

Capítulo 1 – Introducción al ambiente de red: Práctica de creación y uso de Scripts en el shell de Linux

Christian Amauri

Scripting en el shell de linux

72370 puebla, México

Christian Amauri, christian.amadoro@alumno.buap.mx

En este reporte documentaremos una práctica llevada a cabo en la Facultad de ciencias de la computación de la Benemérita Universidad Autónoma de Puebla, para la materia de administración de redes. En la cual debemos crear comandos básicos para el shell de Linux. En la introducción planteamos un poco el propósito de la práctica y sus objetivos/limitaciones, en el Desarrollo explicamos la metodología de trabajo y se presentan 3 ejercicios, en las conclusiones reflexionamos un poco acerca de los resultados y finalmente tenemos la sección de referencias, en donde se encuentran 3 links como material de apoyo.

1 Introducción

El shell de Linux es una interfaz de línea de comandos que permite a los usuarios interactuar con el sistema operativo de manera más eficiente y poderosa que a través de una interfaz gráfica de usuario. Una de las principales ventajas del uso del shell de Linux es la capacidad de crear y ejecutar comandos personalizados para automatizar tareas y realizar operaciones específicas en el Sistema.

En esta práctica comenzamos con los comandos más básicos que Podemos implementar en un archivo con extensión .sh, se proponen 3 ejercicios que implementan básicamente; entrada/salida de datos, y un poco de sintaxis. Además de hablar un poco sobre cómo debemos crear, guardar e interactuar con dichos archivos.

2 Desarrollo

Para crear scripts en el Shell de linux, lo primero que debemos hacer es ir al cmd y escribir cd, seguido de la ruta en la que queremos guardar/consultar nuestros archivos. En nuestro caso: cd Escritorio. Creamos entonces un script con extension .sh usando el commando “nano”:

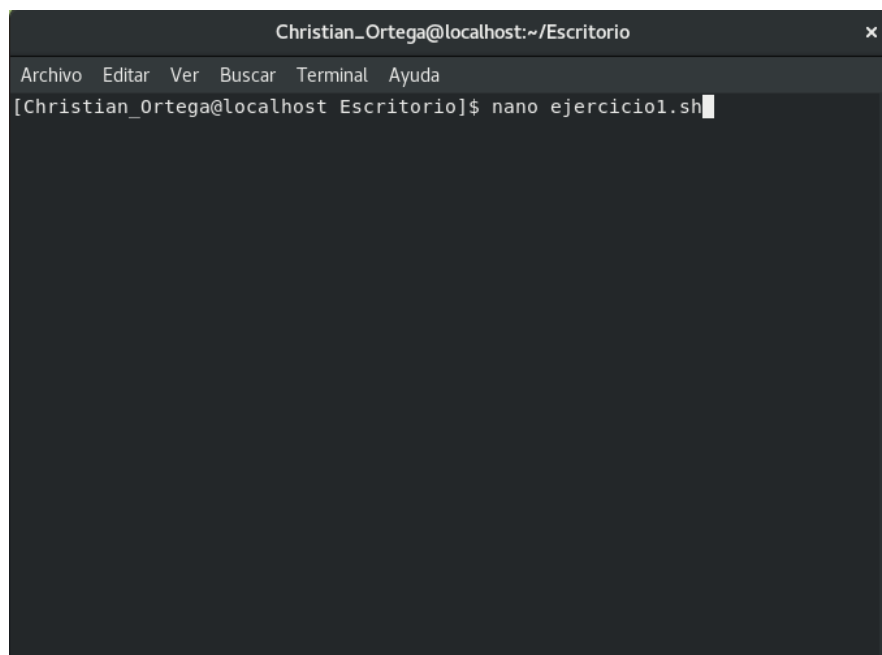
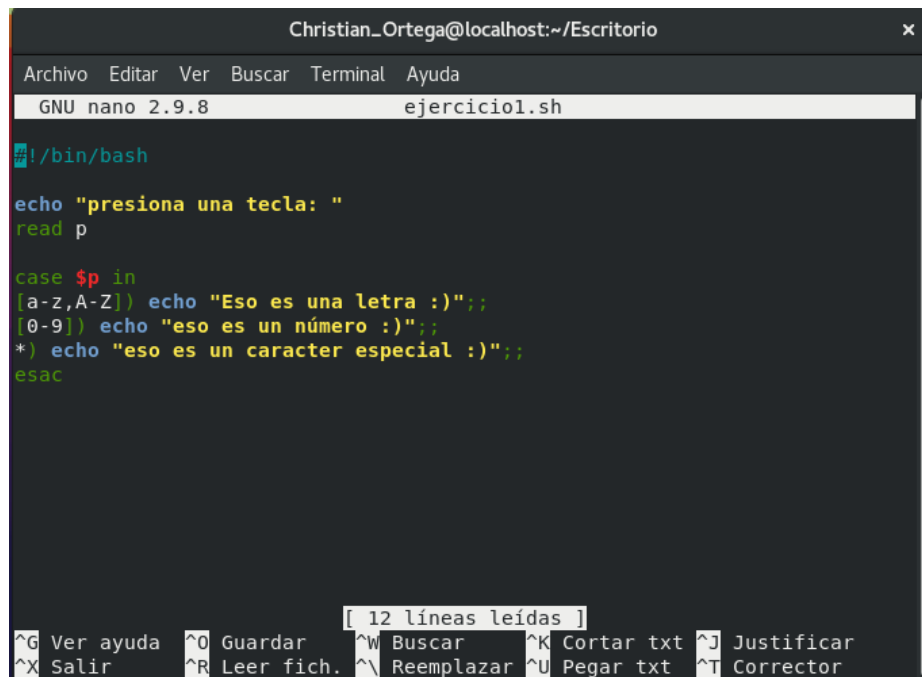


