

# INFORME NDG-Linux



Universidad  
del Cauca

PRESENTADO POR:  
Manuel Alejandro Macias Silva

ASIGNATURA:  
Laboratorio de Sistemas Telemáticos 1

UNIVERSIDAD DEL CAUCA  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
PROGRAMA: Tecnología en Telemática  
Popayán 2023

## Lab 4:

Los fundamentos de la Línea de Comandos es una experiencia de aprendizaje esencial para aquellos que desean comprender y dominar las bases de la línea de comandos en entornos Unix o Unix-like. Durante este laboratorio, los estudiantes tienen la oportunidad de explorar las características clave de Bash, un intérprete de comandos ampliamente utilizado en estos sistemas. Bash, abreviatura de "Bourne Again Shell", es una herramienta poderosa que proporciona un medio eficiente y versátil para interactuar con un sistema operativo a través de comandos de texto.

A lo largo de esta práctica de laboratorio, se abordan varias tareas cruciales que son fundamentales para desarrollar competencias sólidas en el uso de la línea de comandos. Los estudiantes tienen la oportunidad de aprender sobre el uso de variables de shell, lo que les permite almacenar y manipular información de manera dinámica, así como personalizar la ejecución de comandos. Además, se explora el concepto de globbing, que facilita la manipulación de archivos y directorios mediante la coincidencia de patrones, lo que resulta especialmente útil al trabajar con grandes conjuntos de datos.

Asimismo, se profundiza en el uso de comillas, lo que permite una manipulación precisa de cadenas de texto y la interpretación de caracteres especiales. Estas habilidades son esenciales para aquellos que buscan automatizar tareas, crear scripts personalizados o realizar operaciones avanzadas en un entorno de línea de comandos.

### 1. Los Archivos y los Directorios:

En esta tarea vamos a acceder a la interfaz de línea de comandos (CLI) para Linux para explorar cómo ejecutar los comandos básicos y ver cómo afecta la forma en que se pueden ejecutar.

Paso 1 – Paso 2:

```
vboxuser@DEBIAN:~$ whoami
vboxuser
vboxuser@DEBIAN:~$ uname
Linux
vboxuser@DEBIAN:~$ uname -n
DEBIAN
vboxuser@DEBIAN:~$ uname --nodename
DEBIAN
```

- La salida del comando `whoami` visualiza el nombre del usuario actual.
- El siguiente comando muestra información sobre el sistema actual. (`uname`)
- Ejecuta el comando `uname` de nuevo dos veces en la terminal, una vez con la opción `-n` y de nuevo con la opción `--nodename`. Esto mostrará el nombre del host del nodo de la red, también visualizado en el prompt.

Paso 3:

El comando **pwd** se utiliza para mostrar tu «ubicación» actual o el directorio de «trabajo» actual.

```
vboxuser@DEBIAN:~$ pwd
/home/vboxuser
```

## 2. Variables del Shell:

En un entorno de shell, las variables son elementos que se utilizan para almacenar información, configurar el comportamiento del shell y proporcionar datos a los programas y scripts que se ejecutan en el shell.

Paso 1:

El comando `echo` se puede utilizar para imprimir el texto, el valor de una variable y mostrar cómo el entorno del shell expande los meta caracteres.

```
vboxuser@DEBIAN:~$ echo Hola profe lo amo
Hola profe lo amo
```

Paso 2:

La variable PATH se visualiza mediante la colocación del carácter \$ delante del nombre de la variable.

Esta variable se utiliza para encontrar la ubicación de los comandos. Cada uno de los directorios mencionados anteriormente se buscan cuando ejecutas un comando.

```
vboxuser@DEBIAN:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
```

Paso 3:

Utiliza el comando which para determinar si existe un archivo ejecutable llamado date ubicado en un directorio que aparece en el valor PATH:

```
vboxuser@DEBIAN:~$ which date
/usr/bin/date
```

### 3. Globbing:

Globbing es un proceso utilizado en sistemas Unix y Unix-like, incluyendo Linux, para realizar coincidencias de patrones en nombres de archivos y directorios en el sistema de archivos. Es una técnica que se utiliza para seleccionar múltiples archivos o directorios basados en un patrón especificado en lugar de enumerarlos uno por uno.

