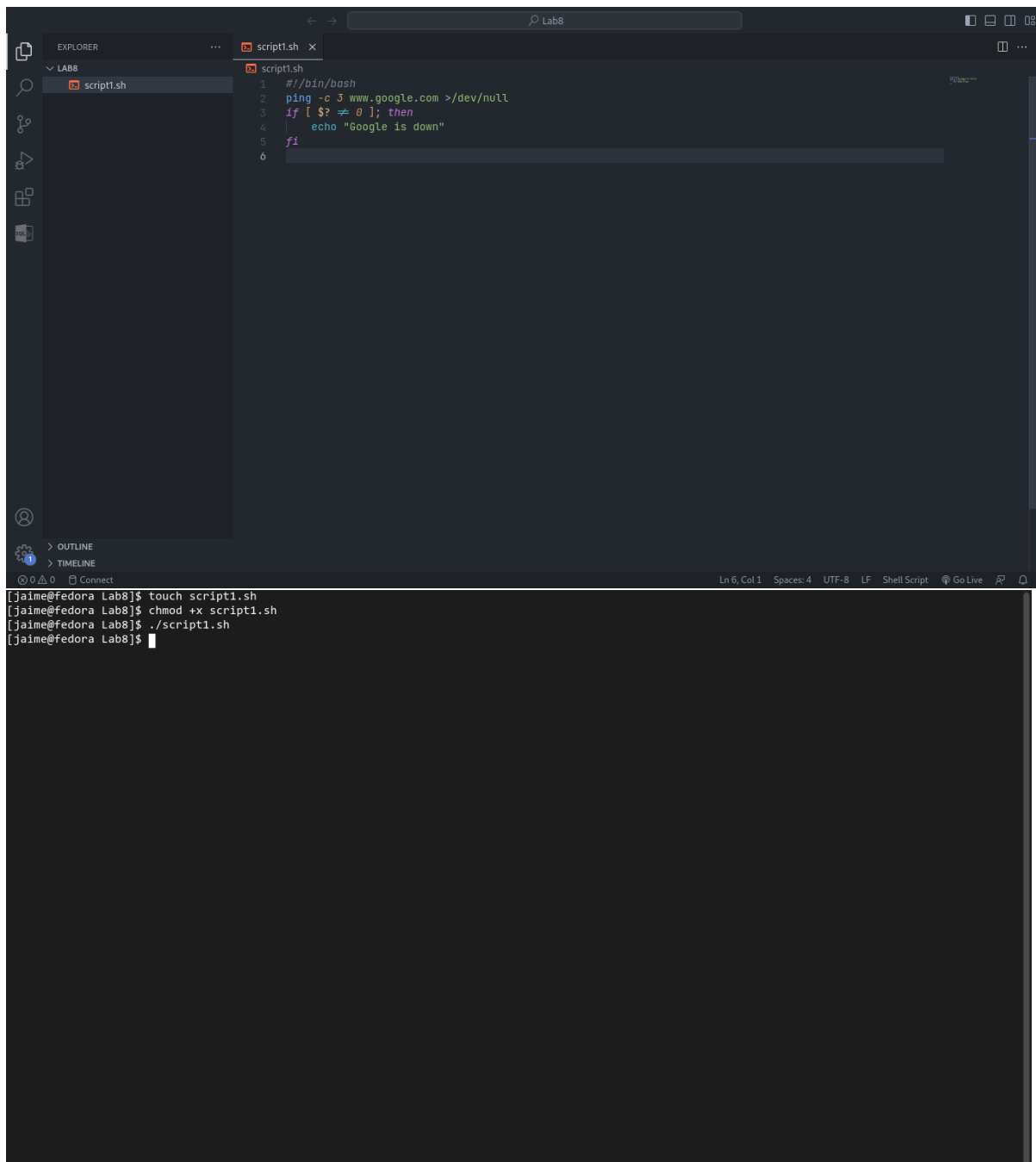


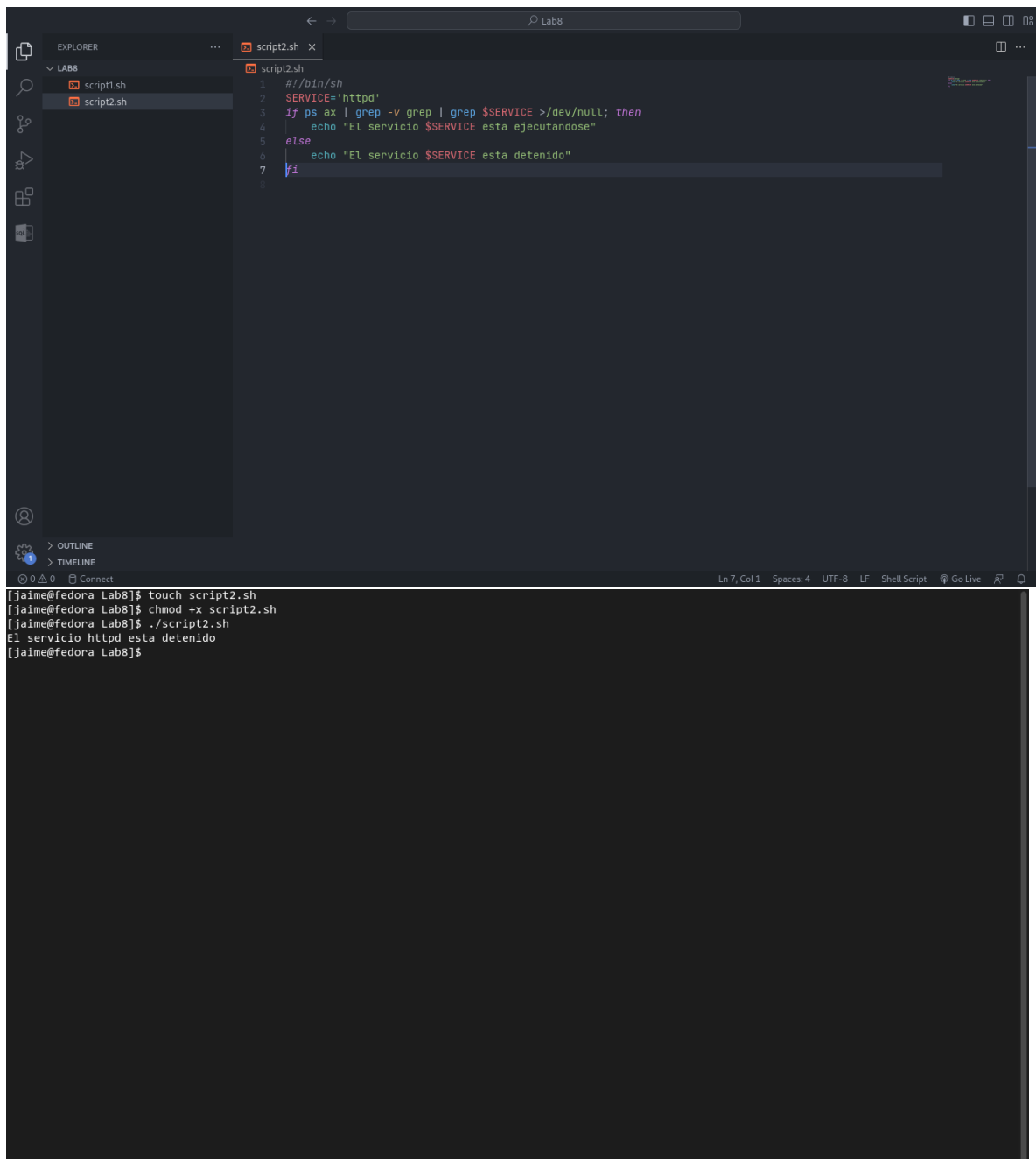
Procedimiento 1



```
#!/bin/bash
1
2 ping -c 3 www.google.com >/dev/null
3 if [ $? != 0 ]; then
4     echo "Google is down"
5 fi
6
```

```
[jaime@fedora Lab8]$ touch script1.sh
[jaime@fedora Lab8]$ chmod +x script1.sh
[jaime@fedora Lab8]$ ./script1.sh
[jaime@fedora Lab8]$
```

