







# PR\_01.1

## Git/GitHub

-  Instalar Git en tu ordenador.
-  Darse de alta en GitHub.
-  Crea un repositorio en GitHub para almacenar las prácticas del módulo, por ejemplo, `BigData2526` Hazlo público.
-  Clona el repositorio remoto en local.
-  A partir de ahora, vete guardando los archivos de todas las prácticas en la carpeta del punto anterior. Para cada práctica, crea una subcarpeta con su nombre, por ejemplo, `PR_01.1`
-  No te olvides de actualizar el repositorio con regularidad y sobre todo, al finalizar las prácticas.
- **Envíame la URL del repositorio por el chat de Teams.**

## Introducción a Linux

- *Para cada pregunta, añade capturas de pantalla donde se muestre la ejecución de los comandos que se pide.*
- *Entrega la práctica en formato `pdf` . El archivo llámalo con el número de la práctica + tu nombre, por ejemplo: `PR_01.1_Juan_García.pdf`*

### 1.- Conceptos Básicos y Variables

1. Muestra el contenido de tu variable de entorno `HOME` . Luego, usa `cd` junto con esa variable para navegar a dicho directorio y verifica con `pwd` que te encuentras en la ubicación correcta.
2. Ejecuta el comando `whoami` . Ahora, crea una variable local llamada `USUARIO_ACTUAL` que contenga el resultado del comando anterior y muéstrala en la terminal.
3. Intenta crear un archivo llamado `dos palabras.txt` sin usar comillas. Observa el resultado con `ls` . ¿Qué ha ocurrido y por qué? Ahora, bórralo(s) y créalo correctamente.

4. Usa el comando `type` para averiguar si `ls` y `cd` son internos o externos al shell. ¿Qué diferencia práctica crees que implica esto?
5. Muestra tu `PATH` actual. Crea un directorio `~/mi_bin` y añádelo temporalmente al *principio* de tu `PATH`. Verifica que el cambio se ha realizado correctamente.

## 2.- Obtener Ayuda y Localizar Archivos

1. Abre la página del manual para el comando `chmod`. ¿En qué sección del manual se encuentra? ¿Qué indica ese número de sección sobre el tipo de comando?
2. Usando la función de búsqueda dentro de la página del manual de `ls`, encuentra la opción que ordena los archivos por tamaño.
3. Imagina que has olvidado dónde se guarda el archivo de configuración de usuarios. Sabiendo que se llama `passwd`, usa `find` para buscarlo desde el directorio raíz (`/`). Anota la ruta completa que has encontrado.
4. Crea un archivo vacío llamado `test_locate.txt` en tu directorio home. Inmediatamente después, búscalo con `locate`. ¿Aparece en los resultados? ¿Por qué sí o por qué no?
5. Basado en el ejercicio anterior, ¿qué comando (probablemente con `sudo`) necesitas ejecutar para que `locate` sí encuentre tu archivo? Ejecútalo y verifica que ahora sí lo encuentras.

## 3.- Navegación y Listado de Archivos

1. Navega al directorio `/etc`. Desde ahí, sin usar `cd`, lista el contenido de tu directorio `home` usando una ruta con el atajo `~`.
2. Desde tu directorio `home`, navega a `/` y luego a `var` y finalmente a `log` usando una sola línea de comando y rutas relativas.
3. Lista el contenido de `/etc` en formato largo. En la salida, identifica el propietario, el grupo y los permisos del archivo `passwd`.
4. Compara la salida de `ls -l /etc` y `ls -lh /etc`. ¿Qué hace la opción `h` y por qué es útil para las personas?
5. Ejecuta `ls -R ~`. ¿Qué hace la opción `R`? ¿Por qué podría ser peligroso usarla en el directorio raíz (`/`)?

## 4.- Manipulación de Archivos y Directorios

1. Crea la estructura de directorios `proyecto/src` , `proyecto/doc` y `proyecto/bin` usando un único comando `mkdir` .
  2. Crea un archivo `~/notas.txt` . Muévelo a `~/proyecto/doc` y, en el mismo comando, renómbralo a `README.md` .
  3. Copia el archivo `README.md` de `proyecto/doc` a `proyecto/bin` . Luego, borra el archivo original de la carpeta `doc` .
  4. Intenta borrar el directorio `proyecto` con `rmdir` . ¿Qué error obtienes? Ahora, usa `rm` con la opción correcta para borrar el directorio y todo lo que contiene.
  5. Navega a `/etc` . Usando un solo comando `ls` con `globbing` , lista todos los archivos que empiecen con la letra `s` y terminen con `.conf` .
- 

