

1

<b>Comenzado el</b>	miércoles, 19 de noviembre de 2025, 18:58
<b>Estado</b>	Finalizado
<b>Finalizado en</b>	miércoles, 19 de noviembre de 2025, 19:47
<b>Tiempo empleado</b>	48 minutos 49 segundos
<b>Calificación</b>	15,42 de 20,00 (77,08%)

**Pregunta 1**

Correcta

Se puntúa 1,00 sobre 1,00

¿Que hace el comando test -w /home/pepe/ ?

- ☒ a. Evalua si el archivo o directorio existe y se tiene el permiso de escritura ✓
- ☐ b. Evalua si el archivo o directorio existe y es un archivo regular
- ☐ c. Evalua si el archivo o directorio existe y es un directorio
- ☐ d. Evalua si el archivo o directorio existe
- ☐ e. Evalua si el archivo o directorio existe y se tiene el permiso de lectura

**Pregunta 2**

Incorrecta

Se puntúa 0,00 sobre 1,00

¿Qué imprime la siguiente secuencia de comandos? (Responder solamente el resultado de la secuencia de comandos, sin explicaciones).

```
echo 1zTrao dpaoqda | tr Taqz1rd irbuQea
```

Respuesta: 

**Pregunta 3**

Sin contestar

Se puntúa como 0 sobre 1,00

Dado el siguiente script, ¿Que afirmaciones son verdaderas acerca de su ejecucion?

```
#!/bin/bash
for i in {1..100}; do
  while true; do
    if ! (($i % 25)); then
      echo "$i es divisible por 25"
      continue 2
    elif [ $i -eq 53 ]; then
      break 2
    elif [ "$i % $i" ]; then
      break
    fi
  done
done
```

- ☐ a. Imprime que los valores 25 y 50 son divisibles por 25
- ☐ b. Es un bucle infinito
- ☐ c. Imprime todos los valores del 1 al 100 divisibles por 25
- ☐ d. Cuando el valor de i llega al 53, el script termina.

**Pregunta 4**

Parcialmente correcta

Se puntúa 0,25 sobre 1,00

¿Cuales de las siguientes equivalencias son ciertas?

- ☐ a. 

find -name archivo  
# equivale a:  
find . -name archivo
- ☒ b. 

cat archivo | wc -c  
# equivale a:  
wc -c archivo | cut -d' ' -f1

 ✓
- ☐ c. 

find -name archivo  
# equivale a:  
echo archivo | find -name
- ☐ d. 

echo hola | cat > salida  
# equivale a:  
cat hola > salida
- ☐ e. 

grep ac archivo  
# equivale a:  
cat archivo | grep ac
- ☐ f. 

echo hola | cat > salida  
# equivale a:  
echo hola > salida

**Pregunta 5**

Correcta

Se puntúa 1,00 sobre 1,00

¿Que resultado tiene el siguiente comando "cat /etc/passwd | cut -f1 -d: | grep "^a" ?

- ☐ a. Imprime los homes de los usuarios que tienen una letra a
- ☐ b. Imprime los nombre de usuario que contenga una letra a
- ☐ c. Imprime las password de los usuarios que contenga una letra a
- ☒ d. Imprime los nombres de los usuarios que empiecen con la letra a ✓
- ☐ e. No imprime nada

**Pregunta 6**

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuales de los siguientes usos de variables son correctos?

- ☒ a. echo \$NOMBRE ✓
- ☐ b. echo APELLIDO
- ☒ c. DIRECCION="56 nro 436" ✓
- ☒ d. echo \${DIRECCION} ✓
- ☐ e. APELLIDO = "sanchez"
- ☐ f. var NOMBRE="pepe"

**Pregunta 7**

Incorrecta

Se puntúa 0,00 sobre 1,00

Qué opciones son verdaderas respecto a la secuencia de comandos:

```
(test -f archivo && grep menta archivo && echo Z) || echo Q
```

- ☐ a. Siempre imprime "Z"
- ☐ b. Sin los paréntesis el resultado sería el mismo
- ☐ c. Si "archivo" existe y contiene el string "menta" imprime "Q", sino imprime "Z"
- ☐ d. Siempre imprime "Q"
- ☒ e. Sin los paréntesis el resultado sería distinto ✗
- ☒ f. Si "archivo" existe y contiene el string "menta" imprime "Z", sino imprime "Q" ✓

**Pregunta 8**

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuales de los siguientes usos de arreglos son correctos?

