

Práctica 02

DOCENTE	CARRERA	CURSO
MSc. Maribel Molina Barriga	Escuela Profesional de Ingeniería de Software	Sistemas Operativos

GRUPO	TEMA	DURACIÓN
6	Comandos en Windows y Linux	5 horas

Integrantes

- José Carlos Machaca Vera
- Jhosep Alonso Mollapaza Morocco
- Patrick Andres Ramirez Santos

Índice

1. Objetivos	2
2. Software utilizado	2
3. Windows	3
3.1. Ejecutamos la consola	3
3.2. Creamos carpeta LABSO	3
3.3. Creamos directorios dentro de LABSO	3
3.4. Usamos el comando TREE	4
3.5. Archivo .txt y comandos DIR, TYPE	4
3.6. Copiar y renombrar datos.txt	6
3.7. Probando RD, DEL y diferentes parámetros de DIR	7
4. Linux	8
4.1. Manipulando el shell	8
4.1.1. Preguntas Shell	8
4.1.2. Preguntas Direccionamiento	10
4.2. Manipulando archivos y directorios	11
5. Conclusiones	13
6. Recomendaciones	13

1. Objetivos

Se busca demostrar como la comprensión de los sistemas operativos a través de la ejecución práctica de comandos en Windows y Linux ayuda a mejorar la versatilidad de los desarrolladores de software. Además se desea tener una guía para manejar archivos y directorios, instalar programas y reconocer las diferencias al usar la terminal.

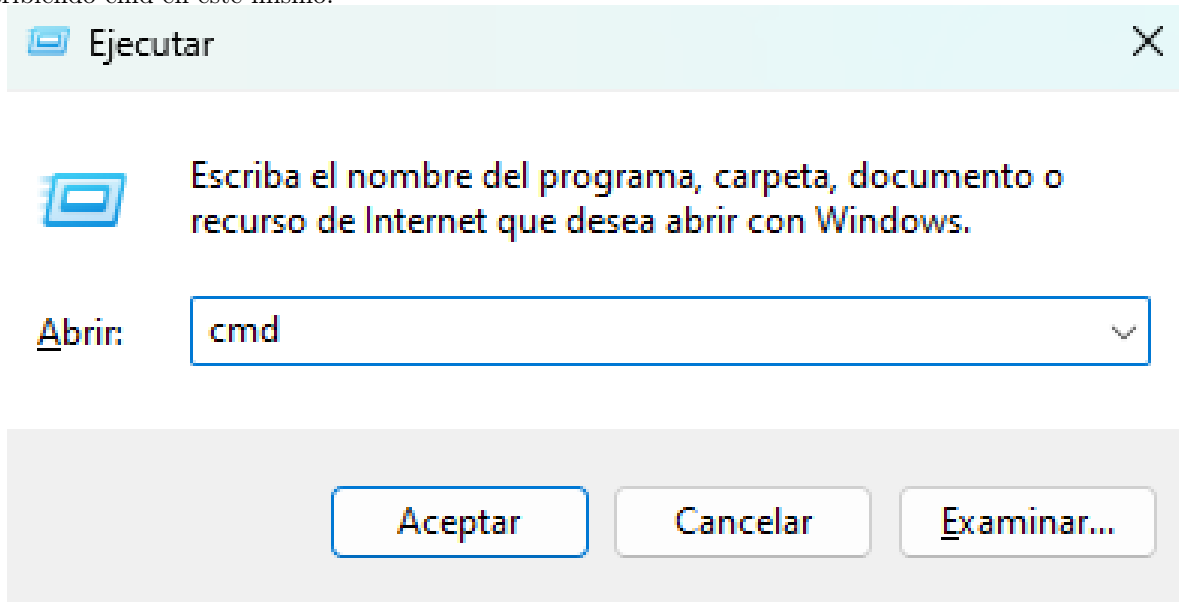
2. Software utilizado

- Overleaf
- Windows
 - Windows 11
 - Símbolo de Sistema
 - Blog de notas
- Linux
 - Fedora 39
 - Warp Terminal
 - Neovim