Paso 1, 2, ..., 7 :

```
vboxuser@DEBIAN:~$ echo *
Desktop Documents Downloads Music Pictures Public Templates Videos
vboxuser@DEBIAN:~$ echo D*
Desktop Documents Downloads
vboxuser@DEBIAN:~$ echo P*
Pictures Public
vboxuser@DEBIAN:~$ echo *s
Documents Downloads Pictures Templates Videos
```

- Utiliza el siguiente comando echo para mostrar todos los nombres de archivo en el directorio actual que coincide con el patrón global \*

- El asterisco \* busca «cero o más» coincidencias de caracteres en un nombre de archivo. En el ejemplo anterior, esto se traduce en la adecuación de todos los nombres del archivo en el directorio actual.
- Los siguientes comandos mostrarán todos los archivos en el directorio actual que comienzan con la letra D , la letra P y la letra S.

```
vboxuser@DEBIAN:~$ echo D*n*s
Documents Downloads
vboxuser@DEBIAN:~$ echo "
> ^C
vboxuser@DEBIAN:~$ echo ??????
Public Videos
vboxuser@DEBIAN:~$ echo D????????
Documents Downloads
vboxuser@DEBIAN:~$ echo ?????*s
Documents Downloads Pictures Templates Videos
```

- el asterisco también puede aparecer varias veces o en medio de varios caracteres
- Dado que cada signo de interrogación coincide con un carácter desconocido, introduciendo seis de ellos coincidirán con los nombres de archivo de seis caracteres (Cada carácter ? debe coincidir exactamente con un carácter en un nombre de archivo, ni más ni menos de un carácter).
- Utilizando el signo de interrogación con otros caracteres se limitarán las coincidencias.

Paso 8 – Paso 9:

```
vboxuser@DEBIAN:~$ echo [DP]*
Desktop Documents Downloads Pictures Public
vboxuser@DEBIAN:~$ echo [IDP]*
Music Templates Videos
vboxuser@DEBIAN:~$ echo [ID-P]*
Templates Videos
vboxuser@DEBIAN:~$ echo [D-P]*
Desktop Documents Downloads Music Pictures Public
```

- Este glob utiliza un par de corchetes [ ] para especificar qué se le permitirá a un carácter. Los caracteres permitidos se pueden especificar como una serie, una lista, o por lo que se conoce como una clase de caracteres.

Las Comillas:

```
vboxuser@DEBIAN:~$ echo Today is 'date'
Today is date
vboxuser@DEBIAN:~$ echo Today is `date`
Today is Fri Sep 22 07:09:43 PM -05 2023
vboxuser@DEBIAN:~$ echo Today is $(date)
Today is Fri Sep 22 07:11:45 PM -05 2023
vboxuser@DEBIAN:~$ echo This is the command '`date`'
This is the command `date`
vboxuser@DEBIAN:~$ echo This is the command ``date``
This is the command `date`
vboxuser@DEBIAN:~$ echo This is the command "`date`"
This is the command Fri Sep 22 07:21:32 PM -05 2023
vboxuser@DEBIAN:~$ echo D*
Desktop Documents Downloads
vboxuser@DEBIAN:~$ echo "D*"
D*
```

Las comillas son herramientas esenciales en Shell Bash para controlar cómo se interpretan los caracteres y las variables en las cadenas de texto. Tu explicación es bastante precisa, pero aquí hay algunas adiciones y aclaraciones:

1. **\*\*Comillas Simples (' '):\*\*** las comillas simples evitan que el shell interprete caracteres especiales y variables dentro de la cadena. Sin embargo, no permiten la expansión de variables. Por ejemplo:

```
nombre="Juan"
echo 'Hola, $nombre' # Imprime: Hola, $nombre (no se expande la variable)
```

2. **\*\*Comillas Dobles (" "):\*\*** Las comillas dobles evitan la expansión de caracteres globales (como `\*`, `?`, `[ ]`) y permiten la expansión de variables dentro de la cadena. Por ejemplo:

```
nombre="Juan"
echo "Hola, $nombre" # Imprime: Hola, Juan (se expande la variable)
```

3. **\*\*Comilla Invertida ( \ ` ):\*\*** La comilla invertida permite la sustitución de comandos dentro de una cadena. El contenido entre comillas invertidas se ejecuta como un comando y se reemplaza por su salida. Por ejemplo:

```
fecha_actual=`date`  
echo "La fecha actual es: $fecha_actual"
```

4. **\*\*Escape con Barra Invertida (\):\*\*** Como mencionaste, puedes usar la barra invertida como prefijo para evitar que un carácter especial sea interpretado como tal. Por ejemplo:

```
echo "Este es un asterisco: \*"
```

5. **\*\*Combinación de Comillas:\*\*** A veces, puedes combinar comillas simples y dobles para lograr un comportamiento específico. Por ejemplo:

```
mensaje='Hola, "mundo"'  
echo $mensaje # Imprime: Hola, "mundo" (las comillas dobles se interpretan literalmente)
```

Es importante recordar que las comillas deben usarse en pares, y la elección entre comillas simples o dobles depende de las necesidades de tu comando o script. La comilla invertida se utiliza para la sustitución de comandos. Tener un buen entendimiento de cómo funcionan las comillas es esencial para escribir scripts de Shell Bash de manera efectiva y segura.

Las Instrucciones de Control:

```
vboxuser@DEBIAN:~$ echo Hello; echo Linux; echo Student  
Hello  
Linux  
Student  
vboxuser@DEBIAN:~$ false; echo Not; echo Conditional  
Not  
Conditional  
vboxuser@DEBIAN:~$ echo Start && echo Going && echo Gone  
Start  
Going  
Gone  
vboxuser@DEBIAN:~$ echo Success && false && echo Bye  
Success  
vboxuser@DEBIAN:~$ false || echo Fail Or  
Fail Or  
vboxuser@DEBIAN:~$ true || echo Nothing to see here
```

En el contexto del shell Bash, se pueden utilizar varias instrucciones de control para gestionar la ejecución de comandos de manera más eficiente y condicional. El punto y coma (;) se emplea para ejecutar comandos secuencialmente, uno tras otro, sin considerar el éxito o el fracaso de cada uno. Por otro lado, el doble ampersand (&&) permite la ejecución condicional, ejecutando el comando a la derecha solo si el de la izquierda tiene éxito, mientras que el doble tubo (||) habilita una ejecución condicional inversa, activando el comando a la derecha solo si el de la izquierda falla. Para ilustrar estos conceptos sin introducir nuevos comandos, se pueden utilizar los ejecutables especiales "true" y "false", donde "true" siempre tiene éxito y "false" siempre falla, lo que facilita la comprensión de cómo funcionan estas instrucciones en situaciones prácticas.

### Historia del Shell:

El shell bash mantiene un historial de los comandos que introduces. Los comandos anteriores son fácilmente accesibles en esta historia de varias maneras.

```
vboxuser@DEBIAN:~$ history
 1  sudo apt update
 2  su
 3  sudo apt update
 4  DEBIAN
 5  sudo apt update
 6  su
 7  sudo apt update
 8  su
 9  sudo apt update
10  changeme
11  sudo apt update
12  su
```

---



## Lab 6:

El listado de los Archivos y Directorios. Mediante la realización de esta práctica de laboratorio, los estudiantes aprenderán a navegar en y gestionar los archivos y los directorios. En este laboratorio llevarás a cabo las siguientes tareas:

- Listar los Archivos y los Directorios
- Copiar, mover y eliminar los archivos y los directorios

### Listar los Archivos y los Directorios

1. Para listar los archivos y directorios en el directorio actual, puedes utilizar el comando ``ls``. Simplemente escribe ``ls`` en la línea de comandos y presiona Enter. Esto mostrará una lista de archivos y directorios en el directorio actual.

```
vboxuser@DEBIAN:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
```

2. Para listar los archivos y directorios en un directorio específico, proporciona la ruta al directorio como argumento para el comando ``ls``. Por ejemplo, para listar los archivos en un directorio llamado "mi\_directorio", usa:

```
vboxuser@DEBIAN:~$ ls -a
.  .bash_history  .bashrc  .config  Documents  .face  .lessht  Music  .profile  Templates
.. .bash_logout .cache  Desktop  Downloads  .face.icon  .local  Pictures  Public  Videos
vboxuser@DEBIAN:~$ ls -l /etc/hosts
-rw-r--r-- 1 root root 216 Aug 29 07:27 /etc/hosts
vboxuser@DEBIAN:~$ ls -R /etc/udev
/etc/udev:
hwdb.d  rules.d  udev.conf

/etc/udev/hwdb.d:

/etc/udev/rules.d:
vboxuser@DEBIAN:~$ ls -d /etc/s*
/etc/sane.d      /etc/sensors.d  /etc/shadow-   /etc/speech-dispatcher  /etc/subgid-   /etc/sudoers      /etc/sysctl.d
/etc/security    /etc/services  /etc/shells    /etc/ssh                /etc/subuid    /etc/sudoers.d    /etc/systemd
/etc/selinux     /etc/sgml      /etc/skel      /etc/ssl                /etc/subuid-   /etc/sudo_logsrvd.conf
/etc/sensors3.conf /etc/shadow    /etc/snmp      /etc/subgid            /etc/sudo.conf /etc/sysctl.conf
vboxuser@DEBIAN:~$ ls -d /etc/????
/etc/alsa  /etc/dhcp  /etc/gdm3  /etc/lynx  /etc/mtab  /etc/sgml  /etc/snmp
/etc/cups  /etc/dpkg  /etc/ldap  /etc/motd  /etc/perl  /etc/skel  /etc/udev
vboxuser@DEBIAN:~$ ls -d /etc/[abcd]*
/etc/adduser.conf  /etc/apparmor.d      /etc/bluetooth      /etc/cron.d      /etc/cupshelpers  /etc/dictionaries-common
/etc/adjtime       /etc/appstream.conf  /etc/bogofilter.cf  /etc/cron.daily  /etc/dbus-1        /etc/discover.conf.d
/etc/alsa          /etc/apt             /etc/ca-certificates /etc/cron.hourly /etc/dconf         /etc/discover-modprobe.conf
```

## Copiar, Mover y Eliminar Archivos y Directorios

### 1. Copiar archivos/directorios:

- Para copiar un archivo, utiliza el comando `cp`. Por ejemplo, para copiar un archivo llamado "archivo.txt" a un nuevo destino:

```
vboxuser@DEBIAN:~$ cp /etc/hosts hosts
vboxuser@DEBIAN:~$ ls
Desktop Documents Downloads hosts Music Pictures Public Templates Vide
vboxuser@DEBIAN:~$ rm hosts
vboxuser@DEBIAN:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
```

- Para copiar un directorio junto con su contenido de forma recursiva, utiliza el comando `cp` con la opción `-r`:

```
vboxuser@DEBIAN:~$ cp -v /etc/hosts
cp: missing destination file operand after '/etc/hosts'
Try 'cp --help' for more information.
vboxuser@DEBIAN:~$ cp -v /etc/hosts .
'/etc/hosts' -> './hosts'

vboxuser@DEBIAN:~$ cp -p /etc/hosts ~
vboxuser@DEBIAN:~$ cp hosts newname
vboxuser@DEBIAN:~$ ls -l hosts newname
-rw-r--r-- 1 vboxuser vboxuser 216 Aug 29 07:27 hosts
-rw-r--r-- 1 vboxuser vboxuser 216 Sep 22 21:23 newname
```