Procedimiento 2

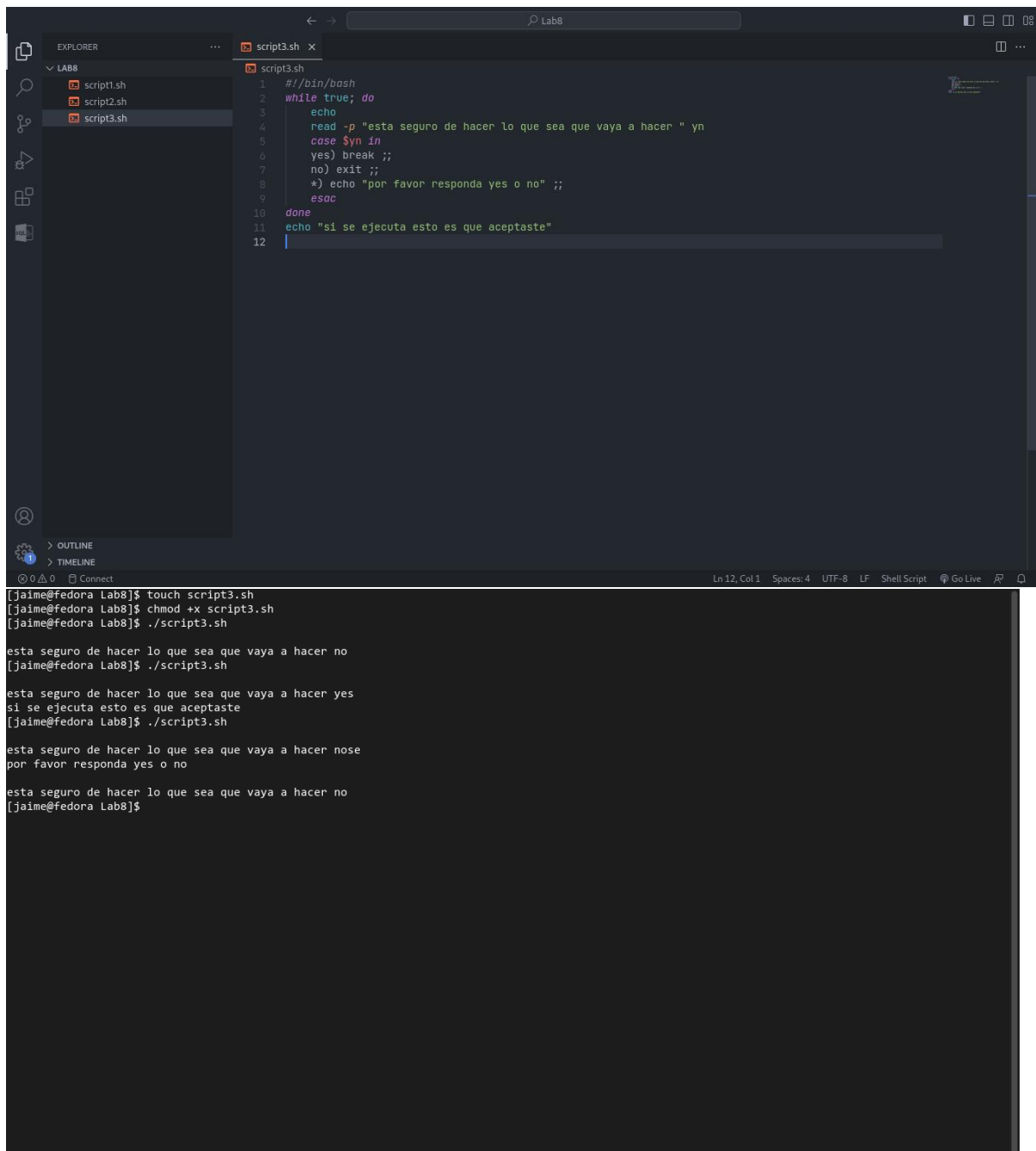


The image shows a Visual Studio Code editor window with a file explorer on the left and a terminal at the bottom. The file explorer shows a directory named 'Lab8' containing two files: 'script1.sh' and 'script2.sh'. The 'script2.sh' file is open in the editor, showing a shell script that checks if the 'httpd' service is running. The terminal at the bottom shows the execution of the script, which outputs 'El servicio httpd esta detenido'.

```
1 #!/bin/sh
2 SERVICE='httpd'
3 if ps ax | grep -v grep | grep $SERVICE >/dev/null; then
4     echo "El servicio $SERVICE esta ejecutandose"
5 else
6     echo "El servicio $SERVICE esta detenido"
7 fi
```

```
[jaime@fedora Lab8]$ touch script2.sh
[jaime@fedora Lab8]$ chmod +x script2.sh
[jaime@fedora Lab8]$ ./script2.sh
El servicio httpd esta detenido
[jaime@fedora Lab8]$
```

