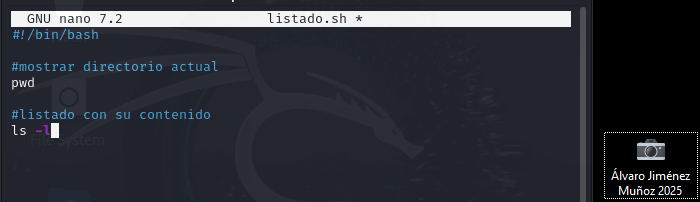
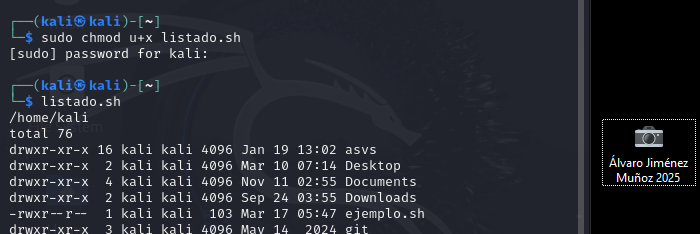
**Ejercicios ShellScripts I**

Recuerda que una vez realizado el script debes darle permisos de ejecución para poder lanzarlo y ver el resultado por pantalla.

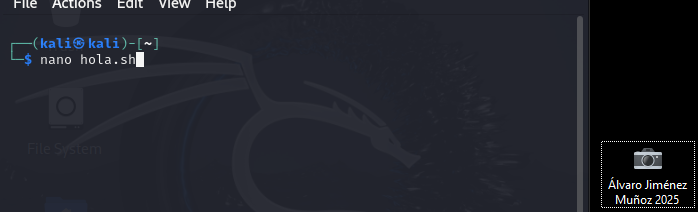
1. Crea un script denominado listado.sh que indique el directorio en el que nos encontramos y a continuación muestre un listado con su contenido.

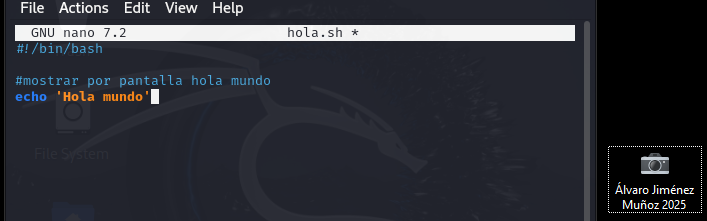


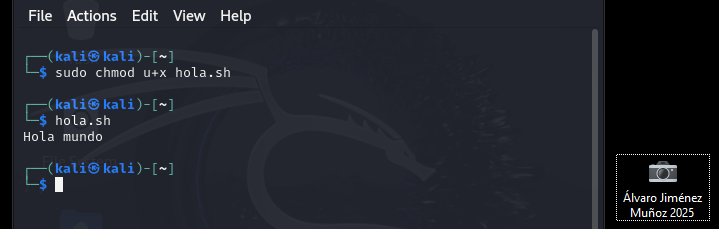




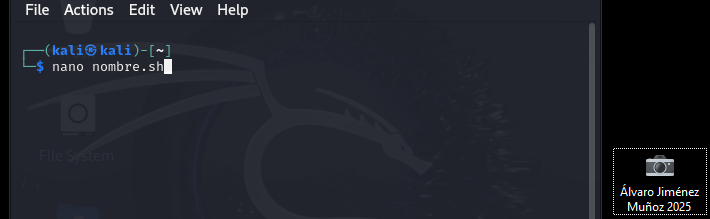
1. Crea un script denominado hola.sh que muestre en pantalla el mensaje ‘Hola mundo’.

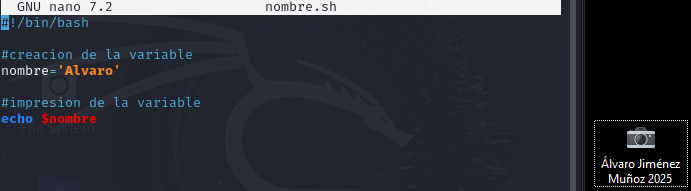


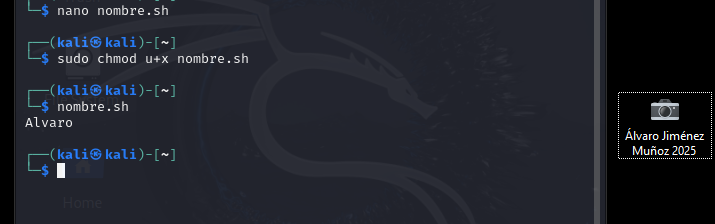




1. Crea un script donde crees una variable que se llame *nombre*. Asígnale tu nombre y la muestre por pantalla.

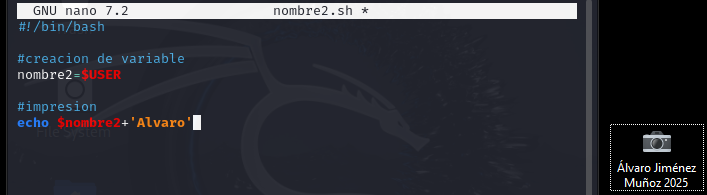






1. Crea un script donde crees una variable que se llame *nombre* asignándole el valor de la variable de entorno USER, concatena tu nombre y la muestras por pantalla.

