichero2.txt
bash: ./ejer2.sh: Permiso denegado
silvia@silvia:~$ chmod +x ejer2.sh
silvia@silvia:~$ ./ejer2.sh fichero1.txt fichero2.txt
fichero1.txt con 0 líneas es igual que fichero2.txt con 0
silvia@silvia:~$
```

### Ejercicio 3

Realizar un script que sume los tamaños de todos los ficheros que se le pasen como argumentos, dando un mensaje de error para aquellos argumentos que no existan o que no sean ficheros. Se entenderá que dichos ficheros se encuentran en el directorio actual.

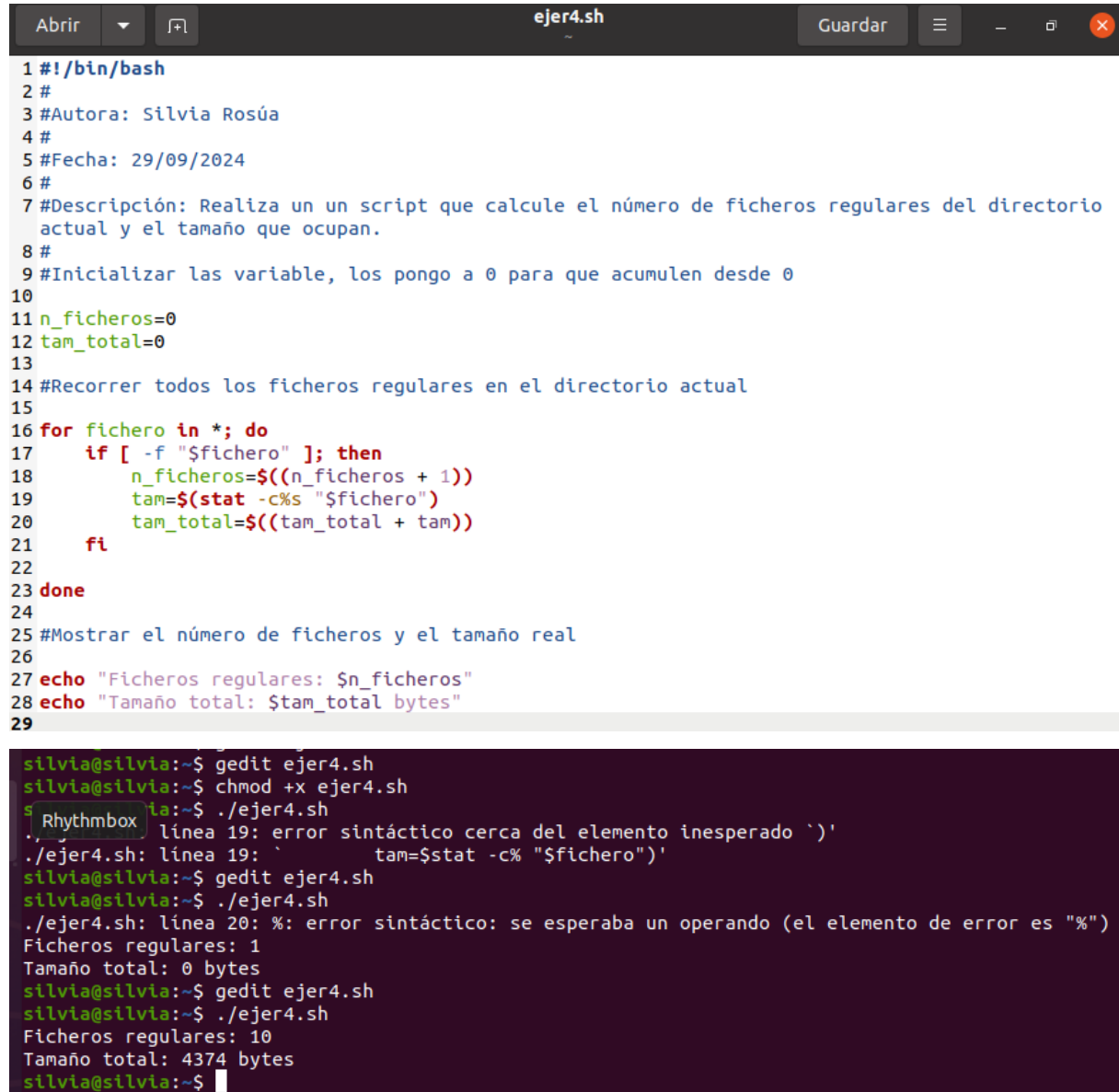
```
1#!/bin/bash
2#
3#Autora: Silvia Rosúa
4#
5#Fecha: 29/09/2024
6#
7#Descripción: Realiza un un script que sume los tamaños de l todos los ficheros que se le pasen
  como argumentos dando mensaje de error para los argumentos que no existan o que nos sean
  ficheros.
8#
9#Inicializar la variable para el tamaño total, se inicia en 0 para acumular el tamaño de los
  ficheros
10 tamaño_total=0
11
12#Recorrer todos los argumentos pasados al script
13 for fichero in "$@"; do
14     #Verificar si el argumento es un fichero
15     if [ -f "$fichero" ]; then
16         #Obtener el tamaño del fichero en bytes y sumarlo al total
17         tamaño=$(stat -c%s "$fichero")
18         echo "tamaño de '$fichero': $tamaño"
19         tamaño_total=$((tamaño_total + tamaño))
20     else
21         #Mostrar un mensaje de error si el argumento no es un fichero o no existe
22         echo "El '$fichero' no existe o no es un fichero."
23     fi
24 done
25
26#Mostrar el tamaño total de los ficheros
27 echo "El tamaño total de los ficheros es: $tamaño_total bytes."
28
```

