

1. Repositorio GitHub

Para clonar un repositorio de GitHub en la maquina virtual tenemos que utilizar el comando *git clone* + link del repositorio de GitHub: <https://github.com/Alexalopezm/Lab1.git>

2. Solución. Parte 1:

1. Ya que se va a trabajar en conjunto con un repositorio en git, se crea un primer script llamado *"verificador.sh"* utilizando el comando *vim* visto en clase, en la carpeta del repositorio.
2. Utilizando la condición *if* [*\$# -ne 1*] se logra verificar si se ha proporcionado exactamente un argumento en la línea de comandos al script. *\$#* es una variable especial que contiene el número de argumentos. Si no se proporciona un solo argumento, muestra un mensaje de error y finaliza el script con un código de retorno no cero (1).
3. Luego dentro de la variable **name** se almacena el argumento proporcionado (nombre del archivo) para su posterior uso.
4. Se comprueba si el archivo especificado por **name** existe utilizando el comando [*-e "\$name"*]. Si existe, muestra un mensaje indicando que el archivo existe.
5. Se utiliza el comando *stat* para obtener los permisos del archivo y los almacena en la variable *permissions*.
6. Luego se define una función llamada **"get_permissions_verbose"** que se utilizará para analizar y mostrar los permisos de manera detallada por medio de *if else fi*.
7. Se llama a la función **"get_permissions_verbose"** y pasa los permisos almacenados en **permissions** como argumento.
8. Por último, si el archivo no existe, muestra un mensaje de error y finaliza el script con un código de retorno 2.
9. Finaliza el script con un código de retorno cero, indicando que se ha completado sin errores.
10. Prueba al ejecutar el código:

```
alexam@maquinav:~/plataformas/Lab1$ ./verificador.sh
Error ingrese el nombre del archivo.
Uso: ./verificador.sh <nombre_archivo>
alexam@maquinav:~/plataformas/Lab1$ ./verificador.sh texto.txt
El archivo 'texto.txt' no existe.
alexam@maquinav:~/plataformas/Lab1$ ./verificador.sh enunciado.pdf
El archivo 'enunciado.pdf' existe.
El usuario tiene permisos de lectura.
El usuario tiene permisos de escritura.
El usuario no tiene permisos de ejecutar.
El grupo tiene permisos de lectura.
El grupo tiene permisos de escritura.
El grupo no tiene permisos de ejecutar.
Los otros usuarios tienen permisos de lectura.
Los otros usuarios no tienen permisos de escritura.
Los otros usuarios no tienen permisos de ejecutar.
```

Figura 1: Prueba en terminal del script verificador.sh

3. Solución. Parte 1:

1. Ya que se va a trabajar en conjunto con un repositorio en git, se crea un primer script llamado *"permisos.sh"* utilizando el comando *vim* visto en clase, en la carpeta del repositorio.
2. Utilizando la condición *if* [*## -ne 2*] se logra verificar si se ha proporcionado exactamente dos argumentos en la línea de comandos al script. *##* es una variable especial que contiene el número de argumentos. Si no se proporcionan dos argumentos, muestra un mensaje de error y corrección de entrada.
3. Luego dentro de las variables **name_user** y **name_group** se almacenan los argumentos proporcionados para su posterior uso.
4. Para verificar si el nombre del nuevo usuario existe se utiliza el comando *if* con la condición *id "\$name_user"& devnull*, si no existe se utiliza el comando *sudo useradd "\$name_user"* para crear el nuevo usuario con el nombre proporcionado.
5. Para verificar si el nombre del nuevo grupo existe se utiliza el comando *if* con la condición *grep -q "\$name_group : "/etc/group*, si no existe se utiliza el comando *sudo groupadd "\$name_group* para crear el nuevo grupo con el nombre proporcionado.
6. Para agregar a el usuario default y el nuevo usuario se utilizó el comando *sudo usermod -aG "\$name_group" "\$USER"* para el usuario default y para el nuevo usuario se utiliza el mismo comando pero se sustituye *"\$USER"* por *"\$name_user"*.
7. Para asignar permisos de ejecución al script *"verificador.sh"* para que solo los miembros del grupo creado puedan ejecutarlo se utilizo *sudo chown : "\$name_group" verificador.sh* y *sudo chmod 750 verificador.sh*, donde 750 en binario se puede representar como 111 101 000 y eso representa que los permisos sean *-rwxr-x---*
8. Prueba al ejecutar el código:

```
alexa@maquinav:~/plataformas/Lab1$ ./permisos.sh
Error ingrese el nombre de usuario y nombre de grupo
Uso: ./permisos.sh <nombre_de_usuario> <nombre_de_grupo>
alexa@maquinav:~/plataformas/Lab1$ ./permisos.sh kyllian lab1
[sudo] contraseña para alexa:
Se ha creado el usuario 'kyllian'.
El grupo 'lab1' ya existe.
Se ha agregado el usuario 'alexa' al grupo 'lab1'.
Se ha agregado el usuario 'kyllian' al grupo 'lab1'.
Se han asignado permisos de ejecución para miembros del grupo 'lab1' al script
'verificador.sh'.
alexa@maquinav:~/plataformas/Lab1$ ls -lah
total 264K
drwxrwxr-x 3 alexa alexa 4.0K sep  4 23:05 ./
drwxrwxr-x 4 alexa alexa 4.0K sep  4 16:50 ../
-rw-rw-r-- 1 alexa alexa 240K sep  4 16:50 enunciado.pdf
drwxrwxr-x 8 alexa alexa 4.0K sep  4 23:48 .git/
-rwxrwxr-x 1 alexa alexa 1.4K sep  4 23:37 permisos.sh*
-rw-rw-r-- 1 alexa alexa   70 ago 29 18:47 README.md
-rwxr-x--- 1 alexa lab1  2.6K sep  4 23:37 verificador.sh*
alexa@maquinav:~/plataformas/Lab1$ ./permisos.sh loky lab1
El usuario 'loky' ya existe.
El grupo 'lab1' ya existe.
Se ha agregado el usuario 'alexa' al grupo 'lab1'.
Se ha agregado el usuario 'loky' al grupo 'lab1'.
Se han asignado permisos de ejecución para miembros del grupo 'lab1' al script
'verificador.sh'.
```

Figura 2: Prueba en terminal del script permisos.sh