Fig. 1. uso del comando “nano” para crear un archivo con extensión .sh.

Una vez creado el archivo se nos abrirá la interfaz de GNU NANO, procedemos a escribir el script. para crear scripts tenemos diversas palabras reservadas y una sintaxis que debemos respetar. (En la sección de referencias de este reporte hay un link de descarga de un archivo pdf que habla sobre la creación de scripts en el shell de Linux). Además de aprender la sintaxis, hay que aprender las opciones de archivo que se nos presenta en GNU NANO. En nuestro caso sólo usaremos guardar (ctrl o) y salir (ctrl x).

La primera línea que debemos escribir para indicar que el archivo que estamos editando es un programa (También conocida como “Sha Bang”), es: `#!/bin/bash` Luego de eso viene la creación/edición del script:



```
#!/bin/bash

echo "presiona una tecla: "
read p

case $p in
[a-zA-Z]) echo "Eso es una letra :)" ;;
[0-9]) echo "eso es un número :)" ;;
*) echo "eso es un caracter especial :)" ;;
esac
```

Fig. 2. Scripting de un archivo con extensión .sh en la interfaz de GNU NANO.

Finalmente, una vez creado y guardado el script debemos darle los permisos necesarios para que pueda ser ejecutado adecuadamente, para ello escribimos el nombre del archivo, precedido por el comando “chmod 777” como se muestra a continuación:

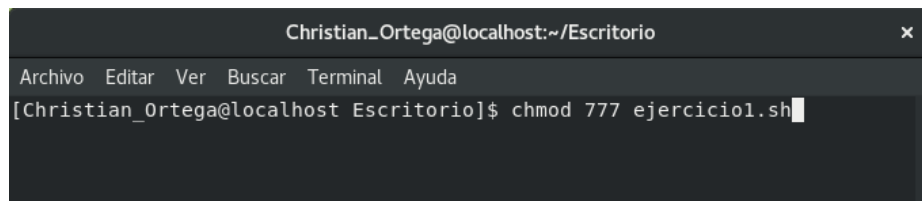
A screenshot of a terminal window titled "Christian_Ortega@localhost:~/Escritorio". The window has a menu bar with "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The command prompt shows "[Christian_Ortega@localhost Escritorio]\$ chmod 777 ejercicio1.sh" with a cursor at the end of the command.