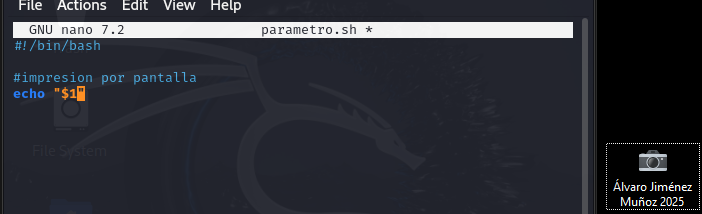


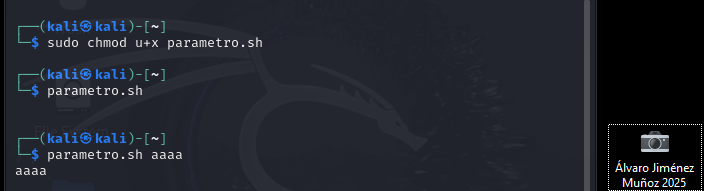




1. Crea un script que reciba un parámetro por línea de comandos y lo muestre por pantalla.







1. Crea un script denominado parametros.sh que al lanzarlo muestre la siguiente información:

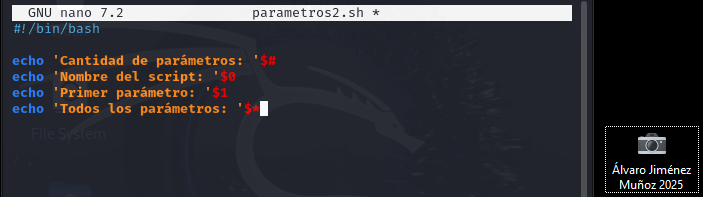
*Cantidad de parámetros:*

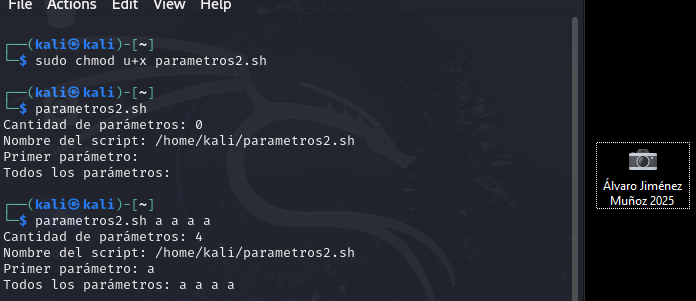
*Nombre del script:*

*Primer parámetro:*

*Todos los parámetros:*

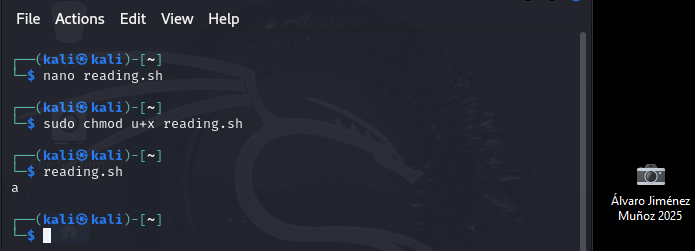
**

**

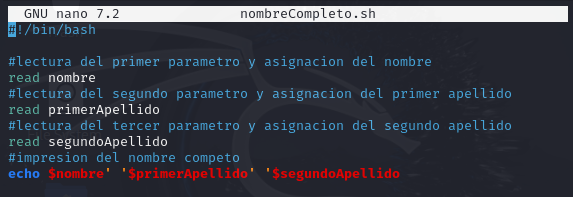
**

1. Crea un script que reciba un parámetro por teclado (read) y lo muestre por pantalla.

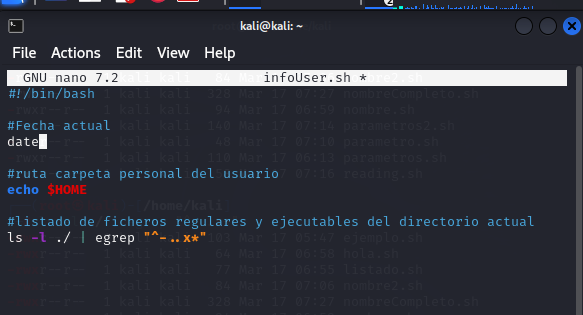




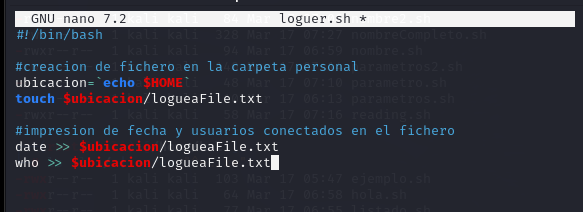
1. Crea un script denominado nombreCompleto que reciba por línea de comandos tres parámetros: nombre, primer y segundo apellido; y cree una variable con el nombre completo. Muestra después el nombre completo por pantalla.



