



1º DAM/DAW Sistemas Informáticos

U6 - Shell Scripts

AP8 - Shell Scripts

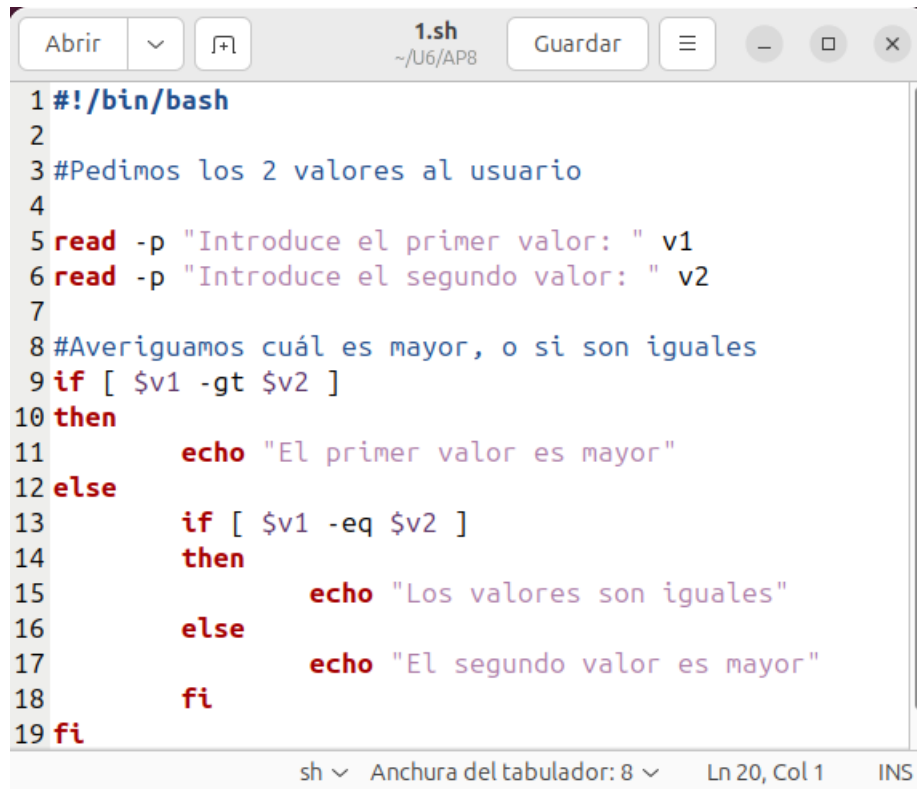


- **Título:** Shell Scripts.
- **Objetivos:**
 - Conocer los fundamentos sintácticos de Shell Scripts.
 - Utilizar las utilidades de entrada/salida en el desarrollo de Shell Scripts.
 - Utilizar estructuras de control en el desarrollo de Shell Scripts
 - Desarrollar Shell Scripts básicos para la automatización de tareas
- **Bibliografía:** puesta a disposición en el curso en Florida Oberta:
 - Documentos.
 - Anexos.
 - Recursos.
 - Foro.