Fig. 3. Otorgar permisos al archivo con extensión .sh que acabamos de crear mediante el comando “chmod 777” y el nombre del archivo.

Para ejecutar el archivo que acabamos de crear, escribimos la línea: ./nombre-del-archivo.sh Y el programa se ejecutará. Siguiendo esta metodología es como creamos programas en el shell de Linux, de esta manera, presentamos los siguientes 3 ejercicios implementados en la práctica:

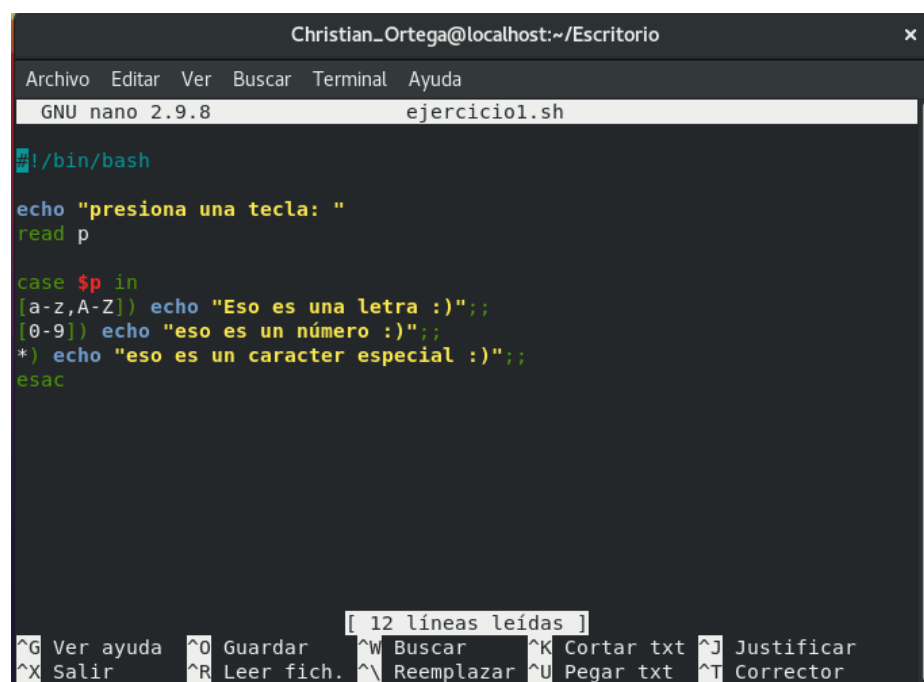
2.1 Ejercicio 1

Se plantea el siguiente ejercicio:

Crea un script que compruebe que tecla has pulsado:

- 1. pide por teclado la tecla que será guardada en la variable tecla
- 2. Si la tecla es una letra se hará todos los comandos que hayan en el caso [a-z,A-Z
- 3. si la tecla es un número se hará todos los comandos que haya en el caso [0-9
- 4. si la tecla no es un número ni una letra, se ejecutan todos los comandos del caso *.

