

SCRIPTS

El documento tiene que ser extensión `.sh`

La primera línea siempre será la siguiente para indicar el shell desde que lo ejecutamos, en este caso bash.

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

A continuación introducimos los comandos a ejecutar:

```
#!/bin/bash

echo "Hola mundo!!!"
```

Tenemos que tener **permisos de ejecución** para poder ejecutarlo: `chmod u+x <archivo.sh>` Si intentamos escribir el nombre directamente, lo busca en \$PATH. O guardamos el script en esa ruta, o indicamos la ruta con `./`

Para ejecutarlo: `./primer_script.sh`

TODO ESTO ESTÁ EN LA MÁQUINA pr0203, en el directorio vagrant

Hola \$USUARIO y \$FECHA actual:

Guardado con lo siguiente:

```
#!/bin/bash

echo "Hola $USER son las $(date +%H:%M:%S)"
```

Asignado permiso al usuario de ejecución y ejecutado como sigue `./script1.sh`

Almacenar comandos en variables

```
#!/bin/bash

ruta=`pwd`
echo $ruta
```

MANERA CORRECTA DE HACERLO:

```
#!/bin/bash

ruta=$(pwd)
echo $ruta
```

Script con Variables:

```
#!/bin/bash

usu=$USER
date=$(date +%H:%M:%S)
ruta=$(pwd)
echo Hola $usu, son las $date y estás en $ruta.
```

Se puede poner también echo Hola \$user SON LAS \$(date +%R')

Script con INPUT de Usuario

El -e es para que lea los escapados, en este caso \n para salto de línea

```
read -p "Di un número del 1 al 5: " -n1 num
echo -e "\nHas dicho el número $num"
```

Script para Contraseña

Hacer un Script

0. Que pregunte por un directorio y muestre el número de ficheros que tiene:
1. Pregunta a usuario por un directorio y muestra el nº de archivos y directorios que tiene.

```
#!/bin/bash
buscarf=$(find . -maxdepth 1 -type f | wc -l)
buscard=$(ls | wc -l)

read -p "Introduce la ruta del directorio para contar sus ficheros y directorios: " $ruta
echo -e "\nEn el directorio hay $buscarf$ruta ficheros"
echo -e "\nEn el directorio hay $buscard$ruta ficheros y directorios"
```

VICTOR:

```
read -p "Indica un directorio: " dir
num_items=$(ls $dir | wc -l)
echo "El directorio $dir tiene $num_items ficheros y directorios."
```

2. Pregunta al usuario por un nombre de usuario e indica qué intérprete de comandos utiliza (usar comando cut)

```
#!/bin/bash
read -p "Indica un nombre de usuario: " usu
echo "El shell de $usu se encuentra en $(grep $usu /etc/passwd | cut -d ':' -f 6)"
```

IFs, then y fi

if si no da errores pasa a ejecutar el **then**, si los hay tira directamente al **else**.

```
usuario=vagrant
if grep $usuario /etc/passwd > /dev/null
then
    echo "Los ficheros bash para $usuario son"
    ls -a /home/$usuario/.b*
fi
echo "Se acabó"
```

```
usuario=vagrant
if grep $usuario /etc/passwd > /dev/null
then
    echo "Los ficheros bash para $usuario son"
    ls -a /home/$usuario/.b*
else
    echo "El usuario no existe"
fi
```

Pide al usuario un nombre de intérprete (p.e /bin/bash, /bin/s e indica cuántos usuarios del sistema tienen ese intérprete. Si no hubiera ninguna muestra un mensaje diciendo que no los hay)

SHELLS disponibles: /bin/sh /bin/bash /usr/bin/bash /bin/rbash /usr/bin/rbash /usr/bin/sh /bin/dash /usr/bin/dash /usr/bin/tmux /usr/bin/screen

```
read -p "Indica un nombre de intérprete: " bash
$num=grep $bash /etc/passwd | wc -l
if grep $bash/etc/passwd > /dev/null
then
    echo "Hay $num usuarios utilizando este intérprete"
```

```
else
    echo "No existen usuarios que tengan este intérprete"
fi
```

Comparadores y tal

```
num=10

if [ $num -gt 8] //greater than
then
```

Pide al usuario su edad e indica:

Si es menor de 12 años pon eres un niño Entre 12 y 18 eres menor de edad Mayor de 18 eres mayor de edad

```
read -n "Indica tu edad: " eda
if [ $eda -lt 13]
then echo "Eres menor de edad"
```

Comprobaciones: #if test num -gt 10

if [num -gt 10] //Si num es mayor que 10, ejecuta el código: then echo "El num \$num es mayor que 10" fi

Para usar **comparación** dentro de un IF, hay que escapar el caracter if [\$user < "root"]

en lugar de if [\$user < "root"]

-z se cumple la condición si la variable con la que viene está vacía:

```
if [ -z $nombre]
then
    echo "No has introducido ningún nombre"
```

-n comprueba que NO está vacío

```
if [ -n $nombre ]
then
    echo "Hola $nombre"
```

Script automático de ssh

```
#!/bin/bash

if [ -d ~/.ssh ] //(-d) comprueba si existe y si es un directorio
then
    echo "Ya existe directorio .ssh"
else
    mkdir ~/.ssh
fi

if [ -f ~/.ssh/authorized_keys ] //(-f) comprueba si existe y es un fichero
then
    echo "Añadiendo clave al fichero de claves"
    echo public_key >> ~/.ssh/authorized_keys
else
    cp public_key ~/.ssh/authorized_keys
fi
```

Operador Y (&&) [booleanos]

```
if [ -d ~/.ssh ] && [ -w ~/.ssh ] && [ $USER = "root" ]
then
    echo "Ya existe el directorio y tienes permisos de escritura"
else
    echo "No existe o no puedes escribir en él"
fi
```

Operador OR (||)

```
if [ $USER="root" || $USER="victor" ]
then
    echo "Eres root o Victor"
fi
```

Operaciones

```
$(( 5+7 ))

read a
read b
suma=$(( $a+$b ))
```