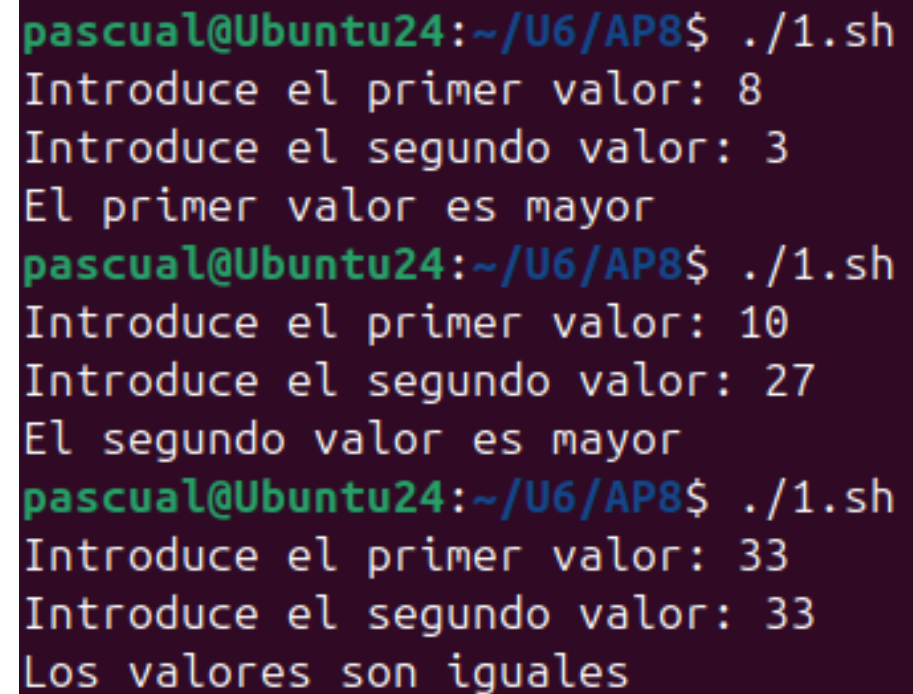
- **Entrega:**
 - Puedes entregar directamente los **ficheros “*.sh”** desarrollados **a través de Florida Oberta**.
 - **Opcional y adicionalmente**, puedes entregar un fichero pdf, debidamente identificado, con capturas de pantalla para mostrar el uso realizado. Así como razonamientos, explicaciones y reflexiones.
 - También puedes entregarlo todo en un fichero comprimido.
 - Recuerda que en este tipo de actividad práctica (AP), no es obligatorio realizar todas las cuestiones de la actividad de forma completa y correcta. El objetivo principal es ir adquiriendo destreza, por lo que se valorará la actividad como:
 - Entregada: aunque no esté 100% completa y correcta.
 - No entregada: o bien no se produce la entrega o bien se produce con un nivel tan bajo que denota que no se ha trabajado adecuadamente.

- Ejercicios:

1. Escribe un script que pida al usuario dos valores y muestre un mensaje indicando qué valor es el mayor de ambos, o bien que son iguales.



```
1 #!/bin/bash
2
3 #Pedimos los 2 valores al usuario
4
5 read -p "Introduce el primer valor: " v1
6 read -p "Introduce el segundo valor: " v2
7
8 #Averiguamos cuál es mayor, o si son iguales
9 if [ $v1 -gt $v2 ]
10 then
11     echo "El primer valor es mayor"
12 else
13     if [ $v1 -eq $v2 ]
14     then
15         echo "Los valores son iguales"
16     else
17         echo "El segundo valor es mayor"
18     fi
19 fi
```



```
pascual@Ubuntu24:~/U6/AP8$ ./1.sh
Introduce el primer valor: 8
Introduce el segundo valor: 3
El primer valor es mayor
pascual@Ubuntu24:~/U6/AP8$ ./1.sh
Introduce el primer valor: 10
Introduce el segundo valor: 27
El segundo valor es mayor
pascual@Ubuntu24:~/U6/AP8$ ./1.sh
Introduce el primer valor: 33
Introduce el segundo valor: 33
Los valores son iguales
```

- **Ejercicios:**

1.

```
#!/bin/bash
#Pedimos los 2 valores al usuario

read -p "Introduce el primer valor: " v1
read -p "Introduce el segundo valor: " v2

#Averiguamos cuál es mayor, o si son iguales
if [ $v1 -gt $v2 ]
then
    echo "El primer valor es mayor"
else
    if [ $v1 -eq $v2 ]
    then
        echo "Los valores son iguales"
    else
        echo "El segundo valor es mayor"
    fi
fi
```

- Ejercicios:

2. Escribe un script que pida al usuario un valor mayor que 0 y muestre un mensaje indicando si es par o impar. Si se introduce un valor menor o igual a 0, debe mostrar un mensaje y volver a pedirlo.

```
2.sh
~/U6/AP8
Abrir  Guardar  -  □  ×

1 #!/bin/bash
2
3 #Pedimos al usuario un valor mayor que 0
4 read -p "Introduce un valor mayor que 0: " v1
5
6 #Validamos el valor y si no es correcto lo pedimos de nuevo
7 while [ $v1 -le 0 ] ;do
8     read -p "Introduce un valor mayor que 0: " v1
9 done
10
11 #Calculamos el resto de dividir entre 2
12 resto=$((v1%2))
13
14 if [ $resto -eq 0 ]
15 then
16     echo "El valor $v1 es un número par"
17 else
18     echo "El valor $v1 es un número impar"
19 fi

sh  Anchura del tabulador: 8  Ln 1, Col 1  INS
```

```
pascual@Ubuntu24:~/U6/AP8$ ./2.sh
Introduce un valor mayor que 0: -8
Introduce un valor mayor que 0: -2
Introduce un valor mayor que 0: 0
Introduce un valor mayor que 0: 5
El valor 5 es un número impar
pascual@Ubuntu24:~/U6/AP8$ ./2.sh
Introduce un valor mayor que 0: 14
El valor 14 es un número par
```

- **Ejercicios:**

2.

```
#!/bin/bash
```

```
#Pedimos al usuario un valor mayor que 0
```

```
read -p "Introduce un valor mayor que 0: " v1
```

```
#Validamos el valor y si no es correcto lo pedimos de nuevo
```

```
while [ $v1 -le 0 ] ;do
```

```
    read -p "Introduce un valor mayor que 0: " v1
```

```
done
```

```
#Calculamos el resto de dividir entre 2
```

```
resto=$((v1%2))
```

```
if [ $resto -eq 0 ]
```

```
then
```

```
    echo "El valor $v1 es un número par"
```

```
else
```

```
    echo "El valor $v1 es un número impar"
```

```
fi
```

- Ejercicios:

3. Escribe un script que pida al usuario una nota, un valor entre 0 y 10, y muestre un mensaje con la calificación equivalente: insuficiente, suficiente, bien, notable o sobresaliente. Si se introduce un valor menor que 0 o mayor que 10, deberá mostrar un mensaje y volver a pedir la nota.

```
3.sh
~/U6/AP8
Abrir Guardar
1 #!/bin/bash
2
3 read -p "Introduce una nota entre 0 y 10: " nota
4
5 while [ $nota -lt 0 ] || [ $nota -gt 10 ] ;do
6     read -p "Introduce una nota entre 0 y 10: " nota
7 done
8
9 if [ $nota -lt 5 ]
10 then
11     echo "La calificación es insuficiente"
12 else
13     if [ $nota -lt 6 ]
14     then
15         echo "La calificación es suficiente"
16     else
17         if [ $nota -lt 7 ]
18         then
19             echo "La calificación es bien"
20         else
21             if [ $nota -lt 9 ]
22             then
23                 echo "La calificación es notable"
24             else
25                 echo "La calificación es sobresaliente"
26             fi
27         fi
28     fi
29 fi
```

```
pascual@Ubuntu24:~/U6/AP8$ ./3.sh
Introduce una nota entre 0 y 10: -1
Introduce una nota entre 0 y 10: 11
Introduce una nota entre 0 y 10: 0
La calificación es insuficiente
pascual@Ubuntu24:~/U6/AP8$ ./3.sh
Introduce una nota entre 0 y 10: 2
La calificación es insuficiente
pascual@Ubuntu24:~/U6/AP8$ ./3.sh
Introduce una nota entre 0 y 10: 5
La calificación es suficiente
pascual@Ubuntu24:~/U6/AP8$ ./3.sh
Introduce una nota entre 0 y 10: 6
La calificación es bien
pascual@Ubuntu24:~/U6/AP8$ ./3.sh
Introduce una nota entre 0 y 10: 8
La calificación es notable
pascual@Ubuntu24:~/U6/AP8$ ./3.sh
Introduce una nota entre 0 y 10: 10
La calificación es sobresaliente
```