- ☐ a. echo \$arreglo[2]
- ☒ b. echo \${#arreglo[\*]} ✓
- ☒ c. echo \${arreglo[1]} ✓
- ☒ d. agregar un nuevo elemento: arreglo+=(5) ✓
- ☐ e. arreglo = (3 4 5)
- ☒ f. arreglo=(3 4 5) ✓
- ☐ g. agregar un nuevo elemento: arreglo+=5

**Pregunta 9**

Correcta

Se puntúa 1,00 sobre 1,00

¿Que información tiene la variable \$#?

- ☐ a. La cantidad de arreglos utilizados en el script
- ☒ b. La cantidad de parametros que se enviaron ✓
- ☐ c. Ninguna es solo un comentario
- ☐ d. La cantidad de variables utilizadas en el script

**Pregunta 10**

Parcialmente correcta

Se puntúa 0,50 sobre 1,00

Se desea hacer un script que imprima su primer argumento en pantalla, por ejemplo:

```
$ ./mi_script.sh "Hola Mundo"
```

debe imprimir "Hola Mundo" (sin las comillas).

¿Cuál de las siguientes es una implementación que cumple este objetivo?

- ☐ a. 

```
#!/bin/sh
for i; do
echo $i
break
done
```
- ☐ b. 

```
#!/bin/sh
echo $0
```
- ☐ c. 

```
#!/bin/sh
echo $argv[1]
```
- ☐ d. 

```
#!/bin/sh
echo argv[0]
```
- ☐ e. 

```
#!/bin/sh
echo ${argv[1]}
```
- ☐ f. 

```
#!/bin/sh
echo argv[1]
```
- ☐ g. 

```
#!/bin/sh
echo ${argv[0]}
```
- ☒ h. 

```
#!/bin/sh
echo $1
```

 ✓

**Pregunta 11**

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuáles de las siguientes son funciones correctamente definidas?

- ☐ a. 

```
x(1, 2){  
    echo $1 | grep $2  
}
```
- ☐ b. 

```
function x(1, 2); do  
    echo $1 | grep $2  
done
```
- ☐ c. 

```
function x(1, 2){  
    echo $1 | grep $2  
}
```
- ☒ d. 

```
x(){  
    echo $1 | grep $2  
}
```

 ✓
- ☒ e. 

```
function x{  
    echo $1 | grep $2  
}
```

 ✓

**Pregunta 12**

Correcta

Se puntúa 1,00 sobre 1,00

Asocie los siguientes operadores de bash con sus significados o comandos equivalentes.

- echo cadena |  
grep archivo ✓ Busca el string "archivo" dentro del string "cadena"
- echo cadena >  
archivo ✓ Sobreescribe "archivo" con el string "cadena"
- \$(./archivo) ✓ Equivale a: `./archivo`
- echo cadena >>  
archivo ✓ Escribe "cadena" al final de "archivo"
- [ expresión ] ✓ Equivale a: test expresión
- echo \$(( expresión )) ✓ Equivale a: expr expresión
- cadena=`./archivo` ✓ Guarda lo que imprime "archivo" al ejecutarse adentro de la variable "cadena"

**Pregunta 13**

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuales de las siguientes sintaxis del comando if son correctas?