La solución de este problema está (además de la salida de datos (mensajes) y entrada de datos (valor de p)), en el uso de una estructura de control tipo switch, cuya sintáxis consiste en las palabras “case - in”, y un paréntesis derecho por cada opción deseada. Finalizando por la palabra esac:



```
Christian_Ortega@localhost:~/Escritorio
GNU nano 2.9.8 ejercicio1.sh

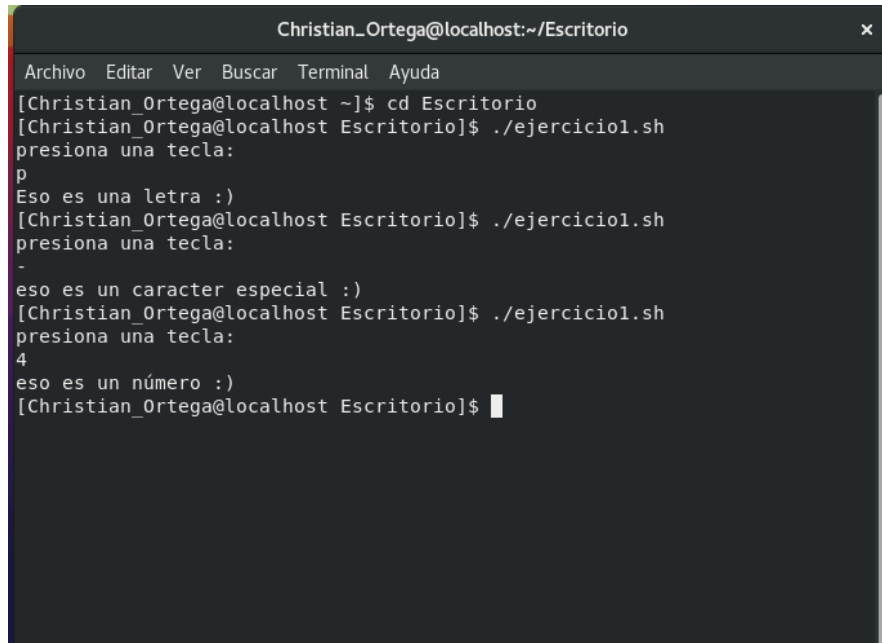
#!/bin/bash

echo "presiona una tecla: "
read p

case $p in
[a-zA-Z]) echo "Eso es una letra :)" ;;
[0-9]) echo "eso es un número :)" ;;
*) echo "eso es un caracter especial :)" ;;
esac

[ 12 líneas leídas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar txt ^J Justificar
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Corrector
```

Fig. 4. Script creado para cumplir el propósito del ejercicio 1.



```
Christian_Ortega@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[Christian_Ortega@localhost ~]$ cd Escritorio
[Christian_Ortega@localhost Escritorio]$ ./ejercicio1.sh
presiona una tecla:
p
Eso es una letra :)
[Christian_Ortega@localhost Escritorio]$ ./ejercicio1.sh
presiona una tecla:
-
eso es un caracter especial :)
[Christian_Ortega@localhost Escritorio]$ ./ejercicio1.sh
presiona una tecla:
4
eso es un número :)
[Christian_Ortega@localhost Escritorio]$
```

Fig. 5. Resultado de la ejecución del Script creado, mostrando los 3 posibles casos planteados por el ejercicio.

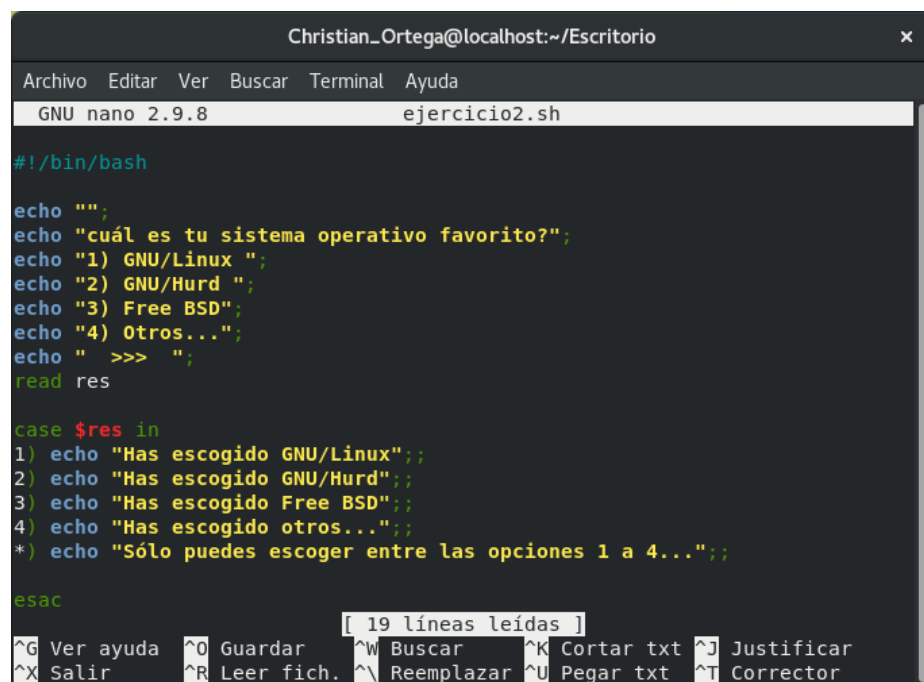
2.2 Ejercicio 2

Se plantea el siguiente ejercicio:

