

---

# U4 ADMINISTRACIÓN Y ASEGURAMIENTO DE LA INFORMACIÓN

---

ACTIVIDAD PRÁCTICA 14: Automatización (SCRIPTS)



25 DE ABRIL DE 2023  
SISTEMAS INFORMÁTICOS  
Sergio Cobo García

# Contenido

Ejercicio 1.....	2
Ejercicio 2.....	2
Ejercicio 3.....	2
Ejercicio 4.....	3
Ejercicio 5.....	4
Ejercicio 6.....	4
Ejercicio 7.....	5
Ejercicio 8.....	6
Ejercicio 9.....	8
Ejercicio 10 .....	8
Ejercicio 11 .....	10
Ejercicio 12 .....	10
Ejercicio 13 .....	11

## Ejercicio 1

01	El comando test dispone de un largo número de funciones que permiten hacer comparaciones. Llene las tres tablas siguientes, respectivas a los tres tipos de operadores de comparación de test. Indique qué valor tomaría la variable ?, en función de los valores introducidos (recuerde que 0 es verdadero en Linux).
----	--

Comparación numérica	
test n1 -eq n2	? valdría 0 si n1 es igual que n2
test n1 -ge n2	? valdría 0 si n1 es mayor o igual que n2
test n1 -gt n2	? valdría 0 si n1 es mayor que n2
test n1 -le n2	? valdría 0 si n1 es menor o igual que n2
test n1 -lt n2	? valdría 0 si n1 es menor que n2
test n1 -ne n2	? valdría 0 si n1 no es igual que n2

Comparación de cadenas de texto	
test s1 = s2	? valdría 0 si s1 es igual que s2
test s1 != s2	? valdría 0 si s1 es no es igual que s2
test -n s1	? valdría 0 si s1 no está vacío
test -z s1	? valdría 0 si s1 está vacío

Comparación de archivos	
test -d f1	? valdría 0 si f1 existe y es un directorio
test -e f1	? valdría 0 si f1 existe
test -f f1	? valdría 0 si f1 existe y es un archivo regular
test -r f1	? valdría 0 si f1 existe y tiene permisos de lectura
test -s f1	? valdría 0 si f1 existe y el tamaño es mayor que 0
test -w f1	? valdría 0 si f1 existe y tiene permisos de escritura
test -x f1	? valdría 0 si f1 existe y tiene permisos de ejecución
test f1 -nt f2	? valdría 0 si la fecha de modificación de f1 es más reciente que la de f2
test f1 -ot f2	? valdría 0 si la fecha de modificación de f1 es más antigua que la de f2

## Ejercicio 2

02	Explique el funcionamiento de las comillas de este ejercicio, y por qué se utilizan.
----	--

Echo "\$comando": ejecuta el valor de la variable comando, en este caso se lleva a cabo el comando ls.

Echo ` \$comando `: ejecuta el contenido de la variable comando (ls).

Echo ' \$comando ': muestra literalmente el texto escrito entre las comillas simples.

## Ejercicio 3

03	Indique justificadamente cuál es la función del siguiente shellscrip, indicando cuál es el significado más lógico de los parámetros. Escribid un comando equivalente a todo este shellscrip.
----	---

Se crea el archivo tmp. Seguidamente se inicia un bucle para que en cada archivo .txt del directorio actual busque la palabra “examen” y si la encuentra añade la línea que la contiene a tmp. A continuación, cuenta el número de líneas en el archivo tmp y lo imprime. Finalmente elimina el archivo tmp.

Un comando que hace lo mismo que el shellscrip es el siguiente: **grep “examen” \*.txt | wc -l**

## Ejercicio 4

04	Escriba un shellscrip que a partir de un parámetro numérico N, ordene alfabéticamente los logins de usuario del sistema y muestre los N últimos.
----	--

```
#EJERCICIO 4
#!/bin/bash

# Verificar si el número de parámetros es correcto
if [ "$#" -ne 1 ];
then
    echo "Número de parámetros incorrecto"
    exit 1
fi

# Obtener la lista de nombres de usuario, ordenarla alfabéticamente y
# mostrar los últimos N usuarios
cat /etc/passwd | sort -t: -k1 | cut -d: -f1 | tail -n $1

exit 0;
```

```
sergio@sergio-VirtualBox:~/Escritorio/PRACTICA 14$ ./ejercicio4.sh 15
sync
syslog
systemd-network
systemd-oom
systemd-resolve
systemd-timesync
sys
tcpdump
tss
usbmux
uucp
uuid
vboxadd
whoopsie
www-data
```