3. Windows

3.1. Ejecutamos la consola

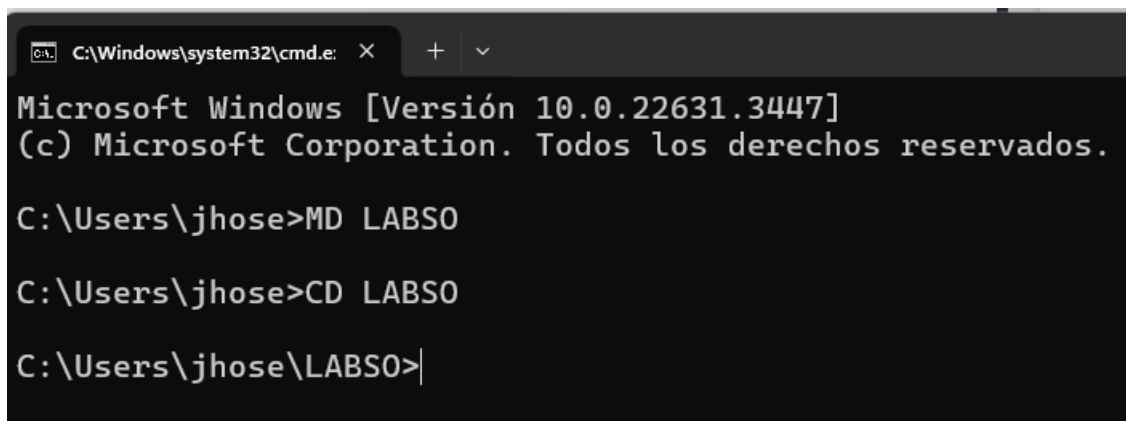
Se realiza mediante el cuadro de diálogo Ejecutar con la combinación de teclas Windows R y escribiendo cmd en este mismo.



3.2. Creamos carpeta LABSO

Una vez dentro del cmd, creamos la carpeta LABSO con el comando MD, una vez creada para ingresar a dicha carpeta es mediante el comando CD.

```
C:\Users\jhose>MD LABSO  
C:\Users\jhose>CD LABSO
```



3.3. Creamos directorios dentro de LABSO

Una vez creada la carpeta LABSO y dentro, creamos otras 2 carpetas, que son MEMORIA y PROCESOS, luego dentro de estas 2 carpetas creadas se crean otras 2 carpetas llamadas TEORIA y PRACTICA.

```
C:\Users\jhose\LABSO>MD MEMORIA PROCESOS
C:\Users\jhose\LABSO>CD MEMORIA
C:\Users\jhose\LABSO\MEMORIA>MD TEORIA PRACTICA
C:\Users\jhose\LABSO\MEMORIA>CD..
C:\Users\jhose\LABSO>CD PROCESOS
C:\Users\jhose\LABSO\PROCESOS>MD TEORIA PRACTICA
C:\Users\jhose\LABSO\PROCESOS>CD..
```

```
C:\Users\jhose\LABSO>MD MEMORIA PROCESOS
C:\Users\jhose\LABSO>CD MEMORIA
C:\Users\jhose\LABSO\MEMORIA>MD TEORIA PRACTICA
C:\Users\jhose\LABSO\MEMORIA>CD..
C:\Users\jhose\LABSO>CD PROCESOS
C:\Users\jhose\LABSO\PROCESOS>MD TEORIA PRACTICA
C:\Users\jhose\LABSO\PROCESOS>CD..
C:\Users\jhose\LABSO>|
```