```
silvia@silvia:~$ ./ejer3.sh fichero1.txt fichero2.txt
tamaño de fichero1.txt: 22
tamaño de fichero2.txt: 619
El tamaño total de los ficheros es: 641 bytes.
```

```
silvia@silvia:~$ ./ejer3.sh fichero1.txt fichero2
tamaño de fichero1.txt: 22
El 'fichero2' no existe o no es un fichero.
El tamaño total de los ficheros es: 22 bytes.
silvia@silvia:~$
```

## Ejercicio 4

Realiza un script que calcule el número de ficheros regulares del directorio actual y el tamaño total que ocupan.



```
Abrir  ejer4.sh  Guardar  -  +  x

1 #!/bin/bash
2 #
3 #Autora: Silvia Rosúa
4 #
5 #Fecha: 29/09/2024
6 #
7 #Descripción: Realiza un un script que calcule el número de ficheros regulares del directorio
  actual y el tamaño que ocupan.
8 #
9 #Inicializar las variable, los pongo a 0 para que acumulen desde 0
10
11 n_ficheros=0
12 tam_total=0
13
14 #Recorrer todos los ficheros regulares en el directorio actual
15
16 for fichero in *; do
17     if [ -f "$fichero" ]; then
18         n_ficheros=$((n_ficheros + 1))
19         tam=$((stat -c%s "$fichero"))
20         tam_total=$((tam_total + tam))
21     fi
22
23 done
24
25 #Mostrar el número de ficheros y el tamaño real
26
27 echo "Ficheros regulares: $n_ficheros"
28 echo "Tamaño total: $tam_total bytes"
29
```

```
silvia@silvia:~$ gedit ejer4.sh
silvia@silvia:~$ chmod +x ejer4.sh
silvia@silvia:~$ ./ejer4.sh
./ejer4.sh: línea 19: error sintáctico cerca del elemento inesperado `)'
./ejer4.sh: línea 19: `        tam=$stat -c% "$fichero")'
silvia@silvia:~$ gedit ejer4.sh
silvia@silvia:~$ ./ejer4.sh
./ejer4.sh: línea 20: %: error sintáctico: se esperaba un operando (el elemento de error es "%")
Ficheros regulares: 1
Tamaño total: 0 bytes
silvia@silvia:~$ gedit ejer4.sh
silvia@silvia:~$ ./ejer4.sh
Ficheros regulares: 10
Tamaño total: 4374 bytes
silvia@silvia:~$
```

## Ejercicio 5

Realiza un script, que dado el login de un usuario como parámetro, muestre el nombre y el tamaño de cualquier directorio de la jerarquía de directorios de dicho usuario. Al finalizar deberá mostrar la suma de todos los tamaños mostrados. En caso de que dicho usuario no exista, debe mostrar un error indicándolo

```
Abrir  ejer5.sh  Guardar  -  +  X
1 #!/bin/bash
2 #
3 #Autora: Silvia Rosúa
4 #
5 #Fecha: 29/09/2024
6 #
7 #Descripción: Realiza un un script que dado el login de usuario como parámetro, muestre el
8 # nombre y tamaño de cualquier directorio de la jereanquia de direcrorios del usuario. Debe
9 # mostrar la suma de todos los tamaños mostrados. si el usuario no existe debe mostrar un error
10 # indicándolo.
11 #
12 #Verificar si se ha pasado un argumento
13 if [ "$#" -ne 1 ]; then
14     echo "Se debe pasar el login de un usuario como argumento."
15     exit 1
16 fi
17 #Obtener el login del usuario
18 usuario=$1
19 #Verificar si el usuario existe utilizo el comando id
20 if ! id "$usuario" &>/dev/null; then
21     echo "El usuario '$usuario' no existe."
22     exit 1
23 fi
24 #Obtener el directorio home del usuario
25 home_dir=$(eval echo ~$usuario)
26 #Inicializar variables
27 tam_total=0
28 #Recorrer todos los directorios en la jerarquia del directorio home del usuario
29 while IFS= read -r -d '' directorio; do
30     # Obtener el tamaño del directorio
31     tam=$(du -sb "$directorio" | cut -f1)
32     echo "Directorio: $directorio, Tamaño: $tam bytes"
33     tam_total=$((tam_total + tam))
34 done < <(find "$home_dir" -type d -print0)
35 #Mostrar el tamaño total de todos los directorios
36 echo "Tamaño total de todos los directorios: $tam_total bytes"
37
```

```
silvia@silvia:~$ gedit ejer5.sh
silvia@silvia:~$ chmod +x ejer5.sh
silvia@silvia:~$ ./ejer5.sh
Se debe pasar el login de un usuario como argumento.
silvia@silvia:~$ ./ejer5.sh rosua
El usuario 'rosua' no existe.
silvia@silvia:~$ ./ejer5.sh silvia
Directorio: /home/silvia, Tamaño:  bytes
Directorio: /home/silvia/.local, Tamaño:  bytes
Directorio: /home/silvia/TRABAJOS/TEXTOS, Tamaño:  bytes
Tamaño total de todos los directorios: 590952808 bytes
silvia@silvia:~$
```