Crea un script que presente el menú de opciones siguiente (y la respuesta indicada):

- ¿Cuál es tu sistema operativo favorito?
- 1) GNU/Linux.
- 2) GNU/Hurd.
- 3) Free BSD.
- 4) Otros.
- ?

La solución de este problema es de la misma forma que de la del problema anterior, debemos usar uso de una estructura de control tipo switch, cuya sintáxis definimos en el problema anterior:



```
Christian_Ortega@localhost:~/Escritorio
GNU nano 2.9.8 ejercicio2.sh

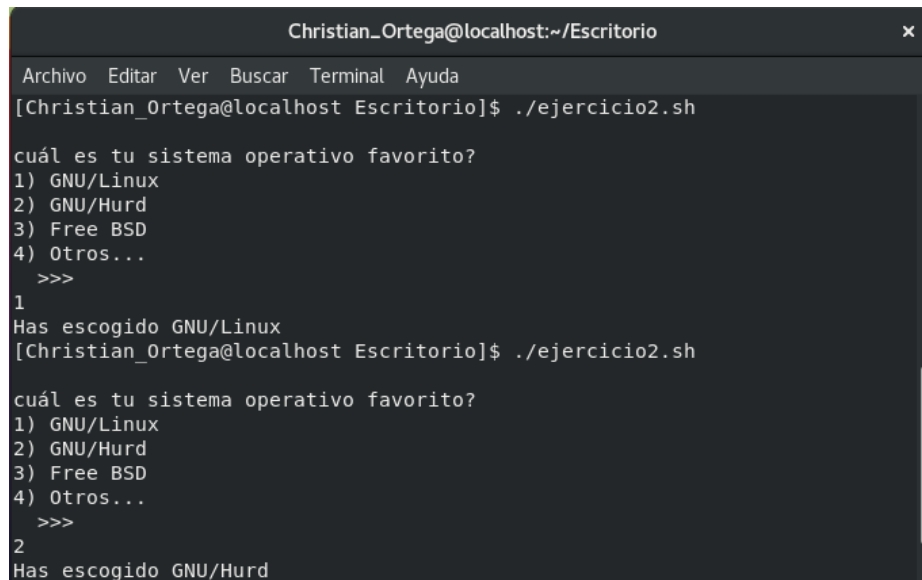
#!/bin/bash

echo ""
echo "cuál es tu sistema operativo favorito?"
echo "1) GNU/Linux ";
echo "2) GNU/Hurd ";
echo "3) Free BSD";
echo "4) Otros...";
echo " >>> ";
read res

case $res in
1) echo "Has escogido GNU/Linux";;
2) echo "Has escogido GNU/Hurd";;
3) echo "Has escogido Free BSD";;
4) echo "Has escogido otros...";;
*) echo "Sólo puedes escoger entre las opciones 1 a 4...";;
esac

[ 19 líneas leídas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar txt ^J Justificar
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Corrector
```

Fig. 6. Script creado para cumplir el propósito del ejercicio 2.



```
Christian_Ortega@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[Christian_Ortega@localhost Escritorio]$ ./ejercicio2.sh

cuál es tu sistema operativo favorito?
1) GNU/Linux
2) GNU/Hurd
3) Free BSD
4) Otros...
>>>
1
Has escogido GNU/Linux
[Christian_Ortega@localhost Escritorio]$ ./ejercicio2.sh

cuál es tu sistema operativo favorito?
1) GNU/Linux
2) GNU/Hurd
3) Free BSD
4) Otros...
>>>
2
Has escogido GNU/Hurd
```