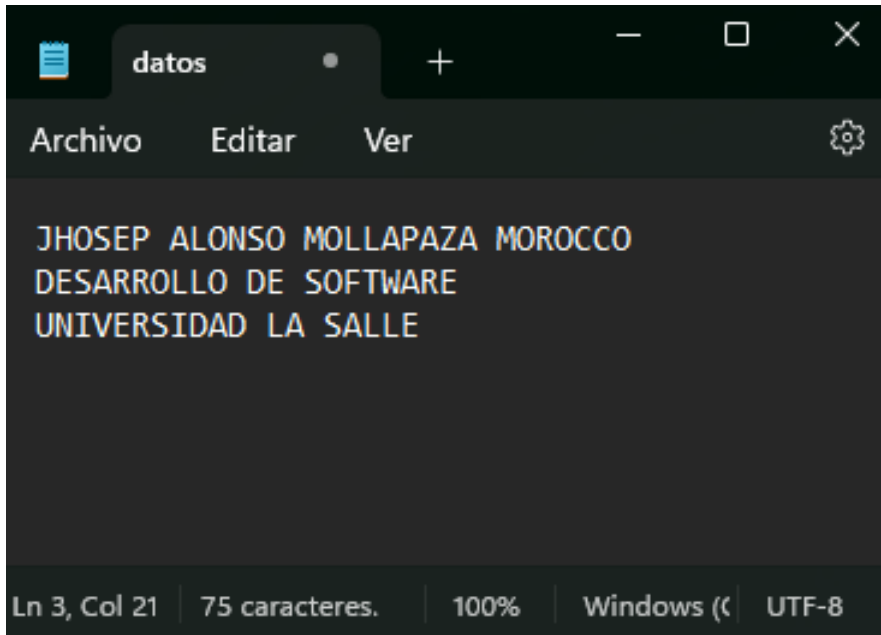
3.4. Usamos el comando TREE

Volvemos a la carpeta LABSO y ejecutamos el comando TREE para observar las carpetas que hay dentro de esta misma.

```
C:\Users\jhose\LABSO>TREE
Listado de rutas de carpetas
El número de serie del volumen es C2DA-2117
C:.
├── MEMORIA
│   ├── PRACTICA
│   └── TEORIA
└── PROCESOS
    ├── PRACTICA
    └── TEORIA
```

3.5. Archivo .txt y comandos DIR, TYPE

Creamos un archivo .txt dentro de la carpeta MEMORIA.



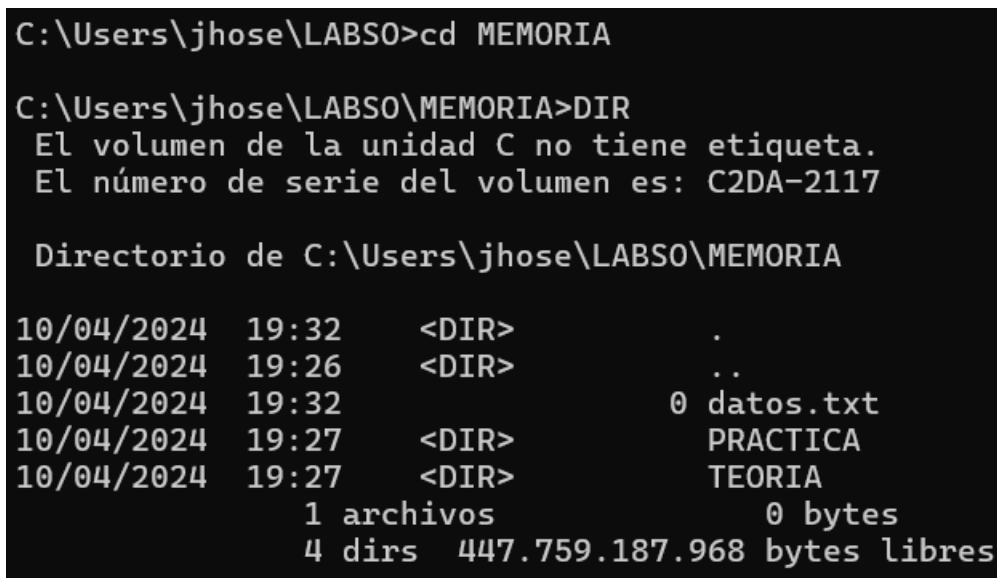
```
Archivo  Editar  Ver  [Settings Icon]

JHOSEP ALONSO MOLLAPAZA MOROCCO
DESARROLLO DE SOFTWARE
UNIVERSIDAD LA SALLE

Ln 3, Col 21 | 75 caracteres. | 100% | Windows (C | UTF-8
```