Procedimiento 3



The image shows a Visual Studio Code editor window with a file explorer on the left and a terminal at the bottom. The file explorer shows a directory named 'LAB8' containing three files: 'script1.sh', 'script2.sh', and 'script3.sh'. The 'script3.sh' file is selected and its contents are displayed in the editor. The script is a shell script that uses a 'while' loop to repeatedly prompt the user for input. The prompt is 'esta seguro de hacer lo que sea que vaya a hacer ' yn'. The user can respond with 'yes' or 'no'. If 'yes' is entered, the script breaks the loop. If 'no' is entered, the script exits. If any other input is entered, the script echoes 'por favor responda yes o no' and continues the loop. After the loop, the script echoes 'si se ejecuta esto es que aceptaste'.

```
1 #!/bin/bash
2 while true; do
3     echo
4     read -p "esta seguro de hacer lo que sea que vaya a hacer " yn
5     case $yn in
6         yes) break ;;
7         no) exit ;;
8         *) echo "por favor responda yes o no" ;;
9     esac
10 done
11 echo "si se ejecuta esto es que aceptaste"
12
```

The terminal at the bottom shows the execution of the script. The user runs 'touch script3.sh', 'chmod +x script3.sh', and './script3.sh'. The output of the script is shown below each command:

```
[jaime@fedora Lab8]$ touch script3.sh
[jaime@fedora Lab8]$ chmod +x script3.sh
[jaime@fedora Lab8]$ ./script3.sh

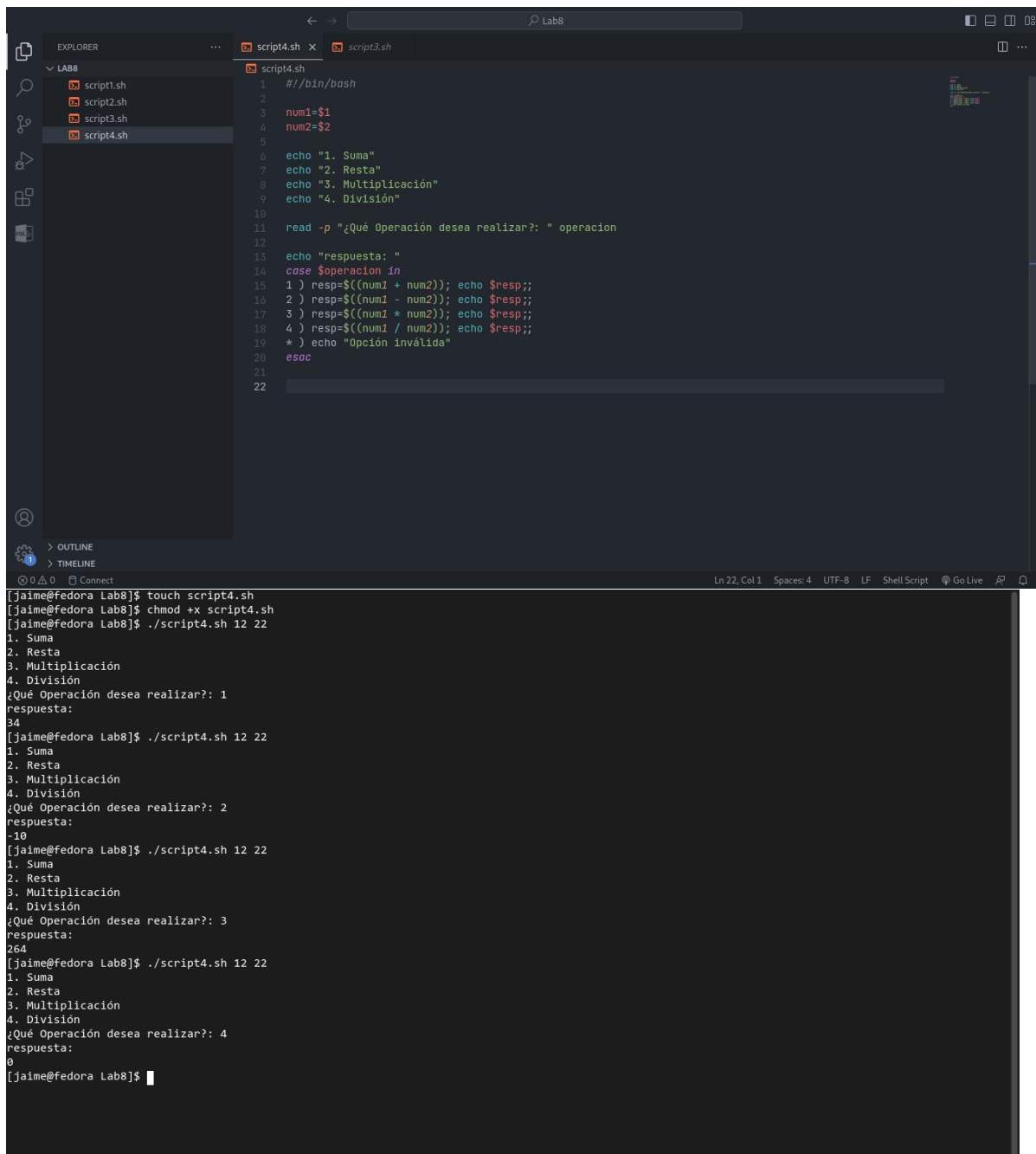
esta seguro de hacer lo que sea que vaya a hacer no
[jaime@fedora Lab8]$ ./script3.sh

esta seguro de hacer lo que sea que vaya a hacer yes
si se ejecuta esto es que aceptaste
[jaime@fedora Lab8]$ ./script3.sh

esta seguro de hacer lo que sea que vaya a hacer nose
por favor responda yes o no
[jaime@fedora Lab8]$ ./script3.sh

esta seguro de hacer lo que sea que vaya a hacer no
[jaime@fedora Lab8]$
```

