

## Prácticas Shellscript

1. Realizar un programa llamado práctica 1 que solicite un número y si es mayor que 200, muestre el mensaje: “ El número introducido es mayor que 200”. En caso contrario que muestre el mensaje, “El número introducido es menor o igual que 200”

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

```
echo "Introduce un número"
read numero
```

```
if [ $numero -gt 200 ];
then
    echo "El número introducido es mayor que 200"
else
    echo "El número introducido es menor o igual que 200"
fi
```

2. Realizar un programa llamado práctica 2 que muestre en pantalla la siguiente pregunta “¿Quién descubrió América?” y según la respuesta muestre “Correcto” o “Incorrecto”.

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

```
echo "¿Quién descubrió América?"
read respuesta
```

```
if [ "$respuesta" = "Cristobal Colón"]
then
    echo "Correcto"
else
    echo "Incorrecto"
fi
```

3. Crear un programa llamado práctica3 que compruebe que al menos introducimos dos parámetros cuando llamamos al script. Si no introducimos al menos dos parámetros el programa no saldrá de la ejecución.

```
#!/bin/bash
while [ $# -lt 2 ]
do
    echo "No has introducido los parámetros correctos"
done
echo "Número de parámetros correcto"
```

4. Construir un programa llamado práctica 4 que solicite una ip y después de introducirla, ejecute un ping con 4 paquetes a ese ordenador.

```
#!/bin/bash
echo "Introduce IP"
read ip
ping $ip -c 4
```

5. Crear un script llamado práctica 5 que nos muestre un mensaje de bienvenida al usuario que está logeado en el sistema, del tipo , “Bienvenido al sistema `usuario`”.

```
#!/bin/bash  
echo “Bienvenido al sistema `whoami`”
```

6. Realizar un un script llamado práctica 6 que realice un backup del directorio de trabajo del usuario. Asignándole el nombre "backup-día de generación".

Ayuda: `backup_$(date +%Parametros)`.

```
#!/bin/bash  
tar -cvf backup_$(date +%d%m%Y).tar /home/usuario  
#Si queremos que comprima el archivo  
tar -czvf backup_$(date +%d%m%Y).tgz /home/usuario
```

7. Realizar un un script llamado práctica 7 que realice la Copia de seguridad de los archivos de un hosting. Primero nos hemos de conectar al servidor remoto y después copiar la carpeta

/var/www en una carpeta propia de nuestro servidor.

Ayuda: rsync.

```
#!/bin/bash  
echo “Introduzca dirección IP del servidor”  
read server  
echo “Introduzca usuario administrador del servidor”  
read usuario  
rsync -avz ${usuario}@$server:/var/www /home/web_servidor
```

8. Realizar un un script llamado práctica 8 que nos realice una migración de usuarios de un servidor Linux a otro nuevo.

Ayuda: los usuarios de un sistema Linux se encuentran en el archivo /etc/passwd, las contraseñas encriptadas en /etc/shadow y los grupos en /etc/group y en /home todas las carpetas de los usuarios.

```
#!/bin/bash  
rsync -avz user@ipremota:/etc/passwd /etc/passwd  
rsync -avz user@ipremota:/etc/group /etc/group  
rsync -avz user@ipremota:/etc/shadow /etc/shadow  
rsync -avz user@ipremota:/home /home
```

9. Realizar un un script llamado práctica 9 que nos muestre un pequeño menú de gestión de

un servidor Linux remoto. El usuario va a tener las siguientes opciones:

- A. Subir ficheros al servidor.
- B. Borrar ficheros del servidor.
- C. Realizar copia de seguridad de los archivos web.
- D. Reiniciar el servicio web.
- E. Reiniciar el servidor.

Ayuda: se recomienda el uso de funciones para un código mejor estructurado.

```

#!/bin/bash
ControlUsuario()
{
    echo "Introduzca dirección IP del servidor"
    read ip
    echo "Introduzca usuario administrador del servidor"
    read usuario
}

SubirFicheros()
{
    ControlUsuario

    echo "Introduzca directorio de los archivos a subir"
    read dirOrigen

    echo "Introduzca directorio donde se van a subir los archivos"
    read dirDestino

    if [ -d $dirOrigen ]
    then
        existe = `echo ssh -o StrictHostKeyChecking=no ${usuario}@$ip ls
$dire2 2      >> /dev/null`
        if [ -z $existe ]
        then
            scp $dirOrigen ${usuario}@${ip}:$dirDestino
        else
            echo "No existe el directorio en el servidor"
        fi
    else
        echo "No existe el directorio en el cliente"
    fi
}

BorrarFicheros(
{
    ControlUsuario

    echo "Introduzca archivo a borrar, con ruta completa"
    read archivo
    ssh ${usuario}@$ip del $archivo
}

CopiaWeb()
{
    ControlUsuario

    rsync -avz ${usuario}@$server:/var/www /home/web_servidor
}

```

```

RestartWeb()
{
    ControlUsuario

    ssh -t ${usuario}@$ip sudo service apache2 restart
}

RebootServer()
{
    ControlUsuario

    ssh -t ${usuario}@$ip sudo reboot
}

menu()
{
    echo introduzca la opción deseada
    echo
    echo A.-Subir ficheros al servidor
    echo B.-Borrar ficheros del servidor.
    echo C.-Realizar copia de seguridad de los archivos web.
    echo D.-Reiniciar el servicio web.
    echo E.-Reiniciar el servidor
    echo F .Salir.
    echo
}

opcion="G"
while [ $opcion != "F" ]
do
    menu
    read opcion
    case $opcion in
        "A") SubirFicheros;;
        "B") BorrarFicheros;;
        "C") CopiaWeb;;
        "D") RestartWeb;;
        "E") RebootServer;;
        "F")
            echo "Saliendo...."
            exit
            break;;
        *)
            echo "Introduce Opción Válida";;
    esac
done

```

```
        esac
done
```

10. Realizar un un script llamado práctica 10 que realice una comprobación de conectividad de todos los equipos que tengamos en nuestra red.

```
#!/bin/bash
echo "Vamos a comprobar la conectividad para la red 192.68.0.0"
nmap -sP $192.168.0.* | tr -s ' ' ':' > conect
```

```
c=0
```

```
for a in `cat conect`
do
    if [ `echo $a | cut -d ':' -f1` = 'Nmap' ]
    then
        c=`expr $c + 1`
        r=`echo $a | cut -d ':' -f5`

        if [ $r != 'addresses' ]
        then
            echo $r
        else
            c=`expr $c - 1`
            echo hay $c equipos conectados
        fi
    fi
done
```