## 5.- Archivado y Compresión

1. Crea un archivo `tar` llamado `log_backup.tar` que contenga todos los archivos del directorio `/var/log` . ¿Qué advertencias de "permiso denegado" aparecen y por qué?
2. Comprime el archivo `log_backup.tar` con `gzip` . Compara el tamaño del archivo original y el comprimido usando `ls -lh` .
3. Lista el contenido del archivo `log_backup.tar.gz` sin extraerlo para verificar que los archivos están dentro.
4. Extrae únicamente el archivo `syslog` (o `messages` ) de `log_backup.tar.gz` a tu directorio `/tmp` .
5. Crea tres archivos ( `a.txt` , `b.log` , `c.jpg` ) y luego crea un archivo `zip` que los contenga.
6. Elimina los tres archivos originales y luego recupéralos desde el archivo `zip` .
7. Usa `zcat` (o `gzcat` ) para leer el contenido de un archivo de log comprimido (ej: en `/var/log` , busca uno que termine en `.gz` ) sin crear un archivo descomprimido.

## 6.- Redirección, Tuberías y Filtros

1. Guarda la lista de archivos de tu directorio home (formato largo) en un archivo `mis_archivos.txt`.
2. Sin borrar el contenido anterior, añade la fecha y hora actual al final del archivo `mis_archivos.txt`.
3. Usa `grep` y una tubería (`|`) para contar el número de directorios que hay en `/etc`. (Pista: `ls -l | grep '^d'`).
4. Muestra las 10 últimas líneas del archivo `/etc/passwd` y, usando otra tubería, extrae solo los nombres de usuario (el primer campo).
5. Muestra una lista de todos los procesos del sistema (`ps aux`), ordénala por uso de CPU (tercera columna) y muestra solo las 5 líneas superiores.
6. ¿Cuál es la diferencia entre usar `>` y `>>` para redirigir la salida de un comando a un archivo? Demuéstralo con un ejemplo.
7. Ejecuta `find /etc -name "*.conf"`. Redirige la salida estándar a un archivo `config_files.txt` y los errores (si los hay) a `errors.txt`.

## 7.- Scripts Básicos

1. Crea un script que imprima tu nombre de usuario y el directorio de trabajo actual usando las variables de entorno correspondientes.
2. Haz el script anterior ejecutable solo para ti (`chmod u+x ...`) y ejecútalo. Luego, intenta ejecutarlo como otro usuario (si es posible) o explica qué pasaría.
3. Modifica el script para que acepte un argumento. Si el argumento es "hola", debe imprimir "mundo". Si es cualquier otra cosa, no debe imprimir nada.
4. Mejora el script anterior para que, si no se proporciona ningún argumento, muestre un mensaje de uso: "Error: Debes proporcionar un argumento."
5. Escribe un script que reciba dos números. Debe imprimir "iguales" si son iguales y "diferentes" si no lo son.
6. Escribe un script que, dado un directorio como argumento, use un bucle `for` para iterar sobre su contenido (`ls $1`) y añada la extensión `.bak` a cada archivo.

---

## 8.- Ejercicios Avanzados

1. Muestra los shells de los usuarios listados en `/etc/passwd`, elimina las líneas duplicadas y ordénalos alfabéticamente. (Pista: `cut`, `sort`, `uniq`).

2. Usando `ps`, `grep` y `wc`, crea un comando de una sola línea que te diga cuántos procesos está ejecutando el usuario `root` actualmente.
3. Lista todos los archivos en `/etc`, filtra los resultados para mostrar solo aquellos que han sido modificados en "Oct" (octubre) y guarda esa lista en `october_files.txt`.
4. Usando `globbing`, lista todos los archivos en `/etc` que contengan un número en su nombre.
5. Usando `find`, busca en `/usr/bin` todos los archivos que sean ejecutables, pero que no sean propiedad del usuario `root`.
6. Compara la diferencia de tamaño y velocidad al comprimir un archivo grande (puedes usar `/var/log/syslog`) con `gzip` y con `bzip2`.
7. Crea un archivo `tar` de tu directorio home, pero esta vez, usa la opción para seguir enlaces simbólicos. Antes, crea un enlace simbólico en tu home para que puedas ver la diferencia.
8. Escribe un script que reciba una ruta a un archivo. Debe verificar si es un archivo regular, un directorio o si no existe, mostrando un mensaje diferente en cada caso. (Pista: `if [ -f ... ]`, `if [ -d ... ]`).
9. Crea un script que intente crear un directorio llamado `test_dir` en `/`. Usando el código de salida (`$?`), el script debe informar si tuvo éxito o si falló por un problema de permisos.
10. Escribe un script que reciba cualquier número de argumentos. El script debe iterar sobre ellos y solo imprimir aquellos que sean números mayores que 10.