Procedimiento 4



The image shows a VS Code editor window with a file explorer on the left and a terminal at the bottom. The file explorer shows a directory named 'LAB8' containing four files: 'script1.sh', 'script2.sh', 'script3.sh', and 'script4.sh'. The editor is open to 'script4.sh', which contains the following code:

```
1  #!/bin/bash
2
3  num1=$1
4  num2=$2
5
6  echo "1. Suma"
7  echo "2. Resta"
8  echo "3. Multiplicación"
9  echo "4. División"
10
11 read -p "¿Qué Operación desea realizar?: " operacion
12
13 echo "respuesta: "
14 case $operacion in
15 1) resp=$((num1 + num2)); echo $resp;;
16 2) resp=$((num1 - num2)); echo $resp;;
17 3) resp=$((num1 * num2)); echo $resp;;
18 4) resp=$((num1 / num2)); echo $resp;;
19 *) echo "Opción inválida"
20 esac
21
22
```

The terminal at the bottom shows the execution of the script. The user runs 'touch script4.sh', 'chmod +x script4.sh', and then './script4.sh 12 22'. The script prompts for an operation, and the user enters '1'. The script outputs '1. Suma' and '34'. The user then enters '2', and the script outputs '2. Resta' and '-10'. The user then enters '3', and the script outputs '3. Multiplicación' and '264'. Finally, the user enters '4', and the script outputs '4. División' and '0'.

```
[jaime@fedora Lab8]$ touch script4.sh
[jaime@fedora Lab8]$ chmod +x script4.sh
[jaime@fedora Lab8]$ ./script4.sh 12 22
1. Suma
2. Resta
3. Multiplicación
4. División
¿Qué Operación desea realizar?: 1
respuesta:
34
[jaime@fedora Lab8]$ ./script4.sh 12 22
1. Suma
2. Resta
3. Multiplicación
4. División
¿Qué Operación desea realizar?: 2
respuesta:
-10
[jaime@fedora Lab8]$ ./script4.sh 12 22
1. Suma
2. Resta
3. Multiplicación
4. División
¿Qué Operación desea realizar?: 3
respuesta:
264
[jaime@fedora Lab8]$ ./script4.sh 12 22
1. Suma
2. Resta
3. Multiplicación
4. División
¿Qué Operación desea realizar?: 4
respuesta:
0
[jaime@fedora Lab8]$
```

Conclusión

En este laboratorio se pudo aprender a usar la funcionalidad de automatización en Linux por medio de los scripts de bash.