Luego usamos el comando DIR dentro de MEMORIA para listar los archivos y/o directorios.

```
C:\Users\jhose\LABSO>cd MEMORIA
C:\Users\jhose\LABSO\MEMORIA>DIR
```



```
C:\Users\jhose\LABSO>cd MEMORIA

C:\Users\jhose\LABSO\MEMORIA>DIR
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: C2DA-2117

Directorio de C:\Users\jhose\LABSO\MEMORIA

10/04/2024  19:32    <DIR>          .
10/04/2024  19:26    <DIR>          ..
10/04/2024  19:32                0 datos.txt
10/04/2024  19:27    <DIR>        PRACTICA
10/04/2024  19:27    <DIR>        TEORIA
                1 archivos                0 bytes
                4 dirs 447.759.187.968 bytes libres
```

Y usamos el comando TYPE para observar el contenido del archivo datos.txt.

```
C:\Users\jhose\LABSO\MEMORIA>TYPE datos.txt
JHOSEP ALONSO MOLLAPAZA MOROCCO
DESARROLLO DE SOFTWARE
UNIVERSIDAD LA SALLE
```

```
C:\Users\jhose\LABSO\MEMORIA>TYPE datos.txt
JHOSEP ALONSO MOLLAPAZA MOROCCO
DESARROLLO DE SOFTWARE
UNIVERSIDAD LA SALLE
```

3.6. Copiar y renombrar datos.txt

Ahora se copiará el archivo datos.txt al directorio TEORIA dentro de PROCESOS, y una vez copiado iremos a esa carpeta para usar DIR y mostrar los archivos.

```
C:\Users\jhose\LABSO\MEMORIA>COPY datos.txt C:\Users\jhose\LABSO\PROCESOS\TEORIA
C:\Users\jhose\LABSO\MEMORIA>CD..
C:\Users\jhose\LABSO>CD PROCESOS
C:\Users\jhose\LABSO\PROCESOS>CD TEORIA
C:\Users\jhose\LABSO\PROCESOS\TEORIA>DIR
```

```
C:\Users\jhose\LABSO\MEMORIA>COPY datos.txt C:\Users\jhose\LABSO\PROCESOS\TEORIA
1 archivo(s) copiado(s).

C:\Users\jhose\LABSO\MEMORIA>CD..

C:\Users\jhose\LABSO>CD PROCESOS

C:\Users\jhose\LABSO\PROCESOS>CD TEORIA

C:\Users\jhose\LABSO\PROCESOS\TEORIA>DIR
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: C2DA-2117

Directorio de C:\Users\jhose\LABSO\PROCESOS\TEORIA

10/04/2024  19:43    <DIR>          .
10/04/2024  19:27    <DIR>          ..
10/04/2024  19:37                77 datos.txt
                1 archivos              77 bytes
                2 dirs 447.748.411.392 bytes libres
```

Ahora cambiaremos el nombre de este archivo datos.txt por nombres.txt y lo verificaremos con el mismo comando DIR

```
C:\Users\jhose\LABSO\PROCESOS\TEORIA>REN datos.txt nombres.txt
C:\Users\jhose\LABSO\PROCESOS\TEORIA>DIR
```

```
C:\Users\jhose\LABSO\PROCESOS\TEORIA>REN datos.txt nombres.txt

C:\Users\jhose\LABSO\PROCESOS\TEORIA>DIR
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: C2DA-2117

Directorio de C:\Users\jhose\LABSO\PROCESOS\TEORIA

10/04/2024  19:47    <DIR>          .
10/04/2024  19:27    <DIR>          ..
10/04/2024  19:37                  77 nombres.txt
                1 archivos                77 bytes
                2 dirs  447.765.762.048 bytes libres
```

3.7. Probando RD, DEL y diferentes parámetros de DIR

Se ha creado una carpeta llamada PRUEBA y un archivo PRUEBA.txt para probar los comandos RD y DEL que son para borrar carpetas y archivos respectivamente.

```
C:\Users\jhose\LABSO>DIR
C:\Users\jhose\LABSO>RD PRUEBA
C:\Users\jhose\LABSO>DEL PRUEBA.txt
```

```
C:\Users\jhose\LABSO>DIR
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: C2DA-2117

Directorio de C:\Users\jhose\LABSO

10/04/2024  19:57    <DIR>          .
10/04/2024  19:18    <DIR>          ..
10/04/2024  19:32    <DIR>          MEMORIA
10/04/2024  19:27    <DIR>          PROCESOS
10/04/2024  19:57    <DIR>          PRUEBA
10/04/2024  19:56                  0 PRUEBA.txt
                1 archivos                0 bytes
                5 dirs  447.755.964.416 bytes libres

C:\Users\jhose\LABSO>RD PRUEBA

C:\Users\jhose\LABSO>DEL PRUEBA.txt

C:\Users\jhose\LABSO>DIR
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: C2DA-2117

Directorio de C:\Users\jhose\LABSO

10/04/2024  20:00    <DIR>          .
10/04/2024  19:18    <DIR>          ..
10/04/2024  19:32    <DIR>          MEMORIA
10/04/2024  19:27    <DIR>          PROCESOS
                0 archivos                0 bytes
                4 dirs  447.756.177.408 bytes libres
```