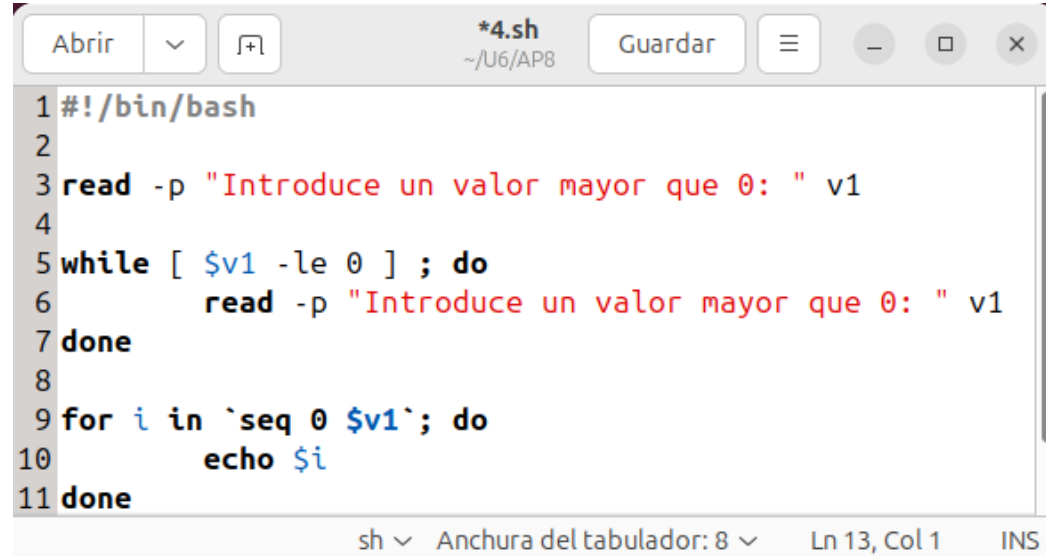

- **Ejercicios:**

3.

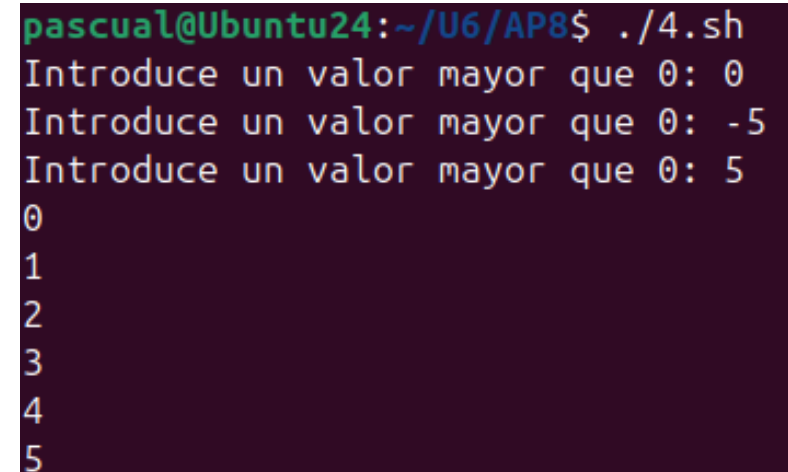
```
#!/bin/bash
read -p "Introduce una nota entre 0 y 10: " nota
while [ $nota -lt 0 ] || [ $nota -gt 10 ];do
    read -p "Introduce una nota entre 0 y 10: " nota
done
if [ $nota -lt 5 ]
then
    echo "La calificación es insuficiente"
else
    if [ $nota -lt 6 ]
    then
        echo "La calificación es suficiente"
    else
        if [ $nota -lt 7 ]
        then
            echo "La calificación es bien"
        else
            if [ $nota -lt 9 ]
            then
                echo "La calificación es notable"
            else
                echo "La calificación es sobresaliente"
            fi
        fi
    fi
fi
```