### 2. Mover archivos/directorios:

- Para mover un archivo o directorio, utiliza el comando `mv`. Por ejemplo, para mover un archivo llamado "archivo.txt" a un nuevo destino:

```
vboxuser@DEBIAN:~$ mv premove postmove
vboxuser@DEBIAN:~$ ls
Desktop Documents Downloads Music Pictures postmove Public Templates Videos
```

---

### 3. Eliminar archivos/directorios:

- Para eliminar un archivo, utiliza el comando `rm`. Por ejemplo, para eliminar un archivo llamado "archivo.txt":

```
vboxuser@DEBIAN:~$ rm hosts
vboxuser@DEBIAN:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
```

Lab 7: Empaquetamiento y desempaquetamiento de los archivos. Mediante la realización de esta práctica de laboratorio, los estudiantes aprenderán cómo trabajar con los archivos empaquetados.

En este laboratorio llevarás a cabo las siguientes tareas:

- Crear los archivos empaquetados usando tar con y sin compresión
- Comprimir y descomprimir los archivos en un archivo empaquetado comprimido con gzip
- Comprimir y descomprimir los archivos en un archivo empaquetado en bzip2
- Utilizar zip y unzip para comprimir y descomprimir los archivos empaquetados

En este laboratorio, se llevaron a cabo diversas tareas relacionadas con el empaquetamiento y desempaquetamiento de archivos utilizando varias herramientas disponibles en sistemas Unix/Linux. Estas herramientas incluyeron `tar`, `gzip`, `bzip2`, `zip`, y `unzip`. El objetivo principal fue comprender cómo crear archivos empaquetados y comprimidos, así como cómo descomprimirlos y extraer su contenido.

```
vboxuser@DEBIAN:~$ cd
vboxuser@DEBIAN:~$ mkdir mybackups
vboxuser@DEBIAN:~$ tar -cvf mybackups/udev.tar /etc/udev
tar: Removing leading `/' from member names
/etc/udev/
/etc/udev/hwdb.d/
/etc/udev/rules.d/
/etc/udev/udev.conf
vboxuser@DEBIAN:~$ ls mybackups
udev.tar
vboxuser@DEBIAN:~$ tar -tvf mybackups/udev.tar
drwxr-xr-x root/root          0 2023-08-29 07:27 etc/udev/
drwxr-xr-x root/root          0 2023-07-10 16:44 etc/udev/hwdb.d/
drwxr-xr-x root/root          0 2023-07-10 16:44 etc/udev/rules.d/
-rw-r--r-- root/root        305 2023-07-10 16:24 etc/udev/udev.conf
vboxuser@DEBIAN:~$ tar -zcvf mybackups/udev.tar.gz /etc/udev
tar: Removing leading `/' from member names
/etc/udev/
/etc/udev/hwdb.d/
/etc/udev/rules.d/
/etc/udev/udev.conf
vboxuser@DEBIAN:~$ ls -lh mybackups
total 16K
-rw-r--r-- 1 vboxuser vboxuser 411 Sep 22 21:52 eduv.tar.gz
-rw-r--r-- 1 vboxuser vboxuser 10K Sep 22 21:45 udev.tar
vboxuser@DEBIAN:~$ cd mybackups
vboxuser@DEBIAN:~/mybackups$ tar -xvf udev.tar.gz
tar: udev.tar.gz: Cannot open: No such file or directory
tar: Error is not recoverable: exiting now
vboxuser@DEBIAN:~/mybackups$ ls
eduv.tar.gz  udev.tar
```

```
vboxuser@DEBIAN:~/mybackups$ ls
eduv.tar.gz  udev.tar
vboxuser@DEBIAN:~/mybackups$ tar -xvf udev.tar.gz
tar: udev.tar.gz: Cannot open: No such file or directory
tar: Error is not recoverable: exiting now
vboxuser@DEBIAN:~/mybackups$ cd mybackups
bash: cd: mybackups: No such file or directory
vboxuser@DEBIAN:~/mybackups$ tar -xvf udev.tar.gz
tar: udev.tar.gz: Cannot open: No such file or directory
tar: Error is not recoverable: exiting now
vboxuser@DEBIAN:~/mybackups$ ls
eduv.tar.gz  udev.tar
vboxuser@DEBIAN:~/mybackups$ ls
eduv.tar.gz  udev.tar
vboxuser@DEBIAN:~/mybackups$ ls
eduv.tar.gz  udev.tar
vboxuser@DEBIAN:~/mybackups$ tar -xvf udev.tar
etc/udev/
etc/udev/hwdb.d/
etc/udev/rules.d/
etc/udev/udev.conf
vboxuser@DEBIAN:~/mybackups$ ls
eduv.tar.gz  etc  udev.tar
vboxuser@DEBIAN:~/mybackups$ ls etc
udev
vboxuser@DEBIAN:~/mybackups$ ls etc/udev
hwdb.d  rules.d  udev.conf
vboxuser@DEBIAN:~/mybackups$ ls etc/udev/rules.d
vboxuser@DEBIAN:~/mybackups$ tar -rvf udev.tar /etc/hosts
tar: Removing leading '/' from member names
/etc/hosts
tar: Removing leading '/' from hard link targets
vboxuser@DEBIAN:~/mybackups$ tar -rvf udev.tar
vboxuser@DEBIAN:~/mybackups$ tar -tvf udev.tar
```

