

<b>Actividad</b>	<b>Sistemas informáticos</b>
<b>Práctica (AP)</b>	U5.AP2. Shell scripts con ficheros

**Objetivos de la actividad:**

Los objetivos de la actividad son:

- Implementar shell scripts que trabajen con ficheros

**Temporalización:**

La duración prevista para dicha actividad es de 5 sesiones de 55 minutos y/o trabajo en casa

**Enunciado de la actividad:**

A continuación, se presentan **2 grupos de 4 ejercicios** (grupo I y grupo II) **de los que solo tendremos que entregar uno de ellos**, mediante la siguiente regla:

*“Si el resto de la división entera entre el número de letras de mi primer apellido y el número 2 es igual a:*

*1, entonces haré los ejercicios del grupo I*

*0, entonces haré los ejercicios del grupo II”*

Por ejemplo, si mi primer apellido es Barceló:

Barceló → 7 letras

$7 \% 2 = 1$

→ Haré los ejercicios del grupo I

Por ejemplo, si mi primer apellido es González:

González → 8 letras

$8 \% 2 = 0$

→ Haré los ejercicios del grupo II

**Desarrolla los 4 ejercicios (programas en Shell Script) del grupo de ejercicios asignado de forma autónoma. Deberás entregar los archivos .sh resultantes dentro de un archivo .zip.**

**GRUPO I**

1. Escribe un Shell script que cree una estructura de directorios similar a la siguiente para los nombres que se encuentren en el fichero **nombres.txt**, y

teniendo en cuenta un número **X** de subdirectorios personal que se deben crear (para el caso de la imagen, X=3). El valor de la X será introducido por el usuario como parámetro a la llamada del script. Por ejemplo, si queremos crear 3 subdirectorios realizaríamos la siguiente llamada: **./script.sh 3**

```
ana
├── personal1
├── personal2
└── personal3
andrea
├── personal1
├── personal2
└── personal3
manolo
├── personal1
├── personal2
└── personal3
pedro
├── personal1
├── personal2
└── personal3
sergio
├── personal1
├── personal2
└── personal3
```

2. Escribe un script que muestre un listado de los números pares que hay en el fichero **numeros.txt** así como un listado de los números impares. Además, deberá mostrar el total de cada tipo.
3. Escribe un script que procese el fichero **listado.txt**. Cada línea de este fichero tiene un usuario, un sistema operativo y un valor numérico que indica el número de procesos que tiene en ejecución ese usuario en este sistema. Lo que se pretende que haga el script es que, recorra las líneas del fichero y calcule cuántos usuarios tienen Linux, cuántos Windows y el total de procesos de cada sistema. El resultado del script debería ser algo como:

```
Linux-> 10 158
Windows-> 9 243
juanmi@sobremesa:~/practica_scripts$
```

4. Escribe un script que cree un listado de todas las direcciones IP que están conectadas en nuestra red local. Para ello, debemos utilizar el comando **ping** que nos devuelve cuántos paquetes han sido enviados correctamente. Podemos enviar un número concreto de paquetes mediante alguna opción de este comando (utilizar el **man** si no lo sabes). El resultado de un ping a una máquina que está conectada sería el que se ve en la siguiente imagen. Como puedes ver, en este caso se ha enviado un paquete y se ha recibido correctamente (1 received), mientras que si no se recibe nada, el mensaje sería diferente:

```
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.  
64 bytes from 10.0.2.15: icmp_req=1 ttl=64 time=1.32 ms  
  
--- 10.0.2.15 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 1.322/1.322/1.322/0.000 ms  
juanmi@sobremesa:~/practica_scripts/script1$
```

## GRUPO II

1. En el fichero **precipitaciones.txt** hay una serie de líneas que indican el día y la cantidad de litros que ha llovido. En este ejercicio deberás crear un script que muestre por pantalla la media de precipitaciones tal y como se ve en la siguiente ejecución:

```
juanmi@sobremesa:~/practica_scripts/examen$ ./script4.sh  
La media de precipitaciones es 6  
juanmi@sobremesa:~/practica_scripts/examen$
```

2. Escribe un script que muestre por pantalla el listado de días que no llovió, asumiendo que el primero es lunes.
3. Escribe un script que simule un comando de borrado de **ficheros.txt**. Para ello, el script tiene que recibir por parámetro un directorio y tiene que borrar todos los ficheros con extensión **.txt** que tenga ese directorio, mostrando al final, un mensaje con la cantidad de ficheros que se han borrado:

```
Se han borrado 10 ficheros  
juanmi@sobremesa:~/practica_scripts$
```

4. Partiendo del ejercicio anterior, antes de realizar el borrado, se deberá comprobar si existe el directorio antes de borrar, mostrando el mensaje **“El directorio no existe”** en caso que el directorio no exista. Hay diversas formas de saber si existe un directorio, una de ellas puede ser mediante el uso del comando **pwd**, que nos puede mostrar la ruta antes y después de intentar posicionarnos en el directorio.

### Evaluación de la actividad:

La actividad no es evaluable. Si lo fuera, se tendría en cuenta el grado de resolución de los ejercicios, la presencia o no de fallos y la calidad del código.

### Recursos adicionales:

Diapositivas y recursos de la Unidad 3 en Florida Oberta

Robbins, Beebe. Classic Shell Scripting. O'Reilly

Búsqueda autónoma en Internet