- **Ejercicios:**

4. Escribe un script que pida al usuario un valor mayor que 0 y muestre el listado de números que van desde 0 hasta este valor. Si se introduce un valor menor o igual que 0, debe mostrar un mensaje y volver a pedirlo.



```
1#!/bin/bash
2
3read -p "Introduce un valor mayor que 0: " v1
4
5while [ $v1 -le 0 ] ; do
6    read -p "Introduce un valor mayor que 0: " v1
7done
8
9for i in `seq 0 $v1`; do
10    echo $i
11done
```



```
pascual@Ubuntu24:~/U6/AP8$ ./4.sh
Introduce un valor mayor que 0: 0
Introduce un valor mayor que 0: -5
Introduce un valor mayor que 0: 5
0
1
2
3
4
5
```

- **Ejercicios:**

4.

```
#!/bin/bash
```

```
read -p "Introduce un valor mayor que 0: " v1
```

```
while [ $v1 -le 0 ] ; do
```

```
    read -p "Introduce un valor mayor que 0: " v1
```

```
done
```

```
for i in `seq 0 $v1`; do
```

```
    echo $i
```

```
done
```

- Ejercicios:

5. Escribe un script que pida al usuario valores, hasta que introduzca el 0. Cuando el usuario introduzca el 0, se mostrará un mensaje que muestre la suma total de valores introducidos, así como el valor medio.

```
1#!/bin/bash
2
3read -p "Introduce un valor (0 para terminar): " v1
4
5suma=0
6contador=0
7
8while [ $v1 -ne 0 ] ; do
9    suma=$((suma + v1))
10    contador=$((contador + 1))
11    read -p "Introduce un valor (0 para terminar): " v1
12done
13
14echo "La suma total es: $suma"
15
16#Para poder calcular la media con 2 decimales, si procede
17media=`echo "scale=2; $suma/$contador" | bc`
18echo "La media es: $media"
```

```
pascual@Ubuntu24:~/U6/AP8$ ./5.sh
Introduce un valor (0 para terminar): 4
Introduce un valor (0 para terminar): 5
Introduce un valor (0 para terminar): 8
Introduce un valor (0 para terminar): 2
Introduce un valor (0 para terminar): 7
Introduce un valor (0 para terminar): 6
Introduce un valor (0 para terminar): 0
La suma total es: 32
La media es: 5.33
```

- **Ejercicios:**

```
5.      #!/bin/bash
        read -p "Introduce un valor (0 para terminar): " v1

        suma=0
        contador=0

        while [ $v1 -ne 0 ] ; do
            suma=$((suma + v1))
            contador=$((contador + 1))
            read -p "Introduce un valor (0 para terminar): " v1
        done

        echo "La suma total es: $suma"

        #Para poder calcular la media con 2 decimales, si procede
        media=`echo "scale=2; $suma/$contador" | bc`
        echo "La media es: $media"
```

- **Ejercicios:**

6. Escribe un script que pida al usuario una cantidad de litros de agua consumidos y muestre el coste total, atendiendo a las siguientes indicaciones:

- Los primeros 50 litros cuestan 20 euros.
- Los restantes hasta 200 litros cuestan a 20 céntimos el litro.
- Los restantes cuestan a 10 céntimos el litro.

El script debe validar que el valor introducido sea mayor que 0 y si no fuera así, mostrará un mensaje y pedirá que se introduzca de nuevo el valor.