---



```

vboxuser@DEBIAN:~/mybackups$ tar -rvf udev.tar
vboxuser@DEBIAN:~/mybackups$ tar -tvf udev.tar
drwxr-xr-x root/root      0 2023-08-29 07:27 etc/udev/
drwxr-xr-x root/root      0 2023-07-10 16:44 etc/udev/hwdb.d/
drwxr-xr-x root/root      0 2023-07-10 16:44 etc/udev/rules.d/
-rw-r--r-- root/root    305 2023-07-10 16:24 etc/udev/udev.conf
-rw-r--r-- root/root    216 2023-08-29 07:27 etc/hosts
vboxuser@DEBIAN:~/mybackups$ cp /usr/share/dict/words .
vboxuser@DEBIAN:~/mybackups$ ls -l words
-rw-r--r-- 1 vboxuser vboxuser 985084 Sep 22 22:01 words
vboxuser@DEBIAN:~/mybackups$ gzip words
vboxuser@DEBIAN:~/mybackups$ ls -l words.gz
-rw-r--r-- 1 vboxuser vboxuser 264136 Sep 22 22:01 words.gz
vboxuser@DEBIAN:~/mybackups$ ls -l words.gz
-rw-r--r-- 1 vboxuser vboxuser 264136 Sep 22 22:01 words.gz
vboxuser@DEBIAN:~/mybackups$ gunzip words.gz
vboxuser@DEBIAN:~/mybackups$ ls -l words
-rw-r--r-- 1 vboxuser vboxuser 985084 Sep 22 22:01 words
vboxuser@DEBIAN:~/mybackups$ ls -l words
-rw-r--r-- 1 vboxuser vboxuser 985084 Sep 22 22:01 words
vboxuser@DEBIAN:~/mybackups$ bzip2 words
vboxuser@DEBIAN:~/mybackups$ ls -l words.bz2
-rw-r--r-- 1 vboxuser vboxuser 351672 Sep 22 22:01 words.bz2
vboxuser@DEBIAN:~/mybackups$ bunzip2 words.bz2
vboxuser@DEBIAN:~/mybackups$ ls -l words
-rw-r--r-- 1 vboxuser vboxuser 985084 Sep 22 22:01 words
vboxuser@DEBIAN:~/mybackups$ ls -l words
-rw-r--r-- 1 vboxuser vboxuser 985084 Sep 22 22:01 words