## Ejercicio 5

---

05	Escriba un shellscript que indique si nos encontramos en los primeros o últimos seis meses del año. Tenga en cuenta que el sistema puede estar en cualquier idioma, por lo tanto, utilice los parámetros del comando "date" para obtener un valor válido para cualquier idioma.
----	---

```
#EJERCICIO 5
#!/bin/bash

#Obtener el mes actual
mes=`date +%m`;

#Compruebo si es mayor o menos a 6 el número de mes
if (test $mes -le 6)
then
    echo "Está en los primeros 6 meses";
else
    echo "Está en los últimos 6 meses";
fi
```

```
sergio@sergio-VirtualBox:~/Escritorio/PRACTICA 14$ ./ejercicio5.sh
Está en los primeros 6 meses
```

## Ejercicio 6

---

06	Escriba un shellscript simple que, a partir de un número indeterminado de argumentos, salude a cada uno de los argumentos pasados.
----	--

```
#EJERCICIO 6
#!/bin/bash

# Verificar si el número de parámetros es correcto
if [ "$#" -eq 0 ];
then
    echo "Número de parámetros incorrecto"
    exit 1
fi

#Imprimo la respuesta a cada uno de ellos
for i in $*
do
    echo "Hola $i"
done;

exit 0;
```

```
sergio@sergio-VirtualBox:~/Escritorio/PRACTICA 14$ ./ejercicio6.sh Sergio
Antonio Paula Rigoberta
Hola Sergio
Hola Antonio
Hola Paula
Hola Rigoberta
```

## Ejercicio 7

07	<p>Escriba un shellscript que, a partir de un único parámetro N, y utilizando el bucle "while", muestre por pantalla una progresión aritmética de N términos (1, 2, 3, 4...) y una progresión geométrica de N términos (1, 2, 4, 8, 16...). El número de términos de las sucesiones será el primer parámetro de este shellscript.</p>
----	---

Progresión aritmética de N términos.

```
#EJERCICIO 7
#!/bin/bash

# Verificar si el número de parámetros es correcto
if [ "$#" -ne 1 ]; then
    echo "Número de parámetros incorrecto"
    exit 1
fi

numero=1

#Imprimo numero si este es <= maximo
while (test $numero -le $1)
do
    echo $numero

    #Sumo 1 a numero
    numero=`expr $numero + 1`
done;
```

```
sergio@sergio-VirtualBox:~/Escritorio/PRACTICA 14$ ./ejercicio7_1.sh 5
1
2
3
4
5
```

Progresión geométrica de N términos.

```
#!/bin/bash

# Verificar si el número de parámetros es correcto
if [ "$#" -ne 1 ]; then
    echo "Número de parámetros incorrecto"
    exit 1
fi

numero=1

#Imprimo numero si este es <= maximo
while (test $numero -le $1)
do
    echo $numero

    #Multiplico numero por 2
    numero=`expr $numero \* 2`
done;
```

```
sergio@sergio-VirtualBox:~/Escritorio/PRACTICA 14$ ./ejercicio7_2.sh 40
1
2
4
8
16
32
```

## Ejercicio 8

08	<p>Escriba un shellscript que, utilizando el bucle "while", muestre el factorial de un número por pantalla. A continuación, haga lo mismo con un bucle "until". El número sobre el cual calcular el factorial será el único parámetro de este shellscript (\$1).</p> <p>Recuerde que el factorial de un número es el producto de todos los números naturales desde 1 hasta ese número. Por ejemplo, el factorial de 6 (6!) será igual a <math>6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1</math>.</p>
----	---

Con while

```
#EJERCICIO 8
#!/bin/bash

# Verificar si el número de parámetros es correcto
if [ "$#" -ne 1 ]; then
    echo "Número de parámetros incorrecto"
    exit 1
fi

factorial=1
i=1

#Compruebo si i <= numero y si es >= 0
while (test $i -le $1 && test $i -ge 0)
do
    #Factorial es el valor de i * el valor de factorial
    factorial=`expr $i \* $factorial`

    #Sumo 1 a la variable i
    i=`expr $i + 1`
done
echo "El factorial de $1 es $factorial"
```

```
sergio@sergio-VirtualBox:~/Escritorio/PRACTICA 14$ ./ejercicio8_1.sh 6
El factorial de 6 es 720
```

Con until

```
#!/bin/bash

# Verificar si el número de parámetros es correcto
if [ "$#" -ne 1 ]; then
    echo "Número de parámetros incorrecto"
    exit 1
fi

factorial=1
i=1

#Compruebo si i no es mayor que numero y si no es menor que 0
until [ $i -gt $1 -o $i -lt 0 ]
do
    #Factorial es el valor de i * el valor de factorial
    factorial=`expr $i \* $factorial`

    #Sumo 1 a la variable i
    i=`expr $i + 1`
done

# Mostrar el resultado al usuario
echo "El factorial de $1 es $factorial"
```

```
sergio@sergio-VirtualBox:~/Escritorio/PRACTICA 14$ ./ejercicio8_2.sh 9
El factorial de 9 es 362880
```