Fig. 7. Resultado de la ejecución del script creado, mostrando los primeros 2 casos planteados por el ejercicio.



```
Christian_Ortega@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[Christian_Ortega@localhost Escritorio]$ ./ejercicio2.sh

cuál es tu sistema operativo favorito?
1) GNU/Linux
2) GNU/Hurd
3) Free BSD
4) Otros...
>>>
3
Has escogido Free BSD
[Christian_Ortega@localhost Escritorio]$ ./ejercicio2.sh

cuál es tu sistema operativo favorito?
1) GNU/Linux
2) GNU/Hurd
3) Free BSD
4) Otros...
>>>
4
Has escogido otros...
```

Fig. 8. Resultado de la ejecución del script creado, mostrando los casos (3 y 4) planteados por el ejercicio.



```
Christian_Ortega@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
cuál es tu sistema operativo favorito?
1) GNU/Linux
2) GNU/Hurd
3) Free BSD
4) Otros...
>>>
e
Sólo puedes escoger entre las opciones 1 a 4...
[Christian_Ortega@localhost Escritorio]$ ./ejercicio2.sh

cuál es tu sistema operativo favorito?
1) GNU/Linux
2) GNU/Hurd
3) Free BSD
4) Otros...
>>>
j
Sólo puedes escoger entre las opciones 1 a 4...
```

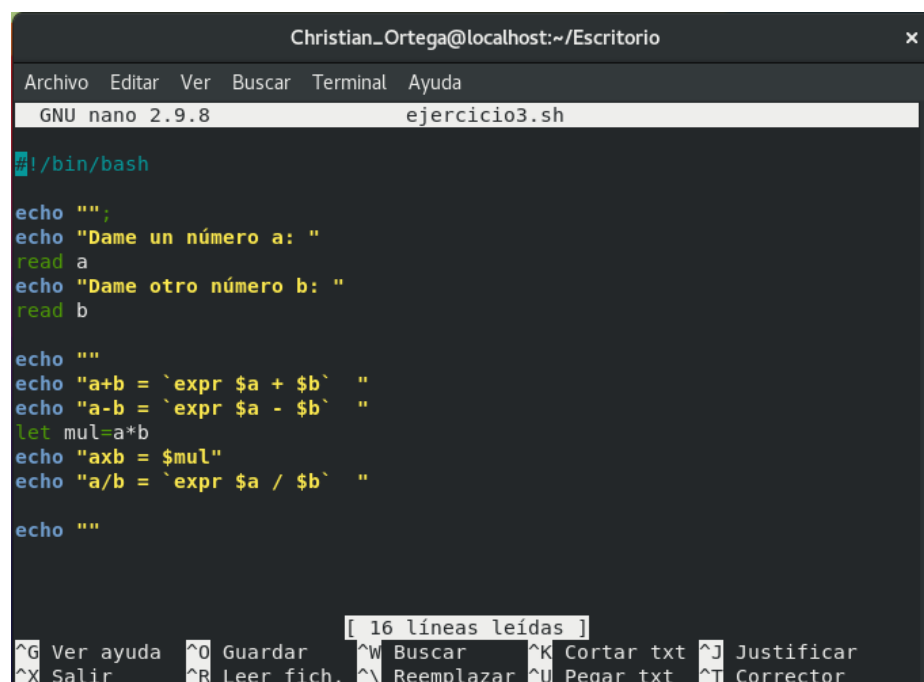
Fig. 9. Resultado de la ejecución del script creado, mostrando 2 casos de error (en caso de que el usuario ingrese datos de tipo NO numérico).

2.3 Ejercicio 3

Se plantea el siguiente ejercicio:

Crea un script que después de ejecutarse pida al usuario 2 números y después presente la suma, la resta, el product y la division.