Probando diferentes parámetros de DIR:

```
C:\Users\jhose\LABSO>DIR /B
MEMORIA
PROCESOS
PRUEBA
PRUEBA.txt
```

DIR /B

```
C:\Users\jhose\LABSO>DIR /W
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: C2DA-2117

Directorio de C:\Users\jhose\LABSO

[.]          [..]          [MEMORIA]    [PROCESOS]    [PRUEBA]      PRUEBA.txt
              1 archivos      0 bytes
              5 dirs  447.755.579.392 bytes libres
```

DIR /W

4. Linux

4.1. Manipulando el shell

Verificamos que la shell esté usando BASH con el siguiente comando:

```
echo $BASH
```

```
[pat@fedora ~]$ echo $BASH
/bin/bash
[pat@fedora ~]$ wh
whatis      whatis.man-db  whereis     which       while       whirlpool-hash  who         whoami      whois       whois.md
[pat@fedora ~]$ 361u
```

4.1.1. Preguntas Shell

1. ¿Cuáles comandos nos muestran el listado de usuarios activos en el sistema? Existen varios comandos pero se suele utilizar users que muestra los nombres de los usuarios, who que además imprime la terminal que están usando, fecha y hora de login y dirección IP o equivalente. Por último está w que imprime los usuarios y sus procesos actuales.

```
$ users
$ who
$ w
```

```
~/uls-24A/operating_systems (0.027s)
users
pat pat

~/uls-24A/operating_systems (0.027s)
who
pat seat0 2024-04-10 19:34 (login screen)
pat tty2 2024-04-10 19:34 (tty2)

~/uls-24A/operating_systems (0.073s)
w
21:28:19 up 1:54, 2 users, load average: 0.67, 0.07, 1.08
USER TTY LOGIN# IDLE JCPU PCPU WHAT
pat seat0 19:34 0.00s 0.00s ? /usr/libexec/gdm-wayland-session /usr/bin/gnome-session
pat tty2 19:34 1:54m 0.07s 0.07s /usr/libexec/gnome-session-binary

~/uls-24A/operating_systems
```