## Ejercicio 9

09	Escriba un shellscript que intente autodestruirse (como archivo). Antes de ejecutar este shellscript, haga una copia de seguridad de su trabajo, por si acaso. No puede utilizar el nombre del archivo para eliminarlo. Comente el comportamiento del shellscript y si se borra o no el archivo.
----	--

```
#EJERCICIO 9
#!/bin/bash

# Verificar si el número de parámetros es correcto
if [ "$#" -ne 0 ]; then
    echo "Número de parámetros incorrecto"
    exit 1
fi

#Copio el archivo ejercicio9.sh y le hago una copia
cp ./${0} ./${0}.bak

echo "Se ha creado una copia de seguridad de su trabajo"

#Elimino el archivo ejercicio9.sh
rm ${0};

exit 0;
```

```
sergio@sergio-VirtualBox:~/Escritorio/PRACTICA 14$ ./ejercicio9.sh
Se ha creado una copia de seguridad de su trabajo
```



## Ejercicio 10

10	<p>Escriba un shellscript que, dados dos números diferentes pasados como parámetros, muestre por pantalla los números que van desde uno hasta el otro (ambos incluidos, crecientemente o decrecientemente). Debe resolver el ejercicio con un bucle until.</p>
----	--

```
#EJERCICIO 10
#!/bin/bash

# Verificar si el número de parámetros es correcto
if ( test $# -ne 2 )
then
    echo "Número de parámetros incorrecto"
    exit 1
fi

numero1=$1
numero2=$2
menor=0
mayor=0

#Compruebo que número es menor
if (test $numero1 -lt $numero2)
then
    menor=$numero1
    mayor=$numero2
elif (test $numero1 -gt $numero2)
then
    menor=$numero2
    mayor=$numero1
else
    echo "Los números son iguales"
fi

#Imprimo desde el menor hasta el mayor
until (test $menor -gt $mayor)
do
    echo $menor
    menor=`expr $menor + 1`
done;
```

```
sergio@sergio-VirtualBox:~/Escritorio/PRACTICA 14$ ./ejercicio10.sh 4 9
4
5
6
7
8
9
```

## Ejercicio 11

11	Escriba un shellscript que, a partir del archivo /etc/passwd, diga cuál es el intérprete de comandos más utilizado por los usuarios del sistema (es decir, aquel que utilizan más usuarios). Este shellscript se puede resolver con una única (y larga) comando y sin utilizar ningún tipo de bucle.
----	--

```
#EJERCICIO 11
#!/bin/bash

interprete_mas_usado=$(cut -d: -f7 /etc/passwd | sort |
uniq -c | sort -rn | head -n1 | tr -s ' ' | cut -f2 | cut -d/ -f4)

echo "El intérprete de comandos más usado es \"$interprete_mas_usado\""
```

```
sergio@sergio-VirtualBox:~/Escritorio/PRACTICA 14$ ./ejercicio11.sh
El intérprete de comandos más usado es "nologin"
```

## Ejercicio 12

12	Elabore un shellscript que reciba como parámetros un número indeterminado de palabras en minúscula. Lo que tendrá que hacer el shellscript será agregar cada una de las palabras a archivos que se llamarán como su inicial.
----	--

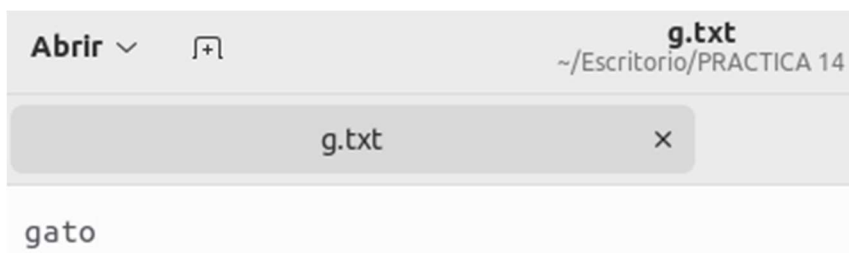
```
#EJERCICIO 12
#!/bin/bash

# Verificar si el número de parámetros es correcto
if [ "$#" -eq 0 ]; then
    echo "Número de parámetros incorrecto"
    exit 1
fi

#Recorre cada palabra
for palabra in "$@"
do
    #Cambiamos de mayúscula a minúscula y obtiene la inicial de la
    palabra
    palabra=`echo $palabra | tr '[:upper:]' '[:lower:]'`
    inicial=$(echo $palabra | cut -c1)

    #Agrega la palabra al archivo que corresponde
    echo $palabra >> ${inicial}.txt
done
echo "Ha funcionado"
```

```
sergio@sergio-VirtualBox:~/Escritorio/PRACTICA 14$ ./ejercicio12.sh pato
gato perro
Ha funcionado
```



## Ejercicio 13

13	Indique justificadamente cuál es la función del siguiente shellsript, indicando cuál es el significado más lógico de los parámetros.
----	--

El primer parámetro recibe el contenido del archivo que tiene rutas a ficheros, el segundo parámetro es la palabra que queremos buscar. Finalmente, si encuentra la palabra imprime A y lo copiará en el directorio /tmp sino imprime B.