1. Crea un script denominado infoUser.sh que muestre la fecha actual, la ruta de la carpeta personal del usuario (variable HOME) y realice un listado de los ficheros regulares y ejecutables que hay en el directorio actual.



1. Crea un script denominado loguer.sh que a su vez cree un fichero en la carpeta personal denominada logueaFile.txt con la fecha y los usuarios conectados en ese momento.



1. Crea un script denominado paramDirectorio.sh al que se le pase 3 parámetros por línea de comandos. El script debe comprobar si son directorios. En caso de ser directorio mostrará un listado con su contenido y si no lo es, entonces mostrará un mensaje por pantalla indicándolo.
2. Crea un script denominado compParam.sh que reciba dos parámetros numéricos y compruebe si el primero es mayor que el segundo

.

1. Modifica el ejercicio anterior para que indique qué parámetro es el mayor y si son iguales, que lo indique.
2. Crea un script denominado compCadenas.sh que reciba dos parámetros de tipo texto. Primero se debe comprobar si alguna de las cadenas está vacía (su longitud es cero), en ese caso indicará un mensaje por pantalla. Si las cadenas no están vacías, las comprobará e indicará si son iguales.
3. Crea un script denominado existFile.sh que reciba como parámetro el nombre de un fichero y compruebe si existe.
4. Crea un script denominado tipoFile.sh que reciba como parámetro el nombre de un fichero y compruebe si es un directorio o un fichero normal. Si es otro tipo de fichero, indicará que el fichero no es ni un directorio ni un fichero normal.
5. Crea un script denominado compFicheros.sh que reciba como parámetro los nombres de dos ficheros e indique cuál de ellos es más reciente.
6. Modifica el ejercicio anterior de forma que controles toda la casuística (problemas) que se puedan producir a la hora de pasar los parámetros:

* Si el número de parámetros es menor que dos 🡪 Deberá aparecer por pantalla un mensaje “El número de parámetros no es correcto. Uso compFicheros.sh fichero1 fichero2”
* Si uno de los ficheros introducidos como parámetro no existe 🡪 Deberá aparecer por pantalla un mensaje “Los parámetros introducidos no son válidos”.

1. Crear un script que indique si hay 1 o más usuarios conectados. Para crear una variable que cuente el número de usuarios conectados lo puedes hacer así:

*numUsuarios=’who | wc –l’*

En caso de haber un solo usuario, el programa mostrará un mensaje por pantalla diciendo “Hay un solo usuario conectado”, mientras que si hay más de uno “Hay más de un usuario conectado”.

1. Crea un script denominado ejecuta.sh que reciba como parámetro el nombre de un archivo, controle si es regular, a continuación lo hace ejecutable (modificación de permisos – ejecución para todos), compruebe si tiene extensión .sh, y si no es así, se la pone.
2. Crea un script denominado nota.sh que reciba un parámetro por teclado (read) que sea la nota de un examen e indique un mensaje por pantalla si se corresponde con un INS, SUF, BIEN, NOT o SOB.
3. Crea un script para que aparezca un menú del siguiente tipo:

*1. vi*

*2. nano*

*3. gedit*

*4. salir*

*Introduzca una opción*

De forma que lea un parámetro por teclado y lance en segundo plano el editor seleccionado. En caso de pulsar 4, el programa finalizará.

1. Crea un script denominado cuenta.sh que contenga un bucle de repetición for que cuente del 1 al 10. Utiliza la siguiente condición *for num in `seq 1 10`*
2. Realiza el ejercicio anterior, usando la siguiente estructura e indica las diferencias.

contador=1

while test $conter -le 10

do

echo $contador

cotnador=`expr $contador + 1`

done

1. Crea un script denominado tablaMult.sh que reciba un parámetro (número) y muestre su tabla de multiplicar.
2. Crea un script que muestre todos los números pares del 1 al 100
3. Crea un script que muestre todos los números pares desde un número que pida por teclado hasta el 100.
4. Crea un script que muestre todos los números múltiplos de 3 desde el 1 hasta el 100.
5. Crea un script que muestre todos los múltiplos de 6 del 1 al 100. Para ello se tendrá que tener en cuenta que un número es múltiplo de 6 cuando lo es de 2 y de 3 a la vez.