2. ¿Cuál sería el comando para desplegar la fecha del último “boot” (Reinicio) del sistema? Si el comando requiere determinadas opciones, inclúyelas. Para mostrar toda la lista de usuarios se usa last reboot y para mostrar solo el último se usa una pipe para llamar head y -1 para mostrar solo 1 elemento desde el final de la lista

```
$ last reboot
$ last reboot | head -1
```



```

~@fedora-~uls-24A/operating - Level01's banking system +
~/uls-24A/operating_systems (0.031s)
last reboot
reboot system boot 6.5.11-300.fc39, Sun Nov 19 20:02 - 21:01 (00:50)
reboot system boot 6.5.11-300.fc39, Sun Nov 19 18:14 - 19:35 (01:21)
reboot system boot 6.5.11-300.fc39, Sun Nov 19 16:28 - 17:17 (00:49)
reboot system boot 6.5.11-300.fc39, Sun Nov 19 09:57 - 11:37 (01:40)
reboot system boot 6.5.11-300.fc39, Fri Nov 17 15:44 - 19:03 (03:19)
reboot system boot 6.5.11-300.fc39, Fri Nov 17 13:04 - 14:31 (01:26)
reboot system boot 6.5.11-300.fc39, Thu Nov 16 15:32 - 15:41 (00:09)
reboot system boot 6.5.11-300.fc39, Thu Nov 16 14:54 - 15:26 (00:32)
reboot system boot 6.5.11-300.fc39, Wed Nov 15 10:30 - 20:00 (09:30)
reboot system boot 6.5.11-300.fc39, Wed Nov 15 03:43 - 10:29 (06:46)
reboot system boot 6.5.11-300.fc39, Tue Nov 14 13:05 - 19:49 (06:43)
reboot system boot 6.5.11-300.fc39, Fri Nov 10 08:45 - 09:22 (2400:37)
reboot system boot 6.2.9-300.fc38.x Fri Nov 10 08:39 - 08:44 (00:05)
reboot system boot 6.2.9-300.fc38.x Fri Nov 10 08:10 - 08:38 (00:28)
reboot system boot 6.2.9-300.fc38.x Wed Nov 8 18:20 - 19:21 (01:01)
reboot system boot 6.5.8-200.fc38.x Wed Nov 8 17:12 - 17:13 (00:00)
reboot system boot 6.2.9-300.fc38.x Tue Nov 7 11:23 - 10:45 (07:21)
reboot system boot 6.2.9-300.fc38.x Mon Nov 6 22:07 - 22:47 (00:39)
reboot system boot 6.2.9-300.fc38.x Sun Nov 5 18:53 - 22:03 (03:09)
reboot system boot 6.2.9-300.fc38.x Sun Nov 5 14:35 - 17:27 (02:51)
reboot system boot 6.2.9-300.fc38.x Sun Nov 5 07:01 - 10:24 (03:22)
reboot system boot 6.2.9-300.fc38.x Sat Nov 4 13:19 - 18:45 (05:25)
reboot system boot 6.2.9-300.fc38.x Fri Nov 3 17:58 - 18:34 (00:35)
reboot system boot 6.2.9-300.fc38.x Thu Nov 2 20:59 - 22:31 (01:31)
reboot system boot 6.2.9-300.fc38.x Tue Oct 31 21:35 - 11:38 (14:02)
reboot system boot 6.2.9-300.fc38.x Tue Oct 31 11:46 - 16:49 (05:01)
reboot system boot 6.2.9-300.fc38.x Tue Oct 31 06:40 - 11:13 (04:33)
reboot system boot 6.2.9-300.fc38.x Mon Oct 30 22:00 - 00:45 (02:44)
reboot system boot 6.2.9-300.fc38.x Mon Oct 30 12:57 - 13:50 (00:52)
reboot system boot 6.2.9-300.fc38.x Sun Oct 29 16:21 - 16:37 (00:15)
reboot system boot 6.2.9-300.fc38.x Sat Oct 28 20:54 - 07:10 (10:16)
reboot system boot 6.2.9-300.fc38.x Sat Oct 28 08:35 - 10:58 (02:22)
reboot system boot 6.2.9-300.fc38.x Sat Oct 28 08:35 - 08:35 (00:00)
reboot system boot 6.2.9-300.fc38.x Sat Oct 28 07:08 - 08:34 (01:26)
reboot system boot 6.2.9-300.fc38.x Sat Oct 28 00:03 - 00:47 (00:43)
reboot system boot 6.2.9-300.fc38.x Thu Oct 26 19:54 - 20:44 (00:50)

wtmp begins Thu Oct 26 19:54:32 2023

~/uls-24A/operating_systems (0.031s)
last reboot | head -1
reboot system boot 6.8.4-200.fc39.x Wed Apr 10 19:34 still running

~/uls-24A/operating_systems

```

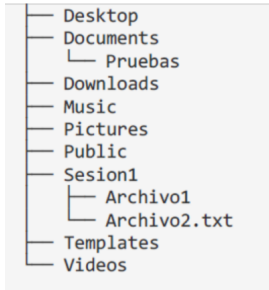
3. Si un archivo tuviese exclusivamente 3 líneas de texto, ¿cuál sería la diferencia de utilizar los comandos head, tail, more y cat?
 - head: Muestra las primeras líneas de un archivo. En este caso, como el archivo solo tiene 3 líneas, mostraría todas ellas.
 - tail: Muestra las últimas líneas de un archivo. Al igual que head mostraría las 3 líneas.
 - more: Permite ver el contenido de un archivo página por página. En un archivo de 3 líneas, mostraría todo el contenido a la vez, ya que no hay suficientes líneas para llenar más de una página.
 - cat: Concatena y muestra todo el contenido de un archivo. En este caso, mostraría las 3 líneas.
4. Si queremos leer el archivo /etc/passwd (el cual contiene el listado de usuarios del sistema) ¿Cuál sería el más apropiado entre los comandos head, tail, more y cat? ¿Por qué?