- ☐ a. `if test -d /home/pepe then echo "es un directorio" fi`
- ☒ b. `if [ "$nombre" == "Maria" ]  
then  
echo "Es igual"  
fi` ✓
- ☒ c. `if test -r /home/pepe; then echo "Tengo permisos de lectura"; fi` ✓
- ☐ d. `if ["$nombre" == "Maria"]  
then echo "Es igual"  
fi`

**Pregunta 14**

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuales de los siguientes usos de expr es correcto?

- ☒ a. `expr 5 != 5` ✓
- ☐ b. `expr - 6 5`
- ☒ c. `expr 4 + 5` ✓
- ☐ d. `expr 4 5 *`
- ☒ e. `expr length "PEPE"` ✓

**Pregunta 15**

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuales de las siguientes sintaxis del comando for son correctas?

- ☒ a. ✓  
for i in \$(seq 1 100); do echo "\$i"; done
- ☐ b.  
for i in `seq 1 100` do  
echo "\$i"  
done
- ☒ c. ✓  
for i in `seq 1 100`  
do  
echo "\$i"  
done
- ☐ d.  
for \$i in `seq 1 100`  
do  
echo "\$i"  
done
- ☒ e. ✓  
for i in `seq 1 100`; do echo "\$i"; done

**Pregunta 16**

Correcta

Se puntúa 1,00 sobre 1,00

¿Que hace el comando "find / -name pepe 2> /dev/null" ?

- ☒ a. muestra los archivos llamados pepe, siempre y cuando tenga permisos de acceso a ellos ✓
- ☐ b. crea un archivo vacio en /dev/null
- ☐ c. envia la salida del comando al /dev/null

**Pregunta 17**

Correcta

Se puntúa 1,00 sobre 1,00

¿Con que simbolo se declara un comentario?

- ☒ a. # ✓
- ☐ b. %
- ☐ c. //
- ☐ d. ?
- ☐ e. ;



**Pregunta 18**

Parcialmente correcta

Se puntúa 0,67 sobre 1,00

Se desea hacer un script que imprima la lista de argumentos que recibe. ¿Cuales de las siguientes implementaciones son correctas?.

- ☐ a. 

```
#!/bin/sh
for arg in argv; do
    echo $arg
done
```
- ☐ b. 

```
#!/bin/sh
echo $#
```
- ☐ c. 

```
#!/bin/sh
for arg in ${argv[*]}; do
    echo $arg
done
```
- ☐ d. 

```
#!/bin/sh
echo $*
```
- ☒ e. 

```
#!/bin/sh
for arg in $@; do
    echo $arg
done
```

 ✓
- ☒ f. 

```
#!/bin/sh
for arg in $*; do
    echo $arg
done
```

 ✓

**Pregunta 19**

Correcta

Se puntúa 1,00 sobre 1,00

¿Que comillas se utilizan para la hacer uso de la sustitucion de comandos?

- ☐ a. " "
- ☐ b. ' '
- ☒ c. ` ` ✓
- ☐ d. ^ ^

**Pregunta 20**

Correcta

Se puntúa 1,00 sobre 1,00

Supongamos que tenemos un archivo llamado "archivo" con el siguiente contenido:

```
eth2 Link encap:Ethernet HWaddr 00:e0:7d:b4:1c:38
inet addr:192.168.1.112 Bcast:255.255.255.255
Mask:255.255.255.0
inet6 addr: fe80::2e0:7dff:feb4:1c38/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:115588267 errors:0 dropped:0 overruns:0 frame:0
TX packets:35601221 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:3240134640 (3.2 GB) TX bytes:2042178372 (2.0 GB)
Interrupt:20 Base address:0xce00
```

que resultado se obtiene al ejecutar:

```
cat archivo | grep inet | cut -d: -f2 | cut -d" " -f1
```

- ☐ a. 00:e0:7d:b4:1c:38
- ☒ b. 192.168.1.112 ✓
- ☐ c. 255.255.255.255
- ☐ d. Ningun resultado
- ☐ e. fe80::2e0:7dff:feb4:1c38/64