La solución de este problema está (además de la salida de datos (mensajes) y entrada de datos (valor de a y b)), en la pura sintáxis de las operaciones aritméticas que nos piden hacer. (Sintáxis documentada en la sección de referencias) Notemos que las operaciones las hacemos al mismo tiempo que vamos mostrando el mensaje de resultado, de modo que no necesitamos declarar variables (a excepción de la multiplicación, que necesita una sintáxis diferente).



```
Christian_Ortega@localhost:~/Escritorio
GNU nano 2.9.8 ejercicio3.sh

#!/bin/bash

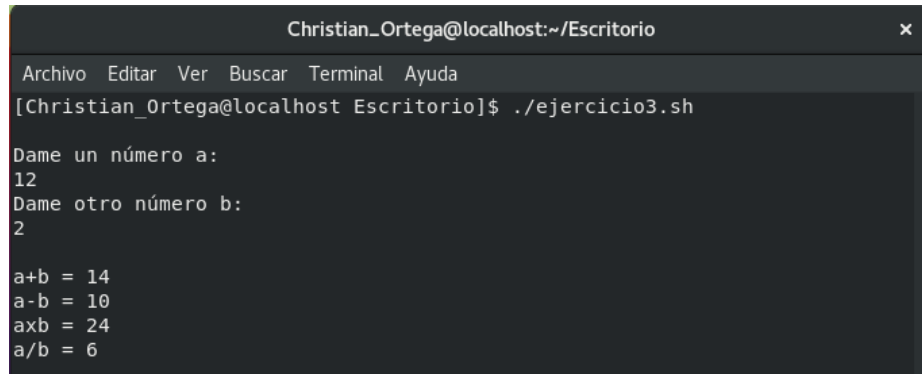
echo ""
echo "Dame un número a: "
read a
echo "Dame otro número b: "
read b

echo ""
echo "a+b = `expr $a + $b` "
echo "a-b = `expr $a - $b` "
let mul=a*b
echo "axb = $mul"
echo "a/b = `expr $a / $b` "

echo ""

[ 16 líneas leídas ]
^G Ver ayuda  ^O Guardar  ^W Buscar  ^K Cortar txt  ^J Justificar
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar txt  ^T Corrector
```

Fig. 10. Script creado para cumplir el propósito del ejercicio 3.



```
Christian_Ortega@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[Christian_Ortega@localhost Escritorio]$ ./ejercicio3.sh
Dame un número a:
12
Dame otro número b:
2
a+b = 14
a-b = 10
a*x = 24
a/b = 6
```

Fig. 11. Resultado de la ejecución del script creado, mostrando los 4 resultados de las operaciones pedidas.



```
Christian_Ortega@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[Christian_Ortega@localhost Escritorio]$ ./ejercicio3.sh
Dame un número a:
4
Dame otro número b:
r
expr: argumento no entero
a+b =
expr: argumento no entero
a-b =
a*x = 0
expr: argumento no entero
a/b =
```

Fig. 12. Resultado de la ejecución del script creado, mostrando el comportamiento del programa al ingresar datos de tipo NO numérico (error).

3 Conclusiones

El uso del shell de Linux para crear scripts es una habilidad fundamental para cualquier usuario de Linux. Los comandos básicos como mover, copiar archivos, cambiar permisos y buscar información en el Sistema son esenciales para el manejo diario del Sistema operativo y permiten a los usuarios trabajar de manera más eficiente y efectiva la línea de comandos.

Sin embargo, para aprender y dominar comandos con usos prácticos reales y complejos, hay que empezar primero por lo más básico; entrada/salida, sintaxis, palabras reservadas, entorno, claves, etc...

4 Referencias

El siguiente enlace nos sirvió para conocer una forma alternativa de implementar la multiplicación en GNU NANO, pues los intentos convencionales fallaron:

<https://hazlolinux.com/programacion-bash/operaciones-aritmeticas-de-bash/>

Aunque los Scripts implementados en esta práctica son muy sencillos, esta referencia nos sirvió como una Buena consulta:

http://trajano.us.es/~fjf/shell/FAST_t3-practica.pdf

Esta suele ser la primera forma en la que se nos presentan las operaciones aritméticas en GNU NANO, aunque no siempre funciona la sintaxis que aquí se presenta:

<https://uniwebsidad.com/foro/pregunta/300/como-se-pueden-sumar-dos-numeros-en-un-script-de-bash/>