Depende de lo que desee saber el usuario, pero cat debería ser el mas útil puesto que usualmente se quiere saber todos los usuarios y se puede mostrar con facilidad.
5. ¿Cuál es el comando que se recomienda utilizar en lugar de more?

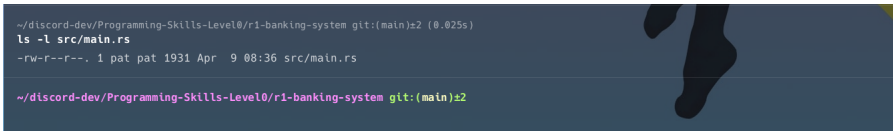
Se utiliza el comando less porque permite navegar hacia atrás en el archivo, además de hacia adelante. Esto lo hace más versátil para la visualización de archivos largos o la inspección de logs.

4.1.2. Preguntas Direccionamiento

Supongamos que nuestro usuario de nombre “fulano” tiene la estructura en su directorio “/home” - obtenida mediante el comando tree - de la siguiente forma:



1. ¿Qué diferencia existe entre Archivo1 y Archivo2.txt? (pista: En linux las “extensiones” como .txt no indican el tipo de archivo, solo se utilizan como convenciones): En Linux las extensiones de archivos solo se utilizan por convención ya que Linux utiliza el contenido y los permisos del archivo para manejarlo. Por ende la diferencia solo radica en el nombre, al menos en lo que concierne al sistema operativo. Esto puede ser comprobado con el comando `ls -l` en un archivo de ejemplo de Rust:



```
~/discord-dev/Programming-Skills-Level0/r1-banking-system git:(main)±2 (0.025s)
ls -l src/main.rs
-rw-r--r-- 1 pat pat 1931 Apr  9 08:36 src/main.rs

~/discord-dev/Programming-Skills-Level0/r1-banking-system git:(main)±2
```

2. Si la línea en bash aparece como (prompt) : fulano@host: /etc\$. ¿Cuál es el comando para desplegar todo el contenido de Archivo2.txt utilizando direccionamiento relativo al directorio en el que nos encontramos? Si el comando requiere determinadas opciones, inclúyelas

```
$ cat ../Sesion1/Archivo2.txt
```

3. ¿Cuál es el comando para desplegar el contenido del directorio Sesion1, incluyendo los directorios lógicos (también llamados simbólicos) y en orden alfabético, utilizando direccionamiento absoluto (es decir, comenzando por la raíz de todos, “/”)?

```
$ ls /Sesion1/ -l
```

4. ¿Cuál es el comando para duplicar la información liberada por tree? Se utilizo -L para manejar la profundidad de los directorios, en este caso solo se utiliza el primer nivel

```
$ tree -L 1
$ tree -L 1 > output.txt
```

5. Valide su respuesta anterior con su propio directorio home, utilizando tanto tree como el comando sugerido por usted.

```

~ (0.028s)
tree -L 1
├── package.json
├── package-lock.json
├── PaperOMG
├── Pictures
├── Portafolio
├── Postman
├── Public
├── Python
├── React
├── Rust
├── SQL
├── Templates
├── tests
├── uls-24A
├── Videos
├── VirtualBox VM
├── VirtualBox VMs
└── Zotero

39 directories, 6 files

~ (0.029s)
tree -L 1 > tree.txt

~ (0.028s)
cat tree.txt
├── Alura
├── armun
├── Aula 6.pdf
├── Azure
├── Books
├── CLionProjects
├── Code In Place
├── Compiler
├── Cursos OMG
├── DataStructures
├── Debian_inst_assets
├── Debian_inst.osp
├── Desktop
├── diagrama.sql
├── discord-dev
└──

```

4.2. Manipulando archivos y directorios

Ejecute los siguientes comandos y conteste las siguientes preguntas:

```

$ mkdir $HOME/Operativos
$ touch $HOME/Operativos/Arch1
$ touch $HOME/Operativos/Arch2
$ touch $HOME/Operativos/Arch3