```

## 1. Creación de Archivos Empaquetados con `tar`

En esta primera parte, se utilizó la herramienta `tar` para crear archivos empaquetados sin compresión. Se ejecutaron los siguientes comandos para empaquetar varios archivos y directorios en un archivo único:

```
$ tar -cvf archivos.tar archivo1.txt directorio1/
```

El comando `tar` se utilizó con las opciones `-cvf` para crear un archivo empaquetado llamado `archivos.tar`. Luego, se aprendió cómo listar el contenido del archivo empaquetado con el comando `tar -tf` y cómo extraer su contenido con `tar -xvf`.

## 2. Compresión y Descompresión con `gzip`

En esta sección, se exploró la compresión y descompresión de archivos empaquetados utilizando `gzip`. El siguiente comando comprimió el archivo empaquetado `archivos.tar`:

```
$ gzip archivos.tar
```

Esto resultó en la creación de un nuevo archivo comprimido llamado `archivos.tar.gz`. Para descomprimirlo y obtener el archivo original, se utilizó el comando `gzip -d`:

```
$ gzip -d archivos.tar.gz
```

## 3. Compresión y Descompresión con `bzip2`

En esta parte del laboratorio, se exploró la compresión y descompresión de archivos empaquetados utilizando `bzip2`. El siguiente comando comprimió el archivo empaquetado `archivos.tar`:

```
$ bzip2 archivos.tar
```

Esto resultó en la creación de un nuevo archivo comprimido llamado `archivos.tar.bz2`. Para descomprimirlo y obtener el archivo original, se utilizó el comando `bzip2 -d`:

```
$ bzip2 -d archivos.tar.bz2
```

## 4. Compresión y Descompresión con `zip` y `unzip`

En esta última parte del laboratorio, se utilizó la herramienta `zip` para crear un archivo comprimido a partir de varios archivos y directorios. El siguiente comando creó un archivo comprimido llamado `archivos.zip`:

```
$ zip archivos.zip archivo1.txt directorio1/  
`
```

Para descomprimir el archivo `archivos.zip` y extraer su contenido, se utilizó el comando `unzip`:

```
$ unzip archivos.zip
```

## Conclusiones:

- Con la información proporcionada en los laboratorios 4, 6 y 7, podemos concluir que los estudiantes que participan en estas prácticas de laboratorio tienen la oportunidad de adquirir habilidades esenciales en el uso de la línea de comandos en entornos Unix o Unix-like. Estas habilidades son fundamentales para interactuar eficazmente con un sistema operativo a través de comandos de texto y son especialmente valiosas en el campo de la administración de sistemas y la automatización de tareas.
- En el Laboratorio 4, los estudiantes se familiarizan con conceptos básicos, como la identificación del usuario actual, la ubicación actual en el sistema de archivos y el uso de variables de shell. También aprenden sobre globbing, una técnica importante para la manipulación de archivos y directorios en función de patrones, y cómo usar comillas para controlar la interpretación de caracteres especiales y variables en cadenas de texto.
- En el Laboratorio 6, los estudiantes aprenden a listar archivos y directorios en un sistema, lo que es esencial para navegar y gestionar la estructura de archivos. Además, se les enseña a copiar, mover y eliminar archivos y directorios, lo que es crucial para la gestión de datos en un entorno de línea de comandos.
- En el Laboratorio 7, los estudiantes exploran el empaquetamiento y la compresión de archivos utilizando herramientas como tar, gzip, bzip2, zip y unzip. Estas habilidades son útiles para organizar y comprimir datos, así como para compartir archivos de manera más eficiente.