- Ejercicios:

6.

```
1#!/bin/bash
2
3read -p "Introduce una cantidad de litros consumidos: " litros
4
5while [ $litros -le 0 ]; do
6    read -p "Introduce una cantidad de litros consumidos: " litros
7done
8
9if [ $litros -le 50 ]
10then
11    coste=20
12else
13    if [ $litros -le 200 ]
14    then
15        resto=$((litros-50))
16        coste=`echo "scale=2; $resto*0.20+20" | bc`
17    else
18        resto=$((litros-200))
19        coste=`echo "scale=2; $resto*0.10+50" | bc`
20    fi
21fi
22
23echo "El coste total del consumo es: $coste"
```

```
pascual@Ubuntu24:~/U6/AP8$ ./6.sh
Introduce una cantidad de litros consumidos: -100
Introduce una cantidad de litros consumidos: 0
Introduce una cantidad de litros consumidos: 30
El coste total del consumo es: 20
pascual@Ubuntu24:~/U6/AP8$ ./6.sh
Introduce una cantidad de litros consumidos: 70
El coste total del consumo es: 24.00
pascual@Ubuntu24:~/U6/AP8$ ./6.sh
Introduce una cantidad de litros consumidos: 150
El coste total del consumo es: 40.00
pascual@Ubuntu24:~/U6/AP8$ ./6.sh
Introduce una cantidad de litros consumidos: 200
El coste total del consumo es: 50.00
pascual@Ubuntu24:~/U6/AP8$ ./6.sh
Introduce una cantidad de litros consumidos: 250
El coste total del consumo es: 55.00
pascual@Ubuntu24:~/U6/AP8$ ./6.sh
Introduce una cantidad de litros consumidos: 500
El coste total del consumo es: 80.00
```

- **Ejercicios:**

6.

```
#!/bin/bash
read -p "Introduce una cantidad de litros consumidos: " litros

while [ $litros -le 0 ]; do
    read -p "Introduce una cantidad de litros consumidos: " litros
done

if [ $litros -le 50 ]
then
    coste=20
else
    if [ $litros -le 200 ]
    then
        resto=$((litros-50))
        coste=`echo "scale=2; $resto*0.20+20" | bc`
    else
        resto=$((litros-200))
        coste=`echo "scale=2; $resto*0.10+50" | bc`
    fi
fi

echo "El coste total del consumo es: $coste"
```


- Ejercicios:

7. Escribe un script que pida al usuario un valor del día numérico del mes (de 1 a 30) y muestre por pantalla qué día de la semana es, asumiendo que el día 1 es lunes. Para cualquier valor introducido fuera del rango, se mostrará un mensaje y se volverá a pedir.

```
1#!/bin/bash
2
3read -p "Introduce un número de día entre 1 y 30: " dia
4
5while [ $dia -lt 1 ] || [ $dia -gt 30 ]; do
6    read -p "Introduce un número de día entre 1 y 30: " dia
7done
8
9#Si el resto es 0, el día es domingo
10#Si el resto es 1, el día es lunes
11#...
12resto=$((dia%7))
13
14contador=0
15
16for i in domingo lunes martes miercoles jueves viernes sabado; do
17
18    if [ $contador -eq $resto ]
19    then
20        echo "El día $dia es $i"
21    fi
22
23    contador=$((contador+1))
24
25done
```

```
pascual@Ubuntu24:~/U6/AP8$ ./7.sh
Introduce un número de día entre 1 y 30: 7
El día 7 es domingo
pascual@Ubuntu24:~/U6/AP8$ ./7.sh
Introduce un número de día entre 1 y 30: 1
El día 1 es lunes
pascual@Ubuntu24:~/U6/AP8$ ./7.sh
Introduce un número de día entre 1 y 30: 50
Introduce un número de día entre 1 y 30: -2
Introduce un número de día entre 1 y 30: 18
El día 18 es jueves
```

- **Ejercicios:**

```
7.      #!/bin/bash
        read -p "Introduce un número de día entre 1 y 30: " dia

        while [ $dia -lt 1 ] || [ $dia -gt 30 ]; do
            read -p "Introduce un número de día entre 1 y 30: " dia
        done

        #Si el resto es 0, el día es domingo
        #Si el resto es 1, el día es lunes
        #...
        resto=$((dia%7))
        contador=0

        for i in domingo lunes martes miercoles jueves viernes sabado; do
            if [ $contador -eq $resto ]
            then
                echo "El día $dia es $i"

            fi
            contador=$((contador+1))
        done
```