```

1. Comando para copiar el contenido del archivo /etc/passwd a Arch1

```

$ cp /etc/passwd Operativos/Arch1
$ cat Operativos/Arch1

```

```

~ (0.027s)
cp /etc/passwd Operativos/Arch1

~ (0.028s)
cat Operativos/Arch1
root:x:0:0:Super User:/root:/bin/bash
bin:x:1:1:bin:/bin:/usr/sbin/nologin
daemon:x:2:2:daemon:/sbin:/usr/sbin/nologin
adm:x:3:4:adm:/var/adm:/usr/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/usr/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt

```

2. Comando que copie el archivo Arch1 del paso anterior con el nombre Arch4.

```

# cp crea el archivo de destino si no existe
$ cp Operativos/Arch1 Operativos/Arch4
$ cat Operativos/Arch1

```

```
~ (0.028s)
cp Operativos/Arch1 Operativos/Arch4

~ (0.028s)
cat Operativos/Arch4
root:x:0:0:Super User:/root:/bin/bash
bin:x:1:1:bin:/usr/sbin/nologin
daemon:x:2:2:daemon:/usr/sbin/nologin
adm:x:3:4:adm:/var/adm:/usr/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/usr/sbin/nologin
sync:x:5:0:sync:/bin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
```

- Desde \$HOME/Operativos ejecute el comando: `mkdir ./Acto1`

```
~ (0.025s)
cd Operativos/

~/Operativos (0.025s)
mkdir ./Acto1

~/Operativos (0.028s)
ls
Acto1 Arch1 Arch2 Arch3 Arch4

~/Operativos
```

- Mueva el archivo Arch4 al directorio creado en el paso anterior.

```
$ mv Arch4 Acto1/Arch4
$ tree
```

```
~/Operativos (0.027s)
tree
.
├── Acto1
│   └── Arch4
├── Arch1
├── Arch2
└── Arch3

2 directories, 4 files

~/Operativos
```

- Despliegue la primera línea de Arch4 con direccionamiento absoluto

```
# Se pasa el retorno de cat a head para que solo imprima 1 linea
$ cat Acto1/Arch4 | head -1
```

```
~/Operativos (0.028s)
cat Acto1/Arch4 | head -1
root:x:0:0:Super User:/root:/bin/bash

~/Operativos
```

- Utilice solamente un único comando para borrar todo el contenido del directorio Operativos

```
# Se utiliza rm con -r para recursivo y -f para evitar archivos protegidos
$ rm -rf ~/Operativos/
```

```
~ (0.027s)
rm -rf ~/Operativos/

~ (0.025s)
cd Operativos/
bash: cd: Operativos/: No such file or directory

✚ Insert suggested command: mkdir -p Operativos/ && cd Operativos/

~
mkdir -p Operativos/ && cd Operativos/
```

5. Conclusiones

- Se observa que tanto Windows como Linux ofrecen una gama de comandos para la gestión de archivos y directorios. Por ejemplo, en Windows se utilizan comandos como MD, CD, DIR, y TYPE, mientras que en Linux, comandos como echo, users, who, y w son esenciales para la interacción con el sistema.
- La creación y manipulación de carpetas en Windows parece ser intuitiva con comandos simples. En contraste, Linux brinda una mayor flexibilidad y potencia a través de su shell, lo cual puede resultar más atractivo para usuarios con experiencia.
- Se nota que en Linux, particularmente en distribuciones como Debian o Arch, hay múltiples métodos para instalar programas, ya sea a través de la terminal o utilizando instaladores como snap, proporcionando así una gran versatilidad¹.

6. Recomendaciones

- Es importante familiarizarse con los comandos básicos de Windows y Linux que se presentan, como MD, CD, DIR, TYPE, COPY, REN, DEL, RD y el uso de la terminal y el shell.
- Realizar búsquedas de documentación de comandos Linux en manpages es bastante útil y práctico
- Las flags de los comandos en Linux pueden concatenarse y lograr efectos mas complejos.
- Windows también puede manejarse desde la terminal y aprender los comandos de antemano puede ser beneficioso.